

Gentran:Basic<sup>®</sup> for zSeries

# Technical Reference Guide

Release 6.4

**Sterling Commerce**  
*An IBM Company*

Sterling Commerce (Mid America), Inc.  
4600 Lakehurst Court Dublin, OH 43016-2000 \* 614/793-4000  
© Copyright 1988 – 2006—Sterling Commerce, Inc.

March 2006

Gentran:Basic for zSeries  
© Copyright 1988 – 2006  
Sterling Commerce, Inc.  
ALL RIGHTS RESERVED

## **Sterling Commerce Software**

### **Trade Secret Notice**

**THE GENTRAN: BASIC FOR ZSERIES SOFTWARE (“STERLING COMMERCE SOFTWARE”) IS THE CONFIDENTIAL AND TRADE SECRET PROPERTY OF STERLING COMMERCE, INC., ITS AFFILIATED COMPANIES OR ITS OR THEIR LICENSORS, AND IS PROVIDED UNDER THE TERMS OF A LICENSE AGREEMENT. NO DUPLICATION OR DISCLOSURE WITHOUT PRIOR WRITTEN PERMISSION. RESTRICTED RIGHTS.**

This documentation, the Sterling Commerce Software it describes, and the information and know-how they contain constitute the proprietary, confidential and valuable trade secret information of Sterling Commerce, Inc., its affiliated companies or its or their licensors, and may not be used for any unauthorized purpose, or disclosed to others without the prior written permission of the applicable Sterling Commerce entity. This documentation and the Sterling Commerce Software that it describes have been provided pursuant to a license agreement that contains prohibitions against and/or restrictions on their copying, modification and use. Duplication, in whole or in part, if and when permitted, shall bear this notice and the Sterling Commerce, Inc. copyright notice.

As and when provided to any governmental entity, government contractor or subcontractor subject to the FARs, this documentation is provided with RESTRICTED RIGHTS under Title 48 CFR 52.227-19. Further, as and when provided to any governmental entity, government contractor or subcontractor subject to DFARs, this documentation and the Sterling Commerce Software it describes are provided pursuant to the customary Sterling Commerce license, as described in Title 48 CFR 227-7202 with respect to commercial software and commercial software documentation.

These terms of use shall be governed by the laws of the State of Ohio, USA, without regard to its conflict of laws provisions. If you are accessing the Sterling Commerce Software under an executed agreement, then nothing in these terms and conditions supersedes or modifies the executed agreement.

Product names mentioned herein may be trademarks and/or registered trademarks of their respective companies. Gentran, Gentran:Basic, Gentran:Control, Gentran:Plus, Gentran:Realtime, Gentran:Structure, and Gentran:Viewpoint are registered trademarks of Sterling Commerce, Inc.

### **Third Party Software:**

Portions of the Sterling Commerce Software may include products, or may be distributed on the same storage media with products, (“Third Party Software”) offered by third parties (“Third Party Licensors”).

### **Warranty Disclaimer**

This documentation and the Sterling Commerce Software which it describes are licensed either "AS IS" or with a limited warranty, as set forth in the Sterling Commerce license agreement. Other than any limited warranties provided, NO OTHER WARRANTY IS EXPRESSED AND NONE SHALL BE IMPLIED, INCLUDING THE WARRANTIES OF MERCHANTABILITY AND FITNESS FOR USE OR FOR A PARTICULAR PURPOSE. The applicable Sterling Commerce entity reserves the right to revise this publication from time to time and to make changes in the content hereof without the obligation to notify any person or entity of such revisions or changes.

The Third Party Software is provided 'AS IS' WITHOUT ANY WARRANTY AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. FURTHER, IF YOU ARE LOCATED OR ACCESSING THIS SOFTWARE IN THE UNITED STATES, ANY EXPRESS OR IMPLIED WARRANTY REGARDING TITLE OR NON-INFRINGEMENT ARE DISCLAIMED.

# Table of Contents

<b>Chapter 1</b>	<b>Getting Started</b>	
	Overview .....	1-1
<b>Chapter 2</b>	<b>Inbound Processing Programs</b>	
	Overview .....	2-1
	System Flow .....	2-3
	Inbound Unwrap Processing .....	EBDI010, EBDI010A ..... 2-7
	Inbound Editor .....	EBDI001 ..... 2-9
	Inbound Split .....	EBDI015 ..... 2-55
	Inbound Data Mapping.....	EBDI041 ..... 2-62
<b>Chapter 3</b>	<b>Outbound Processing Programs</b>	
	Overview .....	3-1
	System Flow .....	3-3
	Outbound Data Mapping .....	EBDI042 ..... 3-5
	Outbound Envelope Generation .....	EBDI011M, EBDI011A 3-63
	Outbound Editor .....	EBDI002 ..... 3-77
	Outbound Split.....	EBDI016 ..... 3-119
<b>Chapter 4</b>	<b>Utility Programs</b>	
	Overview .....	4-1
	Application Change Audit Maintenance Program .....	EBDI066 ..... 4-3
	Application Copy from One Set of VSAM Files to Another.....	EBDI060 ..... 4-7
	Application Definition Maintenance .....	EDIG100 ..... 4-12
	Application Sequence Renumbering .....	EBDI090 ..... 4-18
	Batch Separator .....	EBDI930 ..... 4-27
	COBOL Copybook Generator.....	EDIG110 ..... 4-30
	Code Table Change Audit Maintenance.....	EBDI068 ..... 4-33
	Code Table Copy From File to File.....	EBDI076 ..... 4-36
	Configuration Change Audit Maintenance Program.....	EBDI102 ..... 4-45
	Data Router.....	EBDI095 ..... 4-49
	Data Router Parameters.....	EBDI095 ..... 4-52
	Data Routing.....	EBDI095 ..... 4-58
	Defining the Rules for the Split.....	4-60
	Error Message Change Audit Maintenance Program.....	EBDI105 ..... 4-64
	Error Message Maintenance .....	EBDIT13 ..... 4-68
	Global Parameter Change Audit Maintenance Program.....	EBDI103 ..... 4-71
	JCL File Maintenance.....	EDIRJCLX ..... 4-75
	Mapping Version Comparison and Conversion Report .....	EBDI088 ..... 4-80
	Object Insert Program.....	EBDI065 ..... 4-97
	Online Log File Utility .....	EDIELMT ..... 4-102

Partner Change Audit Maintenance Program.....	EBDI093.....	4-105
Partners Copy From One File to Another.....	EBDI087.....	4-110
Security Change Audit Maintenance Program.....	EBDI100.....	4-115
Separator Change Audit Maintenance Program.....	EBDI104.....	4-118
Sort Generated Acknowledgments Program.....	EBDI110.....	4-122
Standard Change Audit Maintenance Program.....	EBDI033.....	4-123
Standards Copy Utility.....	EBDI032.....	4-126
Standards Selective Download.....	EBDI030.....	4-131
Standards Update.....	EBDI038, EBDI039.....	4-135
Transaction Change Audit Maintenance Program.....	EBDI067.....	4-142
Transaction Copy From One Set of VSAM Files to Another.....	EBDI080.....	4-146
Transaction Mapping Definition Re-synch.....	EBDI089.....	4-151

## Chapter 5 Databank Utility Programs and Reports

Overview.....		5-1
Outbound Processing Flow.....		5-2
Databank Control Card Format.....		5-3
Online Databank Maintenance.....		5-4
Acknowledgment Reconciliation/Monitor Program.....	EDID510.....	5-5
Application Databank Inquiry Report.....	EDID551.....	5-8
EDI Archive Inquiry Program.....	EDID555.....	5-19
Databank Change Audit Inquiry Program.....	EDID500.....	5-27
Databank Change Audit Maintenance Program.....	EDID502.....	5-31
EDI Databank Inquiry Program.....	EDID550.....	5-37
Inbound Application Databank Extract Program.....	EDID405.....	5-59
Inbound Application Databank Maintenance Program..	EDID401.....	5-66
Inbound EDI Databank Maintenance Program.....	EDID301.....	5-73
Outbound Application Databank Maintenance Program	EDID101.....	5-80
Outbound EDI Databank Extract Program.....	EDID205.....	5-87
Outbound EDI Databank Maintenance Program.....	EDID201.....	5-99
Transaction Queue Initialize Program.....	EDID520.....	5-107
EDI Databank Orphan Record Cleanup Program.....	EDIG300.....	5-109
Program Report Names and Descriptions.....		5-112

## Chapter 6 Network Utility Programs

Overview.....		6-1
Commerce:Network Response Processor.....	EDID800.....	6-2
Commerce:Network Response Processor (International).....	EDID800I.....	6-4
GEIS Response Processor.....	EDID805.....	6-6
IBM IN Response Processor.....	EDID810.....	6-7
Network Reconciliation.....	EDID850.....	6-9
Network Reconciliation Maintenance.....	EDID860.....	6-11
Universal Response Processor.....	EDID825.....	6-14

## Chapter 7 Reporting Programs

Overview.....		7-1
Acknowledgment Report.....	EBDI017, EBDI018.....	7-2

Application Change Audit Inquiry .....	EBDI062.....	7-10
Code Table Change Audit Inquiry .....	EBDI064.....	7-13
Configuration Change Audit Inquiry Report Program...	EBDI070.....	7-16
EDI Data Print Report .....	EBDI019.....	7-19
Error Message Change Audit Inquiry.....	EBDI069.....	7-24
Global Parameter Change Audit Inquiry.....	EBDI071.....	7-27
JCL File Report .....	EDIRJCLR .....	7-29
Mapping Cross Reference Report .....	EBDI091.....	7-32
Mapping Report – Application Sequence.....	EBDI053.....	7-42
Mapping Report – Standard Sequence .....	EBDI052.....	7-50
Mapping Report Summary .....	EBDI061.....	7-59
Mapping Validation Report.....	EBDI082.....	7-66
Partner Change Audit Inquiry Report Program.....	EBDI092.....	7-72
Partner File Inquiry Report.....	EBDI008.....	7-76
Partner Profile Print Report.....	EBDI006.....	7-82
Security Change Audit Inquiry.....	EBDI073.....	7-93
Security File Display Report .....	EBDI009.....	7-95
Separator Change Audit Inquiry.....	EBDI072.....	7-97
Standard Change Audit Inquiry Report Program .....	EBDI037.....	7-100
Standard Optimized Table Report .....	EBDI035.....	7-105
Standard Report.....	EBDI036.....	7-116
Transaction Change Audit Inquiry Report Program .....	EBDI063.....	7-128

## Chapter 8 Databank Files

Overview .....		8-1
Acknowledgment Reconciliation File .....	EDIDAR.....	8-3
Change Audit Databank Archive File .....	EDICAAR.....	8-4
Formatted Network File .....	EDIFNF.....	8-5
Inbound Application Databank Archive File .....	EDIIAAR.....	8-6
Inbound Application Databank Change Audit File .....	EDIICA.....	8-7
Inbound Application Databank Directory .....	EDIIA.....	8-9
Inbound Application Databank Message Store.....	EDIAS.....	8-11
Inbound Application Databank Pending File .....	EDIAP.....	8-13
Inbound Application Databank Transaction Queue .....	EDIQ095.....	8-15
Inbound EDI Databank Archive File.....	EDIIEAR.....	8-17
Inbound EDI Databank Change Audit File .....	EDIIECA.....	8-18
Inbound EDI Databank Directory File .....	EDIIEA.....	8-20
Inbound EDI Databank Link File.....	EDIIEL.....	8-22
Inbound EDI Databank Message Store File.....	EDIIES.....	8-24
Inbound EDI Databank Pending File .....	EDIIEP.....	8-27
Inbound EDI Databank Transaction Queue .....	EDIQ097.....	8-28
Network Reconciliation File.....	EDINRC.....	8-30
Network Response File.....	EDINR.....	8-31
Outbound Application Databank Archive File.....	EDIOAAR.....	8-32
Outbound Application Databank Change Audit File .....	EDIOACA.....	8-33
Outbound Application Databank Directory.....	EDIOAA.....	8-35
Outbound Application Databank Link File .....	EDIOAL.....	8-37

## Table of Contents

---

Outbound Application Databank Message Store .....	EDIOAS .....	8-39
Outbound Application Databank Pending File.....	EDIOAP .....	8-41
Outbound Application Databank Transaction Queue....	EDIQ093 .....	8-42
Outbound EDI Databank Archive File .....	EDIOEAR .....	8-44
Outbound EDI Databank Change Audit File.....	EDIOECA .....	8-45
Outbound EDI Databank Directory.....	EDIOEA .....	8-47
Outbound EDI Databank Message Store.....	EDIOES .....	8-50
Outbound EDI Databank Pending File.....	EDIOEP .....	8-53
Outbound EDI Databank Transaction Queue.....	EDIQ091 .....	8-54
Reprocess Work File .....	EDIRW .....	8-56

## Chapter 9 Batch Partner Profile Maintenance

Overview .....	9-1
Before You Begin.....	9-3
Partner Profile Set Up.....	9-5
Batch Partner Processing.....	9-10
Header Record.....	9-16
Control Records.....	9-18
Group Records.....	9-67
Transaction Records .....	9-88
Name and Address Records.....	9-109
User Defined Data Records.....	9-119
Data Split Record .....	9-121
Error Rejection Record.....	9-124
CROSSREF Record.....	9-127
RELPART Record.....	9-129
Processing Reports .....	9-131

## Appendix A Global Parameters

Overview .....	A-1
Introduction to Global Parameters.....	A-2

## Appendix B Determining DASD Standards Space Requirement

Overview .....	B-1
Calculating DASD Requirements.....	B-2

## Glossary

Definition of Terms .....	G-1
Reserved Word Constants .....	G-11
Application Data Types .....	G-16
EDI Data Types .....	G-19

## Index

# Getting Started

## Overview

Welcome to Gentran:Basic® for zSeries Release 6.4!

Gentran:Basic provides you with the flexibility, efficiency, and thoroughness necessary to meet your most demanding requirements.

The *Gentran:Basic for zSeries Release 6.4 Technical Reference Guide* introduces the system programs, terminology, and functional requirements. Each chapter contains programs arranged by function. These functions include:

- Outbound Processing
- Inbound Processing
- System Utilities
- Databank Utilities
- Network Utilities
- Reporting Programs
- Databank Files
- Batch Partner File Maintenance

See the topic “System Overview” in Chapter 1 of the *Gentran:Basic for zSeries Release 6.4 User’s Guide* for introductory information on system interaction.

The Batch Partner Profile File Maintenance chapter contains a detailed description of batch maintenance performed on the Partner file by the EBDI005 program. The maintenance is detailed down to the record level for the entire partner profile.





# Inbound Processing Programs

## Overview

This chapter contains detailed descriptions of Gentran:Basic Inbound processing programs. Its purpose is to enable you to familiarize yourself with the system, its terminology, and functional requirements.

This chapter contains the following topics:

Topic	Page
<b>System Flow</b> .....	<b>2-3</b>
<b>Inbound Unwrap Processing EBDI010, EBDI010A</b> .....	<b>2-7</b>
<b>Inbound Editor EBDI001</b> .....	<b>2-9</b>
System-level Global Parameters .....	2-9
Runtime Global Parameters .....	2-11
Sequence of Processing.....	2-11
Runtime Global Parameter Processing .....	2-12
Pass-thru Record Processing.....	2-12
Compliance Checking to Standards .....	2-13
Error Rejection Processing.....	2-19
Error Rejection with the Databank.....	2-22
Error Rejection Numbers .....	2-22
Acknowledgment Generation .....	2-23
Determining Acknowledgment Type.....	2-25
Interchange Envelope Processing .....	2-27
Partner Relationship Usage.....	2-32
Partner X-Ref Usage .....	2-32
Group Envelope Processing.....	2-33
Transaction Envelope Processing .....	2-34
Package Structure Processing .....	2-35
Segment Processing .....	2-35
Trailer Envelope Processing .....	2-36
Databank Processing.....	2-36
Output Processing .....	2-37
Files Used.....	2-39
Inbound Editor Reports.....	2-46

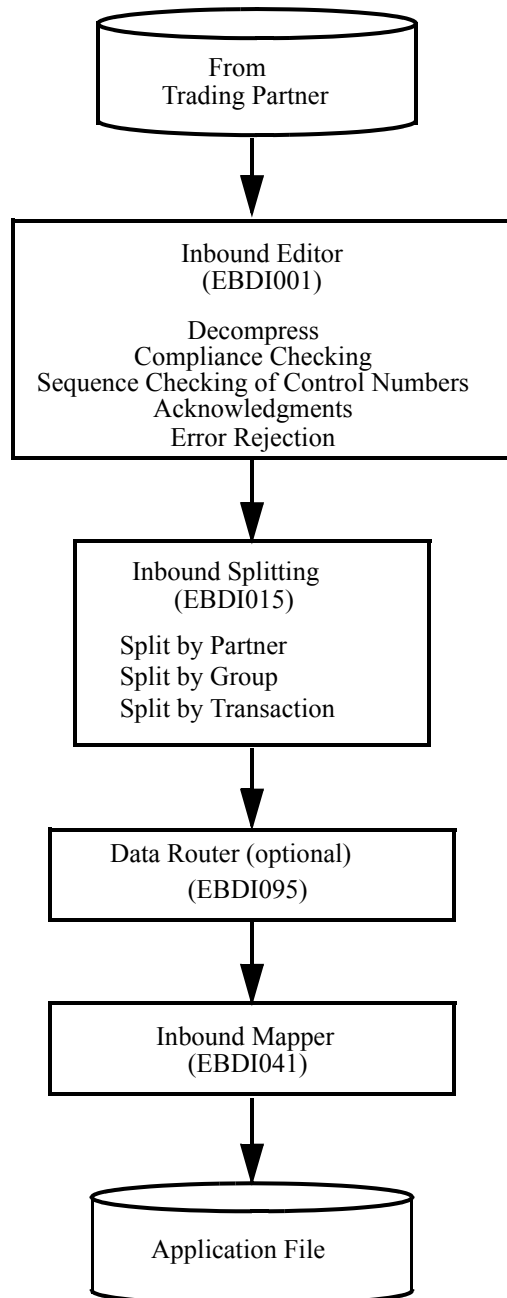
---

<b>Inbound Split EBDI015.....</b>	<b>2-55</b>
Inbound Split Processing .....	2-55
Logical Documents .....	2-56
Error Report .....	2-56
Files Used.....	2-56
Inbound Split Processing Parameters.....	2-59
Runtime System Log List .....	2-60
Miscellaneous Output Summary .....	2-61
<b>Inbound Data Mapping EBDI041 .....</b>	<b>2-62</b>
Processing Overview.....	2-62
Processing Detail – User (Receiver) Determination .....	2-63
Processing Detail – Partner (Sender) Determination .....	2-63
Processing Detail – Transaction Determination.....	2-63
Processing Detail – Sectioning .....	2-64
Processing Detail – Enveloping .....	2-65
Processing Option – Instream Parameters .....	2-66
Processing Option – Databanking.....	2-66
Processing Option – User Exits .....	2-67
Processing Option – EDI Data Print .....	2-69
Processing Option – Return Code Setting.....	2-70
Processing Option – NTE Segment Processing .....	2-70
Processing Option – Alternate Real Application Format.....	2-70
Processing Option – Viewpoint Processing .....	2-71
Processing Option – Verify Map Version.....	2-71
Files Used.....	2-72
Inbound Mapper Audit Trail Report .....	2-94

## System Flow

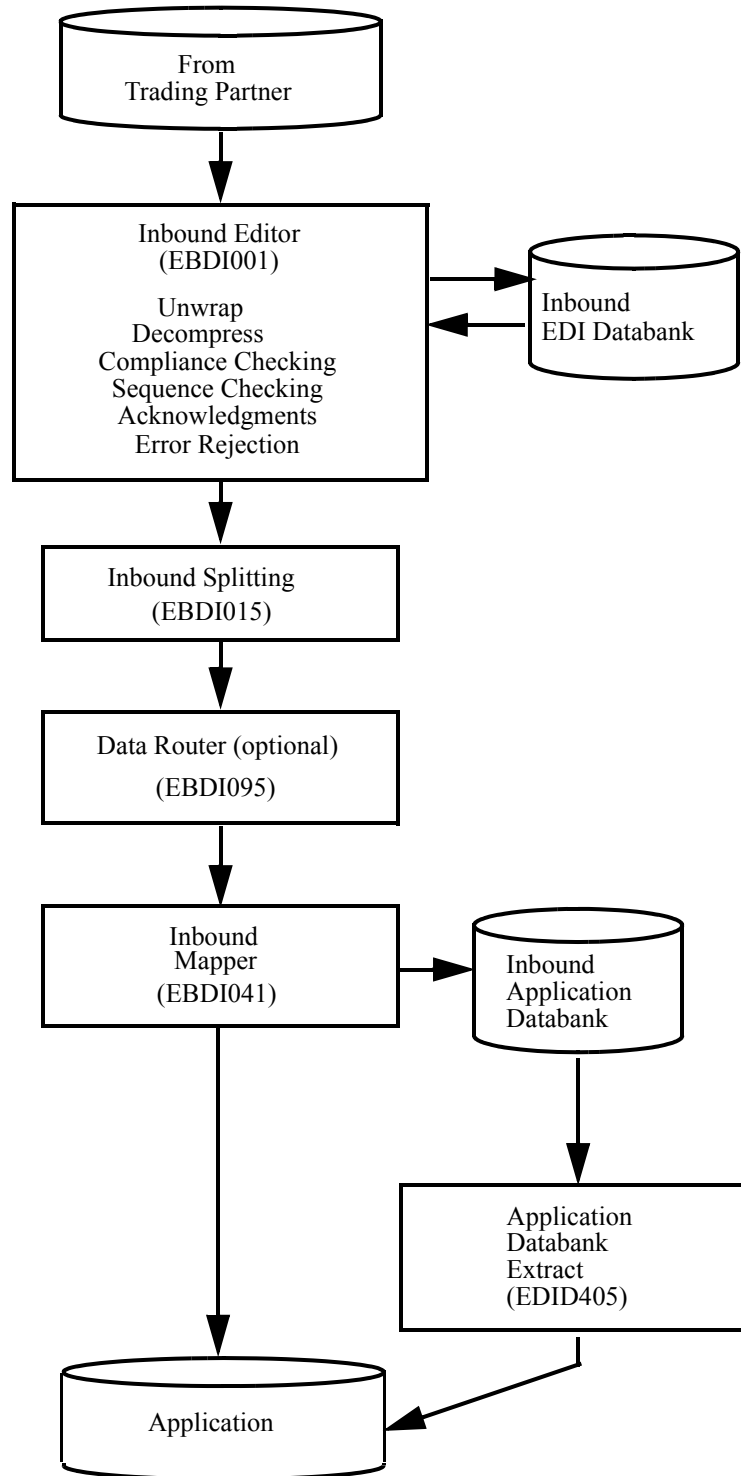
The section illustrates various system flows, depending on how your system is set up. Several variations of the Inbound processing exists. Descriptions of the programs are shown on the following flowcharts.

Figure 2.1 illustrates the inbound processing flow, with mapping, and with no databank message store. Use the optional programs according to your own processing needs. For detailed information on the programs illustrated, see the appropriate sections in this guide.



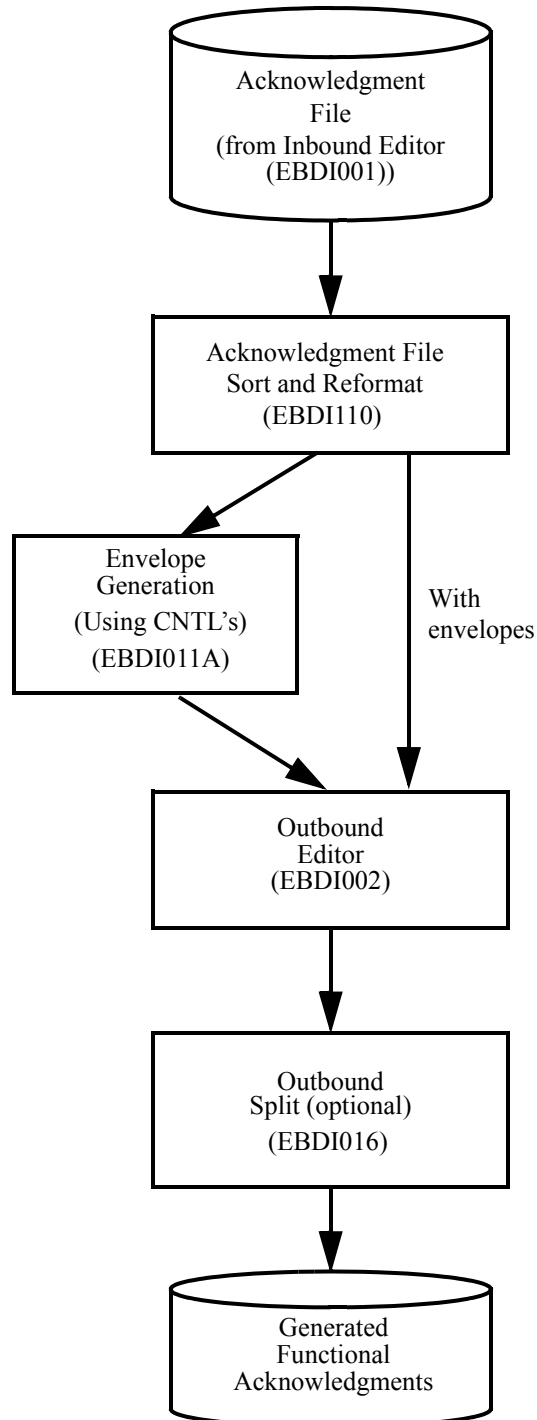
**Figure 2.1 Inbound Processing – With Mapping – No Databank Message Store**

Figure 2.2 illustrates the inbound processing flow, with mapping and with a databank message store. Use the optional program according to your own processing needs. For detailed information on the programs illustrated, see the appropriate sections in this guide.



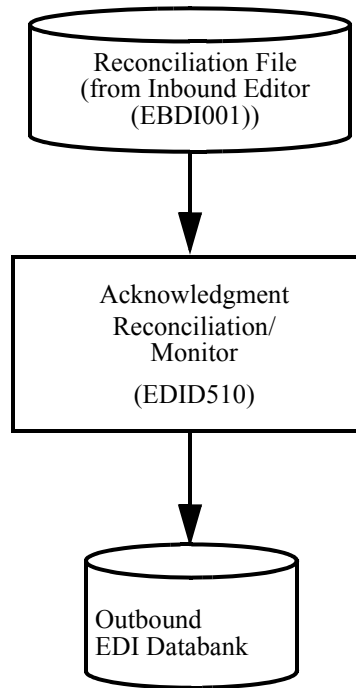
**Figure 2.2 Inbound Processing – With Mapping – With Databank Message Store**

Figure 2.3 illustrates the acknowledgment generation processing flow. Use the optional program according to your own processing needs. For detailed information on the programs illustrated below, see the appropriate sections in this guide.



**Figure 2.3 Acknowledgment Generation**

Figure 2.4 illustrates the acknowledgment reconciliation flow. For detailed information on the programs illustrated, see the appropriate sections in this guide.



**Figure 2.4 Acknowledgment Reconciliation**

## Inbound Unwrap Processing

## EBDI010, EBDI010A

The EBDI010 and EBDI010A programs make up the *unwrap* process for EDI data, which is normally compressed and *wrapped*. The wrapped data can result in multiple segments in a single record; thus, making it difficult to read. The unwrap process separates the data into one segment per record which will provide easier readability. This optional, stand-alone process can run as the first step in inbound processing.

You should only use this process when internal requirements deem it necessary for you to read the data in an unwrapped format.

You may not want to use this process when you are concatenating data sets into the unwrap process, and you have a history of receiving incomplete interchanges in your EDI data. Instead, you can concatenate a file containing a single record with the value `###NEWBATCH` in positions 1 through 11 of that record between each of the input data sets. This `###NEWBATCH` record will not enable EBDI010 or EBDI010A to properly unwrap all data that follows an incomplete interchange. However, it does enable the programs to correctly recover and unwrap the interchanges that follow the `###NEWBATCH` record.

There are two file size options for specifying the size limit for input records that the unwrap process will unwrap. You can:

- Use the EBDI010 program to limit the size of the files to 80-bytes.
- Use in the EBDI010A program to limit the size of the files to 512-bytes.

### Sample Input:

```
BG*COMMID*PASS*SENDER*RECEIVER*890425*1631*00017?GS*RO*GSENDER
*GRECEIVER*890425*1631*000000017*T*C2/6?ST*301*000000013?
```

### Sample Output:

```
BG*COMMID*PASS*RECEIVER*890425*1631*00017?
GS*RO*GSENDER*GRECEIVER*890425*1631*000000017*T*C2/6?
ST*301*000000013?
```

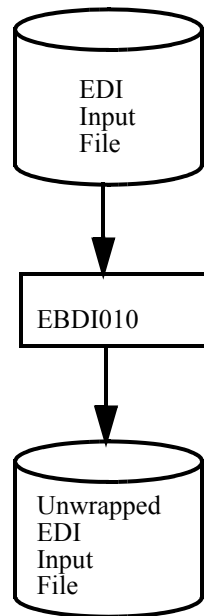
## Files Used

The following files are used by programs EBDI010 and EBDI010A:

Filename	Description
SYS040	Input File This file contains the input as received from your trading partner.
SYS050	Output File This file contains the data, unwrapped and ready for the Inbound Editor (EBDI001).

## Unwrap Process

Figure 2.5 illustrates the Unwrap process.



**Figure 2.5 EBDI010 — Unwrap Process**



---

## Inbound Editor

**EBDI001**

The inbound Editor always performs the following functions:

- data formats
- compliance checking data to standards
- sequence checking control numbers
- databank controls
- acknowledgments
- control records for data separation
- error rejection

By specifying global parameters and partner profile options, you can control how the Editor processes these functions. Global parameters and the partner profile both control your system's processing operation.

In addition to controlling how you want the Editor to validate data, global parameters also enable you to specify the level of detail to include in reports and the level of databank processing to be performed through two variations of global parameters: System-level and Runtime.

See Appendix A for individual global parameter definitions and record layouts.

This section provides details about global parameters and partner profile options that you can select to validate and process your data according to your requirements.

### System-level Global Parameters

System-level global parameters define to the Editors the features and options that you want to use.

#### *General Processing Options*

- ICS Tables
- Code Check
- Output Message
- Error Report

#### *Databank Processing Options*

- Partner Databank
- Databank Level

*Partner Processing Options*

- BG Partner
- Group Version
- Partner Sequence Check
- Verify Interchange
- Verify Group
- Verify Transaction
- Partner Version

*Partner Lookup Options*

- Partner XREF
- ISA Processing Sequence
- BG Processing Sequence
- GS Sender Lookup
- GS Receiver Lookup

*Receiver Processing Options*

- Verify Interchange
- Verify Group

*Rejection Processing*

- Error Rejection
- Error Rejection TRADACOMS

*Data Separation Options*

- Directed Output Files
- Split by Partner
- Transaction Test/Production

*Acknowledgment Options*

- Partner Acknowledgment
- Acknowledge Interchange
- Acknowledge Group
- Acknowledge Transaction
- Acknowledge Errors
- Use CNTL for Acknowledgments
- Generate TCR
- Generate A2

System-level global parameters exist in the Configuration file and are set up through the Administrative Main Menu, on the Global Parameter Maintenance system. These options represent the system configuration used by the majority of inbound runs; therefore, they rarely need changed.

See Appendix A for information on overriding System-level global parameters.

See Chapter 5 in the *Gentran:Basic for zSeries Release 6.4 User's Guide* for additional information on system-level global parameters.

## Runtime Global Parameters

Runtime global parameters provide instructions to the Editor at execution time and may vary from run to run. For previous Gentran:Basic users, global parameter records defined as Runtime are:

- All or Nothing
- Detail Report Always
- Use Subelement Separator for ANSI
- Use Subelement Separator for XXXXXXXXXXXXXXXbXX
- Generate 999 For XXXXXXXXXXXXXXXbXX
- Generate 997 For XXXXXXXXXXXXXXXbXX
- Version Conversion
- Output Databank Only
- Inbound EDI Databank Reprocess
- Implementation Convention Conversion

These global parameters are set up in the JCL as Instream parameters (SYS030) and include override parameters for System-level global parameters.

See Appendix A in this guide for additional runtime override information.

## Sequence of Processing

The inbound Editor, during its start-up and initialization process, processes all System-level and Runtime global parameters. If all of the values in the System-level global parameter record (located in the Configuration file) are not set up, the Editor uses its built-in default values for any missing values.

See Chapter 5 in the *Gentran:Basic for zSeries Release 6.4 User's Guide* for additional information on system-level global parameters, or Appendix A of this guide for default value information.

The sequence of parameter processing performed by the Editor is as follows:

1. **Default Values** – The Editor initializes parameters based on pre-determined values as shown in Appendix A in this guide.
2. **System-level Global Parameters** – The Editor overlays default settings with these values.

3. **Runtime Global Parameters** – The Editor loads internal tables with Runtime global parameters and overlays System-level global parameter values with its corresponding override parameter value, if present.

The Runtime global parameters are processed in the same sequence that they are read. If you have more than one Runtime override parameter for the same System-level global parameter, the last one read is used.

## Runtime Global Parameter Processing

Processing begins with the global parameters. These parameters are positional (pre-described format) and are ignored by the Editor if they start (position 1) with a space, ‘X,’ or ‘\*.’ Otherwise, they are validated by the Editor and produce an error message (Error Number 558) on the Error Report should an error be detected. The global parameters provide instructions to the Editor and the error message is only a warning that the Editor did not understand some of these instructions.

Valid Runtime global parameters are used to update the Editor’s internal decision tables. In this section, runtime global parameters are included where they are used in the actual processing.

See Appendix A for an overview of parameter features, options, and default values.

## Pass-thru Record Processing

The ‘###’ indicates a pass-thru record and is written to the output file without being validated. This allows nonstandard information to be sent or received by Gentran:Basic.

### *###NEWBATCH Pass-Thru Record*

If multiple batches of data are concatenated as input in the Inbound Editor and there is a history of these batches containing incomplete interchanges, the ###NEWBATCH record can be passed to EBDI001 between each of the concatenated files. This helps EBDI001 recognize that a new batch of data has started. The ###NEWBATCH record can be used by concatenating a file containing a single record with the value ###NEWBATCH in positions 1 through 11 between each of the files containing the input EDI data. The ###NEWBATCH records are not written to the output file.

### *TRADACOMS Documents*

ANA TRADACOMS documents are organized differently than ASC X12 documents. In general, most documents are composed of Header, Detail, and Summary sections. For an ASC X12 document, these three sections are contained within a single transaction. TRADACOMS documents are different; TRADACOMS Header, Detail, and Summary sections are in separate transactions (messages).

Gentran:Basic has added the PREVIOUS and NEXT transaction/release level from the Standards Subsystem to the Batch Optimized table file. This change enables the Inbound Editor to treat the individual TRADACOMS messages that comprise a single document as one logical document. How logical documents affect error rejection and data separation is explained later in this section.

See the topics “Error Rejection Processing” and “Output Processing” later in this section for further information.

Compliance testing that is unique to logical documents includes checking for missing messages within a logical document and checking for an incorrect message release level.

## Compliance Checking to Standards

### *Delimiters*

Delimiters control and separate individual EDI segments and elements. Interchange Control Headers (ISA, ICS, BG, UNB/UNA, and STX/SCH) set delimiters or default delimiters to pre-determined values. Once the delimiters are set, each functional group, transaction, and data segment within the interchange can be translated. Inbound, the interchange control Header sets delimiters for each interchange. Outbound, the partner profile sets delimiters for each interchange. Each delimiter has a unique value within the interchange. Gentran:Basic uses the following delimiters:

- **Segment Separator**  
ANA/TRADACOMS only, this is a special delimiter that delimits the segment identification from the elements. The default value is = (equal sign).
- **Element Separator**  
All interchanges use element separators to delimit individual elements. Each element can be either a simple element or composite element (containing subelements).
- **Subelement Separator**  
Component Elements or subelements are supported within Gentran:Basic. The subelement separator is an additional delimiter that separates the component pieces of a composite or group element. A group element is collection of elements that are functionally connected. The subelement is the component piece. A composite element defines a relation of individual subelements.
- **Repetition Separator**  
Repeating data elements and repeating data structures are supported within Gentran:Basic. The repetition separator is an additional delimiter that separates each occurrence of a repeating data element (simple element) and/or data structure (composite element).
- **Release Character**  
For EDIFACT and TRADACOMS, string element types (AN) or ID element types can use release character delimiters. Release character delimiters indicate that the character following the release character is part of the string and not a defined delimiter (element separator, segment separator, subelement separator, or segment terminator). Release characters are not included in maximum field lengths.  
  
See the sub-topic “Element Compliance” later in this section for more details on string and ID element types.
- **Decimal Notation**  
For EDIFACT and TRADACOMS, a decimal point can have a period (.) or comma (,) as a value. For ANSI/X12, the default value a decimal point is a period (.).
- **Segment Terminator**  
All interchanges use a segment terminator to signal the end of EDI segments.

## Standard Version Selection

The Gentran:Basic Editors use information contained in the envelopes to identify which standard version and version agency to use for compliance editing. This occurs at the interchange, group, and transaction level, depending on the type of interchange.

Standard versions and transactions that are used in compliance editing are pre-selected online, through the Standards Maintenance Menu – Transaction in Use screen. The selected versions and transaction are downloaded through the Selective Download Process (EBDI030 program). The optimized table is the output of this process and is used by the Editors to compliance edit the data.

Gentran:Basic builds a 12-byte Standard Version Identifier from the envelopes as follows:

### Envelopes

ISA12	==>	Interchange Standard Version Identifier
STX0101,STX0102	==>	Standard Version Identifier
ICS03	==>	Interchange Standard Version Identifier
GS08	==>	Standard Version Identifier
UNG0501, UNG0502, UNG0503	==>	Standard Version Identifier
UNH0201, UNH0202, UNH0204	==>	Standard Version Identifier

The Standard and Version Identifier is defined as follows:

Version Number	Version Release	Version Association
Bytes 1 – 3	Bytes 4 – 6	Bytes 7 – 12

Because various standards committees use the same version, the agency code is now required to differentiate between versions. The envelope agency is used with the version to select a standard unless a Version Association is present. This association is checked against the EDISASC Standards Association file. The file is a list of association/agency relationships defined to Gentran:Basic. If a version association is detected by the Editors, the association identifies the corresponding agency. This information is provided to the batch table optimization process.

Gentran:Basic uses the Standard Version Agency from the envelopes as follows:

### Envelopes

GS07	==>	Standard Version Agency
UNG04	==>	Standard Version Agency
UNH0203	==>	Standard Version Agency

For more information, see the Standards Subsystem chapter in the *User's Guide* or in the Utility Programs chapter in the *Technical Reference Guide*.

After identifying and loading a control syntax or a user transaction document, the Editor begins compliance checking. Gentran:Basic validates each detected transaction, segment, and element to the supported standard. This ensures absolute adherence to the supported EDI Standards and that each EDI Standards requirement is properly implemented. The Editor validates segment compliance, element compliance, and code list compliance.

## Segment Compliance

The segment compliance validates each non-suspended EDI segment. The Editor scans all input EDI data. Segments are checked for placement within the identified transaction. *Gentran:Basic* validates segments that start with either a two- or three-byte segment identification. Repeated segments within the identified transaction are checked for maximum usage and looping. *Gentran:Basic* verifies mandatory segments for active loops. Segments are classified as control segments or data segments based on the following criteria:

- Any segment that is defined to the identified transaction and is positioned between the transaction control Header segment and transaction control Trailer segment is considered a data segment.
- Any segment that is not included within an identified EDI transaction is considered either a control segment or invalid data segment.
- Non-EDI data is considered anything prior to the interchange Header and after interchange Trailer. Interconnect information and network specific records are not processed by *Gentran:Basic*. The Inbound Editor suspends and bypasses this information. Passthru '###' records, \$TCR, and \$\$ADD records are not considered EDI data and are restricted to passing information between processes.

Each Editor checks Audit ID indicators, conditional codes, and segment activities during segment and element compliance editing. Auditing User fields are extracted and processed after each element.

See the topics “Online Standards Maintenance” and “Segment Element Maintenance” in Chapter 4 for additional information on Audit ID indicators.

Conditional codes and segment activity requirements are processed after each element is validated.

See Chapter 5 in the *Gentran:Basic for zSeries Release 6.4 User's Guide* for information on Online Standards Maintenance, Conditional Code Maintenance, or Segment Activity Maintenance.

After a segment has been found within the identified transaction, segment elements, element dictionaries, and optional code lists validate individual elements. Each element is passed through element compliance.

## Element Compliance

The Editor uses the element types defined in each dictionary entry within the batch optimized tables to edit each element. *Gentran:Basic* supports an 18-digit floating point number for internal accumulators and hash totals. The Editor divides element types into six major classifications. The following section describes each classification.

### Date and Time Types

DT	- YYYYMMDD	- Standard Date — fixed 8 bytes — 003072 — X12
	- YYMMDD	- Standard Date — fixed 6 bytes — pre 003072 — X12
D8	- YYYYMMDD	- Standard Date — fixed 8 bytes — EDIFACT
CD	- MMDDYYYY	- Alternate Format — fixed 8 bytes — international
TM	- HHMM	- Standard Time — fixed four bytes — X12

T6	- HHMMSS	- Standard Time w/option seconds
T8	- HHMMSSHS	- Standard Time w/option seconds/hundredths

### Real Decimal Types

R0 through R9, and R – Maximum decimal precision is 9 digits to right of decimal. Real number usage, decimal point, and sign are considered in length of the element. The element type validates decimal point placement. Hex characters '00' or Low values are invalid.

### Binary Type

B – Code types are supported but cannot be edited because of their character interpretation.

### Implied Decimal Types

N0 through N9 – Maximum decimal precision is 9 digits to right of implied decimal. The element type determines decimal pointer placement. Hex characters '00' or Low values are invalid.

### String Types

AN – String elements can be supported. Special characters, spaces, period, comma, uppercase/lower case letters, and numeric characters can be contained in a String type. String type can support code list up to 40 bytes in length. Code verification is active for String types.

### ID Types

ID – Code types are a special String type. ID types contain all numeric characters or all letters. ID types can support code lists up to 40 bytes in length. Code verification is active for ID types.

### Code List Compliance

Code checking is optional. Code list compliance checking can be configured through the Administrative Main Menu on the Global Parameter Maintenance subsystem.

Online code files are available to the Editors. The Editor determines which code file to use based on the maximum length (defined by the standard) of the element detected.

File	Minimum Length	Maximum Length
EDISCD1	1	5
EDISCD2	6	10
EDISCD3	11	20
EDISCD4	21	40

The Editors keep an internal Smart Code List of valid codes. This code list is a stack of up to 100 code values and provides a buffer of code values for the Editors to use during element validation within each transaction. The stack is arranged last-in-first-out (LIFO), which means that when a new valid code is placed on the stack, the last valid code drops off the stack. The code file is checked only when the code value is not on the Smart Code List.



## Simple Codes

Simple Codes are the most common type of codes. With Simple Codes, strings of characters are used and validated against one of four online standards code files. Simple Codes are validated individually.

## Composite Codes

Composite Codes have multiple parts to the detected code value. The Composite Code Definition is used to substring each code value. Each part is validated against one of the four code files. If any part of the code is invalid, the complete composite code is invalid.

When using compliance checking, you must consider:

- What should happen if the Editor detects an error.
- Whether you need to send an acknowledgment back to the trading partner. If so, does the trading partner want Details (transaction) or Summary (group) acknowledgments?
- Whether to receive documents in sequential order.

**Note:** Receiving documents in sequence, helps to determine if any documents are missing.

In the following sections, these and other considerations have been functionally grouped with other information on which global parameters to use. Error rejection is only flagged with ###REJ records in the Editor. The Error Rejection file is created in the Inbound Split Program EBDI015.

**Note:** Error rejection causes the counts on the Trailer records to be incorrect. If you create an error rejection record for incorrect Trailer counts, you generate another set of ###REJ records. If you set up error rejection records for envelope Headers (ISA, ICS, BG, GS, UNB) and a suspense level error occurs, Gentran:Basic creates an output file with one envelope record surrounded by ###REJ records.

## Error Handling

When compliance checking a document, one of the following results can occur:

- Document is clean – no errors
- Document is suspended
- Document has errors

Each error message has an associated return code. Error messages and additional user information are associated with errors generated by the Editors. The Inbound and Outbound Editors have separate return codes, which provide control over error processing.

You can use the Print flag in Error Message Maintenance to specify whether to suppress error messages or modify return code values.

See Chapter 5 in the *Gentran:Basic for zSeries Release 6.4 User's Guide* for additional information about Message Maintenance.

**Document Is Clean**

A clean document is a document with no errors.

**Document Is Suspended**

Suspension automatically occurs if:

- The standards version selected is not on your table file.
- The interchange envelope is missing key information such as Sender ID, a separator, or a terminator.
- The segment contains data that the Editor cannot interpret.

Suspension is not always automatic; you can control suspension through global parameters. If the VERIFY global parameters are active and the partner profile is not found, the Editor suspends. If the VERIFY global parameters are active and the partner profile is found but the receive flag on the group or transaction record is not set to **Y**, the Editor suspends. If the global parameters, Partner Version Always Interchange or Partner Version Always Group, are active and the partner version on the group/transaction record is either spaces or not on the table file, the Editor suspends.

**TRADACOMS Logical Documents**

The VERIFY global parameters require each transaction (message) that comprises a logical document to be on the partner profile. Otherwise, suspension occurs. If suspension occurs on the Header message, the entire logical document is suspended.

**Document Has Errors**

Gentran:Basic considers all compliance errors as warnings unless you indicate otherwise. Gentran:Basic recognizes two types of errors: compliance, and Gentran:Basic (non-compliance related). Currently, the error handling process treats both types equally.

**Error Message File**

Gentran:Basic assigns all errors a numeric value. These numeric values, as well as an error description are maintained on the Error Message file. Along with the description, each error message record has two, 1-byte flags for error rejection processing. There is one flag for Inbound and one flag for Outbound representing the interchange, group, and transaction levels of the document. As shipped, both flags are set with spaces. You update the flags on the error message file through the error rejection screen on the online Partner subsystem.

See the topic “Error Rejection Screen (EDIM055)” Chapter 2 in the *Gentran:Basic for zSeries Release 6.4 User’s Guide* for additional information.

See the sub-topic “Error Rejection Numbers” on page page 22 for a complete list of error numbers used by Gentran:Basic for inbound processing.

**Partner Sequence Error****Incremental Definition**

The Editor checks the control numbers on the input against the partner profile. If an interchange, group, or transaction control number exists, Gentran:Basic compares the input control number to the one on file. If the input control number is not one more than the control number on the partner profile, an error message is generated. Otherwise, the control number is updated on the partner profile with the control number from the input. If the partner record does not exist, or is out of sequence, an error message is generated.

**Chronological Definition**

The Editor checks the control numbers on the input against the partner profile. If an interchange, group, or transaction control number exists, Gentran:Basic compares the input control number to the one on file. The control number can only be greater than the control number on the partner profile or an error message is generated. Otherwise, the control number is updated on the partner profile with the control number from the input. If the partner record does not exist, or is out of sequence, an error message is generated.

**Error Rejection Processing**

When using the Gentran:Basic Error Rejection feature, the Inbound Editor uses information from the global parameters file, the partner profile, and the Error Messages file to determine what action to take when errors occur.

**Global Parameters File**

In the System-level global parameters, the Partner Error Control flag must be set to **Y** (Yes). This tells Gentran:Basic that Error Rejection is being used. The system, in turn, uses the Error Message file and the partner profile to determine what actions to perform, if any.

**Partner Profile**

In the partner profile, error rejection records must be coded. To reject inbound data, either:

- create error rejection records for the Sender ID of the interchange or group being received

**OR**

- create error rejection records for the Global Partner  
(!!!GENTRAN-RESERVED-PARTNER- ID-1 for the Partner/Qualifier Mode  
or !!!GENTRAN-RU1/!!!GENTRAN-RP1 for the Relationship Mode)

The information in the partner Error Rejection record indicates the level (i.e., interchange, group, or transaction), the specific error number, and whether this error is to be rejected inbound, outbound, or both.

**Partner IDs for Error Rejection**

Your first choice when setting up Error Rejection is for ALL partners. This is accomplished through the Administrative Main Menu, on the Error Rejection (EDIM055) screen. For the selected error number, enter **A** and update. For example, if you set up error rejection as **A** (Always) for error number 0062 (missing mandatory element), and reject Inbound, the system rejects any transaction set whenever an error 0062 occurs, regardless which Partner ID is coded on the group.

Individual Partner IDs enable you to control Error Rejection by partner. For example, when error 0062 occurs, the system reads the Partner ID in the error record to determine whether the transaction should be rejected.

Your second choice is to set up rejection on the global partner. This allows you to control error rejection at a global level. For example, when error 0062 occurs, the system reads the global partner error records to determine whether the transaction should be rejected. The global partner can be set up to reject all transactions/groups for a particular error or to reject specific transactions/groups for all partners.

Lastly, you can choose to set up rejection by individual Partner IDs. This allows you to control Error Rejection by trading partner. For example, when error 0062 occurs, the system reads the Partner ID in the error record to determine whether the transaction should be rejected.

### **TRADACOMS Logical Documents**

When setting up error rejection records on the partner profile for a TRADACOMS partner, use the Header transaction (message). The Header message is always used by the Editor when looking up an error number on the partner profile even if the error occurs in the Detail or Summary messages.

**Note:** You cannot set up Error Rejection online as A AND by global or individual partner. The system will block the addition of partner rejection records.

## ***Levels of Error Rejection***

You can control Error Rejection at three levels for Inbound and Outbound processing.

### **Partner Level (Interchange)**

This level specifically refers to errors at the interchange Header/Trailer level. This enables you to control errors that occur ON the interchange Header/Trailer segments.

### **Group Level**

Group ID is the key to the partner profile for this level, specifically referring to errors at the group Header/Trailer level. It allows the user to control errors that occur ON the group Header or Trailer segments. When using the Group level you have the option of coding this record for a specific group (i.e., purchase orders, invoices) or for all Groups. To code the partner error record for a specific group, enter the specific Functional Group ID (i.e., PO, IV) into the Level ID field. To code the partner error record for all groups, code the Level ID field with 'ALL'. The effect of coding 'ALL' causes the system to perform the indicated action (reject inbound) no matter what type of group Header Functional ID is in the inbound data. The coding of specific group Functional IDs allows the customer to control error rejection on a group by group basis.

### **Transaction Level**

Transaction Set ID is the key to the partner profile for this level, specifically referring to errors at the transaction set level. It allows you to control errors that occur on any segments WITHIN the transaction Header and Trailer segments. When using the Transaction level you have the option of coding this record for a specific transaction (i.e., 875, 880), or for all transactions. To code the partner error record for a specific transaction, enter the Transaction ID into the Level ID field. To code the partner error record for all transactions, code the Level ID field with 'ALL'. The effect of coding 'ALL' causes the system to perform the indicated action (reject inbound) no matter what type of Transaction Set ID is in the inbound or outbound data. The coding of specific Transaction Set IDs allows the customer to control error rejection on a transaction by transaction basis.

## ***Error Numbers in Error Rejection***

Error types that can be rejected are determined by their identifying number. Each error is described in the *Gentran:Basic for zSeries Release 6.4 System Message Guide*. When the system is processing Inbound or Outbound data and an error occurs, the action taken when using Error Rejection is determined by whether that error number has a corresponding partner error record. If a partner error record exists for this error number, the action specified in the error record is taken (reject Inbound). Each error that you want to control must have a corresponding partner error record. By default, the Gentran:Basic system treats errors as warnings.

### *Direction Control in Error Rejection*

The system provides two fields when determining the action for each specific error: Inbound Rejection and/or Outbound Rejection.

Gentran:Basic enables you to control each error number and corresponding action separately for Inbound and Outbound processing. It also enables you to set up an error number and corresponding action for both Inbound and Outbound processing on the same error record.

The Inbound Rejection field tells Gentran:Basic whether to reject this error on incoming data. The possible values are **Y** (Yes) or **N** (No).

The Outbound Rejection field tells Gentran:Basic whether to reject this error on incoming data. The possible values are **Y** (Yes) or **N** (No).

### *Error Message File*

The Error Message file contains all error messages issued during the Inbound and Outbound Editor processing. The Error Message file contains each error number, the error message and two special fields, Rejection Decision Inbound and Rejection Decision Outbound, used by the Error Rejection feature.

#### **Rejection Decision Inbound**

The Rejection Decision Inbound field on the error message record is used by the Inbound Editor (EBDI001) when this error occurs during inbound processing.

#### **Rejection Decision Outbound**

The Rejection Decision Outbound field on the error message record is used by the Outbound Editor (EBDI002) when this error occurs during Outbound processing.

### **Inbound and Outbound Rejection Decision Values**

For the Inbound and Outbound rejection decision fields, two possible values impact how the Editors proceed.

- A – The value **A** causes the system to reject this interchange regardless of what is coded on the partner profile. This value is assigned using the online Error Message and Control Subsystem.
- P – The value **P** tells the system that this error is coded for a specific partner. For Inbound processing, the system reads the partner profile error records based on the incoming Sender ID to determine the action. For Outbound processing, the system reads the partner profile error records based on the incoming Receiver ID to determine the action. The value **P** is assigned when Error Rejection is set up for a specific Partner ID at the partner level and rejection Inbound or Outbound is coded **Y**.
- D – A value **D** tells the system that this error is coded for the global partner (!!!GENTRAN-RESERVED-PARTNER). The system reads the partner profile for the global partner to determine the action. The value of **D** is assigned when Error Rejection is set up at the global partner level and rejection Inbound or Outbound is coded **Y**.

## Error Rejection with the Databank

### All or Nothing

When activated, this Runtime global parameter enables Gentran to provide only valid, error-free data to the databank. The parameter monitors all interchanges and prevents and stores data with errors from passing through to the trading partner. When an error occurs, all databank directory and message store information for this run is written “interrupted.” After correcting the errors, either by the inbound trading partner or through the outbound application, Gentran releases the data for translation.

Errors produced by the Editor and described in the System Message Guide activate the interrupt action, while continuing the current run. After the current run is complete, two events occur:

1. Directory and message store updates that have been made to the databank are indicated as interrupted databank information.
2. The external user return code equals 0016.

You can use the databank “housekeeping” feature in Gentran:Basic to remove this interrupted information.

See the topics “Outbound EDI Databank Maintenance (EDID201)” and “Inbound EDI Databank Maintenance” sections (EDID301) for more information.

**Note:** When using the ALL or NOTHING parameter, you are instructing Gentran to accept only (inbound) and/or send (outbound) EDI data which is totally error-free and compliant with standards or implementation convention.

## Error Rejection Numbers

This section will assist you in setting up Error Rejection. The section lists the error numbers used by Gentran:Basic for Inbound processing. It is up to YOU to choose which errors are important to your processing.

### *Error Rejection Numbers – Inbound*

This section describes Inbound Error Rejection numbers.

### Error Numbers at the Partner Level

Error numbers at the Partner level are:

ICS	ISA	BG	GS	EDIFACT	TRADACOMS
	21	21	21	21	21
21	22	22	22	22	22
22	78	142	207	142	147
23	142	207	218	207	207
24	207	218	532	534	
25		219	533	535	
26		220	534	538	
29		380	535	568	
40			536		

41	538
42	539
43	568
44	
78	
142	
155	
207	
221	

**Error Numbers for all Interchanges**

Error numbers for all Interchanges are:

45,46,67,183,354,355,356

**Date and Time Error Numbers**

Date and Time error numbers are:

32,34,35,36,37,38,39,83,33

**Error Numbers at the Group Level**

Error numbers at the Group level are:

47,48,49,55,67,141,195, 562,222

**Error Numbers at the Transaction Level**

Error numbers at the Transaction level are:

50,51,52,53,54,57,60,61,62,63,64,65,79,80,81,82,  
84,85,86,87,88,89,90,91,92,93,131,140,145,146,147,326,  
327,385,386,387

**Databank Error Numbers**

Databank Error numbers are:

689,693,695,696,697,698,699

**Acknowledgment Generation**

Gentran:Basic can automatically generate acknowledgments for ASC X12, UCS/WINS, TDCC, and EDIFACT Standards. Three types of acknowledgments are possible: CONTRL, 997, and 999. The decision to generate acknowledgments is under your control through the System-level and Runtime global parameters.

System-level global parameters that apply to all trading partners are:

- ACKNOWLEDGE INTERCHANGE ON
- ACKNOWLEDGE GROUP ON
- ACKNOWLEDGE TRANSACTION ON

These global parameters indicate to the Editor the level of acknowledgment to be generated for all trading partners. If you want compliance errors to be identified and sent to your trading partner, you must activate the System-level global parameter, ACKNOWLEDGE ERRORS ON.

If you do not want to send acknowledgments for all transactions received or only to selected trading partners, acknowledgment generation must be controlled at the trading partner level. This is accomplished by activating the System-level global parameter, PARTNER ACKNOWLEDGMENT.

If you deactivate the System-level global acknowledgment parameters above, by placing overrides for the respective System-level global parameter flags, the control for generating acknowledgments is passed to the partner profile. For example:

```
ACKNOWLEDGE INTERCHANGE OFF
ACKNOWLEDGE GROUP OFF
ACKNOWLEDGE TRANSACTION OFF
ACKNOWLEDGE ERRORS OFF
PARTNER ACKNOWLEDGEMENT
```

Partner level acknowledgment control gives you the most flexibility in determining the generation of acknowledgment and their type. The Partner Control record contains the acknowledgment flags for interchange and errors. The Partner Group Information record indicates whether a group level or a transaction level acknowledgment is desired. If the generate group or transaction field is spaces, no acknowledgment is generated for that specific group.

Gentran:Basic provides additional functionality for transaction level acknowledgments. This only applies if partner level acknowledgment is active. In the Acknowledge this Transaction field on the Transaction Information screen, you can specify **Y** (Yes) or **N** (No). If you select No, the Editor will not generate a transaction level acknowledgment (AK2/AK5 or UCM) for this partner and transaction. If this is the only transaction received in a group from your trading partner, no acknowledgment will be generated at the group level either.

**Note:** The System-level global parameters ACKNOWLEDGE GROUP and ACKNOWLEDGE TRANSACTION override the partner.



## Determining Acknowledgment Type

### *EDIFACT Standards*

For EDIFACT Standards, the following versions of CONTRL are supported:

UNB Element 0101	Version of CONTRL Generated
1	002001, UN, CONTRL
2	002002, UN, CONTRL
3	<del>Dbb3bb</del> , UN, CONTRL
4	<del>4bb1bb</del> , UN, CONTRL

The CONTRL acknowledgment is invoked using the ACKNOWLEDGE System-level global parameters and/or through the partner profile by using the PARTNER ACKNOWLEDGEMENT System-level global parameter.

### *ASC X12 Standards*

Gentran:Basic provides two different methods of selecting either a 997 or 999 acknowledgment. The first method is invoked by using the System-level global parameter PARTNER ACKNOWLEDGEMENT. This parameter indicates to the Editor that the Transaction Acknowledgment Type field from the Partner Transaction Information record is initially used to determine acknowledgment type. If this field is spaces (not used), the Editor checks to see if the runtime global parameter GENERATE 999 FOR (version) is active. If the group version is found, a 999 type of acknowledgment is generated, otherwise the Editor defaults to 997 type of acknowledgment. In this method, the initial control for acknowledgment type is transaction driven.

The other method for determining the type of acknowledgment to generate is basically the same as the first, except the transaction record is not used. The default in Gentran:Basic for acknowledgment generation is 997; the 999 type of acknowledgment requires the Runtime global parameter GENERATE 999 FOR (version) to be active.

### *Generating an Optional A2 segment*

Gentran:Basic provides a method of generating the optional A2 acknowledgment segment for 999 types of acknowledgments. This is accomplished by activating the System-level global parameter GENERATE A2 ACCEPTANCE. Without this global parameter, no A2 acceptance segments are generated.

### *Using TCR for Acknowledgments*

The Runtime global parameter USE TCR FOR ACKNOWLEDGEMENTS generates the interchange sender's transmission control record (TCR) information and includes it with the Outbound acknowledgment interchange. All \$TCR information passes through the Outbound Editor, which removes the \$TCR Pass-Thru ID and leaves the free-form, 80-byte information at the beginning of the interchange.

**Note:** Do not select the USE TCR FOR ACKNOWLEDGEMENTS in conjunction with the USE CNTL FOR ACKNOWLEDGEMENTS global parameter.

All transmission control record information is retrieved for \$TCR user-defined user records. To use this parameter, the interchange sender's profile must contain the \$TCR free-form information.

See the topic "User Defined Screen" in Chapter 2 in the *Gentran:Basic for zSeries Release 6.4 User's Guide* for additional information.

**Note:** With OUTPUT DATABANK ONLY, the \$TCR free-form, 80-byte information is only placed on the Outbound EDI Databank Message Store (EDIOES).

### *Creating the Acknowledgment File*

The two formats that can be used to create your Acknowledgment file are:

1. Envelope method
2. CNTL method

#### **Envelope Method**

In the envelope method of creating acknowledgments, the inbound Editor builds both the interchange/group/transaction envelopes and the acknowledgment segments. The input Sender and Receiver ID's are reversed for the acknowledgment envelopes. For 997 acknowledgments, the version can be changed on the acknowledgment document from the input version received. To accomplish this, add the FA Functional Group ID under the input Sender ID on the partner profile. In the outbound (GS08) version and controlling agency field (GS07), place the version you want for the acknowledgment envelope. The Editor checks for this Partner record and substitutes this version on the group envelope when it builds the acknowledgment document.

**Note:** For 999 acknowledgments, which are only valid with the old TDCC Standards, the first character of the version is replaced with G for General version. For example, U3/0 would become G3/0 on the 999 acknowledgment.

To properly assign error codes to the acknowledgment documents (in particular, TA1, AC1, and CONTRL), the entire document must be processed prior to their generation. However, the sequence of generation does not agree with the standard segment sequence. Before the outbound Editor can process the acknowledgment file, it must be resequenced to meet EDI standard requirements.

The resequencing is accomplished by adding a sort key to the acknowledgments when they are created by the Editor. This Acknowledgment file containing a sort sequence key is then input to program EBDI110, which sorts the records and strips the sequence key from the acknowledgments. The resulting file is ready for outbound processing.

#### **CNTL Method**

An alternate method of creating acknowledgments is the CNTL method. In this method, the Editor creates control records (CNTL records using the resolved Sender ID) rather than the actual interchange/group/transaction envelopes. A sort key is added to the acknowledgments before they are processed by the Sort/Reformat program EBDI110. The resulting Acknowledgment file goes through the Envelope Generation program (EBDI011A) in the Outbound flow. This method is very flexible, because you can use any envelope type you want to send back to your trading partner.

With this method, you can use UCS/RAIL/Motor control numbering, change version numbers, and also have a different sender/receiver than the one you originally received. To create this method of acknowledgment generation, activate the global parameter USE CNTL FOR ACKNOWLEDGMENTS.

The element separator, subelement separator, and terminator from the inbound document are passed on the CNTL record from the inbound Editor to the Acknowledgment Envelope Generation program (EBDI011A). This ensures that the delimiters used in the outgoing acknowledgment are consistent between the envelopes and data segments.

### *Creating “RAIL” Functional Acknowledgments*

When an inbound ISA interchange envelope is compliance-checked, the Inbound Editor program (EBDI001) examines the content of the ISA01 for 04 and the first two characters of the ISA02 for SW. If an acknowledgment is requested, the presence of these two values in the ISA envelope triggers the Editor to generate a Rail functional acknowledgment. Both methods of creating acknowledgments outlined in “Creating the Acknowledgment File” on page 26 support the Rail functional acknowledgment.

### **Interchange Envelope Processing**

Interchange envelopes must have a Segment ID of BG, ISA, ICS, GS, UNA, SCH, STX, or UNB. Anything else causes the input records to be written to the suspense file until a valid interchange is found. Also, an error report is generated.

See the program description for EBDI030 for additional information on standard versions.

### *Compliance Checking Edit Criteria*

Each element of the interchange is validated against a selected standard. Here are the primary edits for an interchange:

1. Is the element Mandatory or Optional? If mandatory, is it present? If not, data is suspended.
2. Is element within the minimum/maximum field size established by the standard? If not, data is suspended.
3. Is the element in proper format (CD, DT, D8, TM, T6,T8,R, AN, Nx, Rx, ID)? If not, data is suspended.
4. If applicable, does it contain a proper value as found in the Code record?
5. What global parameters are in effect?

### *Standards Version Selection*

Compliance Checking begins with selecting the standards version to be used from the table file.

#### **BG Interchange**

Since the BG interchange contains no version, the Editor uses a default version for compliance checking.

**ISA Interchange**

Segment element ISA12 contains the version number to be used to validate this interchange. If this version is not found on the table file, the Editor checks the VERSION CONVERSION records from the global parameters. If a match is found, the alternate version from this parameter record is used to obtain the correct version of the standards from the table file. Global parameter, PARTNER VERSION (ALWAYS) INTERCHANGE, instructs the Editor to check the partner profile before going to the Version Conversion.

*See Partner Version (Always) Interchange.*

**ICS Interchange**

Segment element ICS04 contains the version number used to validate this interchange. If this version is not found on the table file, the Editor checks the VERSION CONVERSION records from the global parameters. If a match is found, the alternate version from this parameter record is used to obtain the correct version of the standards from the table file. Global parameter, PARTNER VERSION (ALWAYS) INTERCHANGE, instructs the Editor to check the partner profile before going to the version conversion.

*See Partner Version (Always) Interchange.*

**GS Interchange**

Normally a Group envelope Segment ID, under TDCC, the GS envelope may be used as both an Interchange and a Group envelope. Segment element GS08 contains the version number to be used to validate this interchange/group. If this version is not found on the table file, the Editor checks the VERSION CONVERSION records from the global parameters. If a match is found, the alternate version from this parameter record is used to obtain the correct version of the standards from the table file. Global parameter, PARTNER VERSION (ALWAYS) GROUP, instructs the Editor to check the partner profile before going to the version conversion.

*See Partner Version (Always) Group.*

**UNA Interchange**

Although considered an interchange, this segment is not validated against the table file. It is an option in EDIFACT to identify: Component and Data Element Separators, Segment Termination character and other information for processing EDIFACT records.

**UNB Interchange**

This EDIFACT interchange uses the DEFAULT version from the table file for validation processing. The actual version number to be used for validating the remaining segments is located in the Group Envelope (UNG) or the Transaction Envelope (UNH).

For EDIFACT, Gentran:Basic supports the UNOA, UNOB, UNOC, UNOD, UNOE, and UNOF syntaxes. As of Syntax version 4, only the readable delimiters are valid.

**UNB\*UNOA:1**

Where UNOA indicates readable characters  
(such as: + ' ?).

Prior to Syntax version 4

**UNB\*UNOB:1**

Where UNOB indicates unreadable characters  
(Hex 1F, Hex 1D, and Hex 1C).

Following are examples of two delimiter syntax levels for interchange processing:

<b>SYNTAX A</b>	<b>Character</b>	<b>Hex Value</b>
Component Element Separator	:	7A
Data Element Separator	+	4E
Repetition Separator	*	
Decimal Notation	. or ,	
Release Character	?	6F
Segment Terminator	'	7D

Prior to Syntax version 4

<b>SYNTAX B</b>	<b>Character</b>	<b>Hex Value</b>
Component Element Separator		1F
Data Element Separator		1D
Decimal Notation	. or ,	
Release Character		
Segment Terminator		1C

Although not recommended, Syntax B delimiters are Teletype characters. They could be meaningful to many network or communication subsystems. If determining use, first discuss with your trading partner utilizing Syntax B delimiters.

### **SCH Interchange**

Similar utilization as UNA, it is not validated but does contain a segment separator component element separator, element separator, release character and segment terminator for the interchange.

### **STX Interchange**

This TRADACOMS interchange, as with the UNB, the default is used to validate this segment. The version number in STX01 is loaded at transaction validation (i.e., message identification) processing. Default delimiter for this is as follows:

<b>DEFAULT DELIMITER</b>	<b>Character</b>	<b>Hex Value</b>
Segment Separator	=	7E
Component Element Separator	:	7A
Element Separator	+	4E
Segment Terminator	'	7D
Release Character	?	6F

### *Interchange Partner Selection*

Gentran:Basic has a preset lookup sequence for interchange envelopes which is described in Appendix A. If the way you set up your partners agree with the preset sequence, the following global parameters are not needed. However, if you would like to change the sequence of the partner look up, these global parameters are needed.

**The following global parameters are all System-level global parameters and should be set through the online using Administrative Main Menu, Global Parameter Maintenance.**

### **BG Sender ID/Commid**

#### Partner/Qualifier Mode

This parameter indicates to the Editor that, for those trading partners sending you BG interchanges, you have set up the partner profile using the Sender ID or that the Sender ID is the key to the partner XREF file. If the Sender ID is not found, use COMMID to find the partner on the partner profile. Again, if COMMID is not found on the partner profile, use COMMID to look up an alias partner from the partner XREF file. If the partner is still not found, check the global parameters for an active VERIFY PARTNER INTERCHANGE. If this global parameter is active, suspend the interchange.

#### Relationship Mode

This parameter indicates to the Editor that, for those trading partners sending you BG interchanges, you have set up your partner relationship key using BG Sender ID (with spaces for the Qualifier) in the *partner* portion of the key and BG Receiver ID (with spaces for the Qualifier) in the *user* portion of the key. If no Relationship record is found, substitute COMMID for BG Sender ID and read the Partner Relationship file again. If no record is found, a partner profile Relationship key is built using the first 15 characters of the Receiver ID (user portion) and the first 15 characters of the BG Sender ID (partner portion). A read for the Header record on the partner profile is performed. If still no record is found, the global parameters are checked for an active VERIFY PARTNER INTERCHANGE. If this global parameter is active, the interchange is suspended. If a relationship record is found, the Editor loads the partner profile key from the partner relationship record and uses it to read the partner profile and Partner Control files.

### **BG Commid/Sender ID**

#### Partner/Qualifier Mode

This parameter indicates to the Editor that, for those trading partners sending you BG interchanges, you have set up the partner profile using COMMID or that COMMID is the key to the partner XREF file. If COMMID is not found, use the Sender ID to find the partner on the partner profile. Again, if the Sender ID is not found on the partner profile, use the Sender ID to look up an alias partner from the partner XREF file. If the partner is still not found, check the global parameters for an active VERIFY PARTNER INTERCHANGE. If this global parameter is active, suspend the interchange.

#### Relationship Mode

This parameter indicates to the Editor that, for those trading partners sending you BG interchanges, you have set up your partner relationship key using COMMID (with spaces for the Qualifier) in the *partner* portion of the key and BG Receiver ID (with spaces for the Qualifier) in the *user* portion of the key. If no Relationship record is found, substitute BG Sender ID for COMMID and read the Partner Relationship file again. If no record is found, a partner profile Relationship key is built using the first 15 characters of the Receiver ID (user portion) and COMMID (partner portion). A read for the Header record on the partner profile is performed. If there is still no record found, the global parameters are checked for an active VERIFY PARTNER INTERCHANGE. If this global parameter is active, the interchange is suspended. If a relationship record is found, the Editor loads the partner profile key from the partner relationship record and uses it to read the partner profile and Partner Control files.

## ISA Sender ID/Authorization

### Partner/Qualifier Mode

This parameter indicates to the Editor that, for those trading partners sending you ISA interchanges, you have set up the partner profile using the Sender ID or that the Sender ID is the key to the partner XREF file. If the Sender ID is not found, use AUTHORIZATION to find the partner on the partner profile again. If AUTHORIZATION is not found on the partner profile, use AUTHORIZATION to look up an alias partner from the partner XREF file. If the partner is still not found, check the global parameters for an active VERIFY PARTNER INTERCHANGE. If this global parameter is active, suspend the interchange.

### Relationship Mode

This parameter indicates to the Editor that, for those trading partners sending you ISA interchanges, you have set up your partner relationship key using the Sender ID/Sender ID Qualifier in the *partner* portion of the key and Receiver ID/Receiver ID Qualifier in the *user* portion of the key. If no Relationship record is found, substitute AUTHORIZATION for SENDER ID and read the Partner Relationship file again. If no record is found, a partner profile Relationship key is built using the first 15 characters of the Receiver ID (user portion) and the first 15 characters of the Sender ID (partner portion). A read for the Header record on the partner profile is performed. If still no record is found, the global parameters are checked for an active VERIFY PARTNER INTERCHANGE. If this global parameter is active, the interchange is suspended. If a relationship record is found, the Editor will load the partner profile key from the partner relationship record and uses it to read the partner profile and Partner Control files.

## ISA Authorization/Sender ID

### Partner/Qualifier Mode

This parameter indicates to the Editor that, for those trading partners sending you ISA interchanges, you have set up the partner profile using AUTHORIZATION or that AUTHORIZATION is the key to the partner XREF file. If AUTHORIZATION is not found, use the Sender ID to find the partner on the partner profile, again. If the Sender ID is not found on the partner profile, use the Sender ID to look up an alias partner from the partner XREF file. If the partner is still not found, check the global parameters for an active VERIFY PARTNER INTERCHANGE. If this global parameter is active, suspend the interchange.

### Relationship Mode

This parameter indicates to the Editor that, for those trading partners sending you ISA interchanges, you have set up your partner relationship key using Authorization/Sender ID Qualifier in the *partner* portion of the key and the Receiver ID/Receiver ID Qualifier in the *user* portion of the key. If no Relationship record is found, substitute Sender ID for AUTHORIZATION and read the Partner Relationship file again. If no record is found, a partner profile Relationship key is built using the first 15 characters of the Receiver ID (user portion) and the first 15 characters of AUTHORIZATION (partner portion). A read for the Header record on the partner profile is performed. If still no record is found, the global parameters are checked for an active VERIFY PARTNER INTERCHANGE. If this global parameter is active, the interchange is suspended. If a relationship record is found, the Editor loads the partner profile key from the partner relationship record and uses it to read the partner profile and Partner Control files.

### *Interchange Receiver Processing*

For inbound processing, the Verify Receiver Interchange parameter may be used to verify the receiver ID found on the incoming interchange envelopes against the Partner Profile.

**Note:** The Verify Receiver Interchange parameter is only valid if using Partner/Qualifier mode.

#### **ISA Interchange**

The Inbound Editor attempts to locate a Partner Control record using the interchange receiver ID (ISA08) and Qualifier (ISA07). If no record is found, a second attempt to locate a partner control record is made using the Interchange Security ID (ISA02) and Qualifier (ISA01), if the ISA01 is not equal to "00." If both read attempts fail, a warning message is generated.

#### **UNB Interchange**

The Inbound Editor attempts to locate a Partner Control record using the interchange receiver ID (UNB0301) and Qualifier (UNB0302). If no record is found, a warning message is generated.

#### **STX Interchange**

The Inbound Editor attempts to locate a Partner Control record using the interchange receiver ID qualifier (STX0301) if this value is not spaces. If the STX0301 is spaces, then the Inbound Editor attempts to locate a Partner Control record using the recipient identification (STX0302). If no record is found, a warning message is generated.

### **Partner Relationship Usage**

The Partner Relationship file is used by the Editors to convert Sender and Receiver IDs to a user/partner relationship key for the partner profile. The Relationship mode is determined through the online Configuration file. When processing in the Relationship mode, the Partner X-Ref file does not exist.

See the *Gentran:Basic for zSeries Release 6.4 User's Guide* for set up and usage information.

The EDI IDs from the interchange, group, and transaction envelopes are converted to a relational key for the partner profile through the Partner Relationship file. The partner relationship key is composed of two parts; user and partner. The Interchange or Group Receiver ID and Qualifier comprise the *user* portion of the key and the Interchange or Group Sender ID and Qualifier comprise the *partner* portion of the key. After each read of the Partner Relationship file, the status of the read is examined. If found, a partner profile relationship key is returned to the Editor. The Editor uses this relationship key to read the partner profile and Partner Control files. If the user EDI / Partner EDI relationship is not found, the first 15 characters of the Interchange or Group Sender ID and the first 15 characters of the Interchange or Group Receiver ID are used to build the partner profile and Partner Control relationship keys.

**Note:** No overriding parameters exist for this process sequence.

### **Partner X-Ref Usage**

The X-Ref file is used by the Editors to attempt resolution of unsuccessful reads to the partner profile. After each read of the partner profile using either the Interchange or Group Sender/Receiver ID and Qualifier, the status of the read is examined. If the Sender/Receiver ID and Qualifier are not found, the Partner X-Ref file is checked. Should the Sender/Receiver ID and



Qualifier be found on the Partner X-Ref, the Partner ID and Qualifier from the Partner X-Ref record is used in a second read of the partner profile. Processing within the Editors then uses the substituted Partner ID. Note that the output file is unchanged and still contains the original sender and receiver IDs from the input records; however, the Error Report generated by the Editors reflects the substituted Partner ID and Qualifier.

Gentran:Basic provides additional control of the partner selection process. If you have all of your partners set up using the partner X-Ref file, consider activating the global parameter, PARTNER XREF FIRST. This parameter causes the Editor to go directly to the partner X-Ref file without performing unnecessary reads against the partner profile.

Conversely, if you do not use the partner X-Ref file at all, consider activating the global parameter NO PARTNER XREF. This parameter instructs the Editor to process without the partner X-Ref feature and saves time on the number of partner look ups done by the Editor. The Partner X-Ref file can be used only when the online Configuration is set to the Partner/Qualifier mode.

## Group Envelope Processing

Group envelopes must have a Segment ID of GS, UNG, or BAT. Anything else suspends the input records until either an end of input is encountered or a valid interchange is found. Also, an error report is generated. Appendix A contains the present lookup sequence for Group envelopes.

## Group Qualifiers

Assignments of group Qualifiers for GS envelopes in previous releases of Gentran:Basic were determined from the Partner Control record. This has been changed to the System-level global parameters file. Below are optional parameters for controlling the Qualifier used by Gentran:Basic when looking up the group partner on the partner profile. Each of these parameters is explained in detail in Appendix A of this guide.

GS SENDER/RECEIVER QUALIFIER SPACES  
GS SENDER QUALIFIER SPACES  
GS RECEIVER QUALIFIER SPACES  
GS SENDER/RECEIVER QUALIFIER INTERCHANGE  
GS SENDER QUALIFIER INTERCHANGE  
GS RECEIVER QUALIFIER INTERCHANGE  
GS RECEIVER QUALIFIER INTERCHANGE/UN

If none of these global parameters are active, the Editor defaults to interchange for the GS sender and receiver Qualifiers.

For ANA TRADACOM Standards there is a different global parameter to be considered, USE STX QUALIFIER. This global parameter is explained in Appendix A of this guide.

Compliance checking begins with version selection from the table file. Element GS08, UNG07 or UNH02, and STX01 contain the version to be loaded from the table file. If Gentran:Basic can not find this version, the system checks the VERSION CONVERSION record from the Runtime global parameters to see if another version number is to be used. If found, Gentran:Basic makes a second attempt at loading the version from the table file. Only two reads are performed. If on the second attempt the version is still not found, Gentran:Basic suspends the data and generates an error report.

**Note:** The GS Assume logic works in both the Partner/Qualifier and Relationship modes. In the Relationship mode, the GS Qualifier issue is settled before the Partner Relationship file is read.

**The following global parameters are all System-level global parameters and should be set through the online using Administrative Main Menu, Global Parameter Maintenance.**

#### **Partner Version Always Group**

This System-level global parameter overrides the previous process, but does not change it. Gentran:Basic reads the partner profile looking for a group information record that matches the Functional Group ID (element GS01). The read is performed using the Sender ID and Functional Group ID (Partner/Qualifier mode) or using the User/Partner ID and Function Group ID (Relationship mode) from the Group envelope. If the Group record is found, its version number is used. Otherwise, an error report is generated and the above process is performed. An error causes the data to be suspended. If no version is found on the partner control record, the version on the control envelope is used. An error message is issued to indicate the version was not coded on the partner.

#### **Partner Version Group**

This System-level global parameter overrides the above process, but does not change it. Gentran:Basic reads the partner profile looking for a group information record that has the same Functional Group ID. The read is performed using the Sender ID (Partner/Qualifier mode) or using the User/Partner ID (Relationship mode) from the Group envelope. If the Group record is found, its version number is used. Otherwise, no error report is generated and the above process is performed. An error does not cause the data to be suspended.

#### **Verify Partner Group**

This System-level global parameter instructs Gentran:Basic to read the partner profile to verify the Group record exists. The read is performed using the Sender ID and Functional Group ID (Partner/Qualifier mode) or using the User/Partner ID and Function Group ID (Relationship mode) from the Group envelope. If the Group ID is not found, an error report is generated and the entire group is suspended until the end of file is reached or a valid interchange is found.

#### **Verify Receiver Group**

This system-level global parameter instructs Gentran:Basic to read the partner profile to verify that partner control record exists for the group receiver ID. If the record on the partner profile is not found, a warning message is generated.

**Note:** The Verify Receiver Group parameter is only valid if using Partner/Qualifier mode.

### **Transaction Envelope Processing**

Transaction set envelopes must have a Segment ID of ST, MHD, or UNH. Anything else causes the input to be suspended until either an end of input is encountered, or a valid interchange is found. Also, an error report is generated. Appendix A contains the present lookup sequence for Group envelopes.

Gentran:Basic checks the Transaction Set ID against the transaction set numbers from the version selected on the group envelope. If not found, the data is suspended.

Gentran:Basic uses the control number element in conjunction with the control number in the Trailer (SE, UNT) segment to verify segment integrity and prevent duplicate processing of data should the Databank parameter be in effect. An error message is generated if the control number is missing or if the Trailer control number is different from the Header control number.

**The following global parameter is a System-level global parameter and can be set on the Global Parameter Maintenance screens in the Administration subsystem.**

#### **Verify Partner Transaction**

This global parameter instructs Gentran:Basic to verify that a transaction record exists for this Transaction Set ID on the partner profile. If no transaction record is found, the data is suspended.

Gentran:Basic checks the Implementation Convention identifier (if present) against the IC Conversion Records. If not found, a message is generated and the group version is used.

### **Package Structure Processing**

Package structure processing begins with the object Header, and UNO segment, then ends with the object Trailer, and UNP segment. The UNO/UNP segments, used only in EDIFACT, act as transaction set envelopes except that they surround Object (binary) type data instead of EDI data. The UNO does not contain a transaction set, a control number or a Version ID. Therefore, the Editors use the version from the UNG or DEFAULT when evaluating the object Header and Trailer segments. The Editors do not compliance check object data. This means that object data passes directly through the Editors.

### **Segment Processing**

Once processing is complete on the interchange, group and transaction Header envelopes, data suspension is not possible. From this point, validation errors are either warnings or rejected based on whether the System-level global parameter PARTNER ERROR CONTROL is active. If the global parameter is not active, all errors are treated as warnings only. Otherwise, error processing follows the steps noted earlier in this section.

Gentran:Basic validates the sequence of each segment and its respective elements against the standards selected on the group envelope. Errors/warnings are generated for the following conditions:

- Mandatory segment missing
- Mandatory element missing
- Conditional elements not meeting the proper relationship
- Wrong type of data for element
- Incorrect element size (min/max)
- Segments out of sequence
- Too many loops or occurrences of a segment
- Too many elements for a segment
- Invalid code value in element
- Check for missing messages
- Check for wrong release level

For the CTT segment, Gentran:Basic validates the count and, if appropriate, the hash totals. Should an error be detected, an error/warning message would be generated on the error report. Also, ####REJ records may be generated if set up on partner profile and Partner Error Control is active.

## Trailer Envelope Processing

Trailer envelopes must have a Segment ID of SE, GE, IEA, ICE, EG, UNT, UNE, MTR, EOB, END, or UNZ. Anything else causes an error/warning to be generated on the error report. Data suspension does not occur for Trailer segment errors. If error rejection is desired, the partner profile must be set up and partner error control must be active.

**Note:** Previous ####REJs will cause the count to be different.

## Databank Processing

The databank feature for the Inbound Editor has two functions: directory and message store. For the Editor, the directory is primarily an audit function and the message store is like a snapshot of data received.

### *Directory (Audit)*

The level of auditing done by the Editor is determined by the following System-level global parameters and by the partner profile. Each one of these parameters are further explained in Appendix A of this guide.

INBOUND EDI INTERCHANGE DIRECTORY  
INBOUND EDI INTERCHANGE NONE

INBOUND EDI GROUP DIRECTORY  
INBOUND EDI GROUP NONE

INBOUND EDI TRANSACTION DIRECTORY  
INBOUND EDI TRANSACTION NONE

For each level of auditing desired, a System-level global parameter is required. Please note that activating the transaction level directory does not automatically activate the group level directory.

In conjunction with the global directory parameters, audit generation can be controlled through the partner profile. The partner profile level can be activated through the PARTNER DATABANK LEVEL System-level global parameter. For those trading partners with which auditing is desired, the interchange/group/transaction partners need to be updated through online partner maintenance with a D in the EDI Databank Inbound field. If both the partner level and the global level directories are active, the Editor takes the one containing the most detail.

### *Message Store*

To activate the inbound message store feature of the Editor, the following System-level global parameter must be active:

#### **Inbound EDI Interchange Full**

When the global parameter, INBOUND EDI INTERCHANGE FULL, is active, the Editor copies the wrapped input documents to a message store file for backup and reprocessing. To reprocess a

document, you need to use the online databank feature. Through the online databank, you mark those documents that need to be reprocessed. Afterwards, run the Inbound Editor using the runtime level global parameter – INBOUND EDI DATABANK REPROCESS.

Along with any new documents received, Gentran:Basic includes any document found in the reprocess file created from the online selection process.

Online maintenance for inbound documents, such as documents marked for deletion, is also applied to the inbound EDI databank files when the System-level global parameter INBOUND EDI DATABANK REPROCESS is active. You must then run the Inbound EDI Maintenance program (EDID301) to physically delete the document from the databank file.

### *Databank Key Structures*

The partner profile and databank have two modes of key structure, Partner/Qualifier and Relationship. The trading mode is selected through the online Configuration file. After selection, the mode applies to all partners and should not be mixed on the databank.

**Note:** No runtime overrides exist for the trading mode.

#### **Partner/Qualifier Mode**

When creating the directories for the inbound EDI databank, the trading mode to be used is extracted from Record Type 0 in the Configuration file. If Partner/Qualifier mode is indicated, the Sender Partner ID and Qualifier is the key. This is a resolved Partner ID and Qualifier which results after the Editor progresses through the different options of the partner X-Ref and GS partner records to find the partner on the partner profile.

#### **Relationship Mode**

When creating the directories for the inbound EDI databank, the trading mode to be used is extracted from Record Type 0 in the Configuration file. If the Relationship mode is indicated, the Partner Relationship file is used to obtain the user/partner (relationship) partner profile key.

Refer to the earlier section, Partner Relationship Usage, for an explanation on the process of determining the partner profile User/Partner ID used by the databank.

Functional acknowledgments received are written to a reconciliation file to be matched later in the Reconciliation Process (EDID510) program.

See the program description in the topic “Acknowledgment Reconciliation/Monitor (EDID510) program.”

If a databank global parameter is active, Gentran:Basic is checked for duplicates. This is helpful in preventing the same data from being received twice from your trading partners. An error message is generated and can be used to reject duplicate data by setting up your partner profile and by using partner error control.

## **Output Processing**

### *Directed Output Files*

For inbound processing, data separation can occur at the Partner, Group or Transaction level. Actual splitting of data takes place in the inbound Split (EBDI015) program, but the ####DIR

control record is created by the Editor. To initiate the Direction Split feature, the System-level global parameter, DIRECTED OUTPUT FILES, must be active in the Global Parameter file.

Example of a Direction record is as follows (where **b** = Blank):

```
0000000b5ISA###DIRb0001
0000001b5ISA###DIRCM00020016
```

### Direction Record

The following table describes the values on the Direction record.

Position	Value	Source of Value
1	0000000	Record number generated by Editor.
8	<b>b</b> 5	Sequence Number generated by Editor.
10	ISA	Created by the Editor to indicate that Segment ID from interchange record generated by specific partner.
13	###DIR	Inbound split constant.
19	C	Constant Value.
20	M	Two values possible: <b>b</b> – one output file for split <b>M</b> – multiple files for split.
21	0001	Up to 4 occurrences starting in position 21 of four characters each. Denotes last three digits from EBDI001 to be used as input to the Split program DD name (SYSxxx).

### File Selection

#### Partner/Qualifier Mode

Whether a multiple or a single split occurs, depends on the values found in the partner profile Data Separation record for the Receiver ID. If more than four splits are selected, the Editor generates multiple Direction records. If the Sender ID is desired, the System-level global parameter, SENDER ID FOR DATA SEPARATION, must be activated.

#### Relationship Mode

Whether a multiple or a single split occurs, depends on the values found in the partner profile Data Separation record for the User/Partner ID. If more than four splits are selected, the Editor generates multiple Direction records.

### TRADACOMS Logical Documents

TRADACOMS documents can be split by interchange (STX), group (BAT), or transaction message (MHD). When splitting by transaction message, use only the Header message on the partner profile. The Inbound Editor does not create Direction records for either the Detail or Summary messages. The Data Separation (EBDI015 – Split) program has been modified to process logical documents.

## Error Report Processing

### Error Report Always

Unless this System-level global parameter is active, Gentran:Basic only creates an error report if an error occurs. In either case, totals are always generated.

## Detail Report Processing

### Detail Report Always

This runtime global parameter produces an expanded error report with helpful detail information that may be used in problem resolution. This report is typically not used for standard production and may be lengthy. Reserve use for problem resolution situations.

## Viewpoint Processing

If Gentran:Viewpoint is installed, processing is performed automatically and parameters are not required. Control is maintained through the configuration file and Gentran:Viewpoint files. If a prior communications tracking program has processed the data, a Tracking record is included in the input file. The Inbound Editor recognizes and uses the information from the communications tracking program.

Position	Field Content	COBOL Picture	Format	Length	Description
001 – 007	Instream Viewpoint Tracking Identifier	X(07)	AN	07	Literal Constant ###VPTM
008 – 008	Filler	X(01)	AN	01	Not used
009 – 018	Viewpoint Tracking Trace ID	X(10)	AN	10	A unique ID value that identifies the item in the Viewpoint Tracking Management system.

## Files Used

### Input

The following files are used as input:

Filename	Description
SYS001	Tables File This file contains the details for each version of the ASC X12, TDCC, TRADACOMS, and EDIFACT.
ERRCTL	Error Message and Control File This file contains descriptions and control flags for every error number in the Gentran:Basic system.
EDISCD1	This file contains code values with lengths of 1 to 5 bytes. Standards Maintenance is used to update the file. This file is optional. See the CODE CHECK record for global parameters.

Filename	Description
EDISCD2	This file contains code values with lengths of 6 to 10 bytes. Standards Maintenance is used to update the file. This file is optional. See the CODE CHECK record for global parameters.
EDISCD3	This file contains code value with lengths of 11 to 20 bytes. Standards Maintenance is used to update the file. This file is optional. See the CODE CHECK record for global parameters.
EDISCD4	This file contains code values with lengths of 21 to 40 bytes. Standards Maintenance is used to update the file. This file is optional. See the CODE CHECK record for global parameters.
SYS030	<p>Runtime Global Parameter File</p> <p>This file contains all Runtime parameter records that are used by the Editors. These records apply to all partners.</p> <p>The Runtime parameters override the System-level global parameters through the online Administrative Main Menu, Global Parameter Maintenance.</p>
SYS040	<p>Input File (80-character input)</p> <p>This file contains the data from your trading partners.</p>
SYS090	<p>Partner Profile</p> <p>This file contains specific information needed on your trading partners for Gentran:Basic processing.</p>
SYS091	<p>Outbound Partner Control File</p> <p>Default acknowledgment generation and System-level global parameter PARTNER ACKNOWLEDGEMENT use this file for outbound control numbers. If the sender partner profile information does not exist, default control numbers, combining date and time, are used. Global USE CNTL FOR ACKNOWLEDGMENTS does not require the use of this file.</p>
SYS092	<p>Inbound Partner Control File</p> <p>Incoming EDI control numbers are compared or sequenced for coordination with the sender. All control numbers received are placed on the file. Partner, group, and transaction partner profile information has to exist for sequencing at all three levels. Envelopes using numeric EDI control numbers can select chronological or incremental sequence checking. If the envelope control numbers are not numeric, only chronological sequence checking can be used.</p>
SYS095	<p>Partner X-Ref File</p> <p>Alias partner identifications are used to resolve senders and receivers from the incoming interchange. System-level global parameters PARTNER XREF FIRST and NO PARTNER XREF direct the usage of this file.</p>
EDIRW	<p>Inbound Reprocessing File</p> <p>If selected, all interchange key information from the EDIIEA, Inbound EDI Directory file, is used to control inbound EDI databank reprocessing.</p>



Filename	Description
EDIQ097	CICS Queue File This file contains all information for queue management for the databank and is updated through databank maintenance. It is associated to inbound reprocessing.
EDISELD	Standards Element Description File
EDIPREL	Partner Relationship File Partner profile User/Partner IDs are determined from the Sender EDI ID, Receiver EDI ID, and Qualifiers from the interchange and group envelopes.

### Output

The following files are used as output:

Filename	Description
SYS006	Summary Report This report contains summary information for the Editor. Information includes processing parameters and control counts.
SYS007	Summary Report in Data Format This file contains the summary report control counts in data format.
SYS010	Error Report This file contains information from the Editor for each error detected in processing.
SYS050	Output File (30001-character variable output) This file contains all non-suspended data that was processed by the Editor from your trading partner. If selected, this file also contains embedded Direction and Error Rejection control records.
SYS060	Suspense File (80-character output) This file contains all data that did not have valid control records.
SYS070	Functional Acknowledge File (30009-character variable output) If acknowledgments are requested, this file contains the 997, 999, and CONTRL acknowledgments generated.
SYS091	Outbound Partner Control File Default acknowledgment generation and System-level global parameter PARTNER ACKNOWLEDGEMENT use this file for outbound control numbers. If the sender partner profile information does not exist, default control numbers, combining date and time, are used. Global USE CNTL FOR ACKNOWLEDGMENTS does not require the use of this file.

Filename	Description
SYS092	Inbound Partner Control File Incoming EDI control numbers are compared or sequenced for coordination with the sender. All control numbers received are placed on the file. Partner, group, and transaction partner profile information has to exist for sequencing at all three levels. Envelopes using numeric EDI control numbers can select chronological or incremental sequence checking. If the envelope control numbers are not numeric, only chronological sequence checking can be used.
EDIRW	Inbound Reprocessing File If selected, all interchange key information from the EDIIEA, Inbound EDI Directory file, is used to control inbound EDI databank reprocessing.

### Databank

The following are databank files:

Filename	Description
EDICFG	The Configuration File contains records to control System-level features and options.
EDIDAR	Reconciliation File (Output) All 997 functional acknowledgments, 999 acceptance/rejection transactions, and EDIFACT CONTRL messages received are used to place data acknowledgment reconciliation EDID510, acknowledgment reconciliation/monitor.
EDIIEA	Inbound EDI Directory File Inbound interchange directory information can be placed for all ASC X12, TDCC, TRADACOMS, and EDIFACT EDI interchanges. If selected, file contains linking information to the EDIIES, Inbound EDI Message Store file.
EDIIES	Inbound EDI Message Store File Inbound EDI data can be placed on this file and can be used for reprocessing.
EDIIECA	Inbound EDI databank change audit.
EDIAUD3	Inbound Application Databank Update Audit Trail

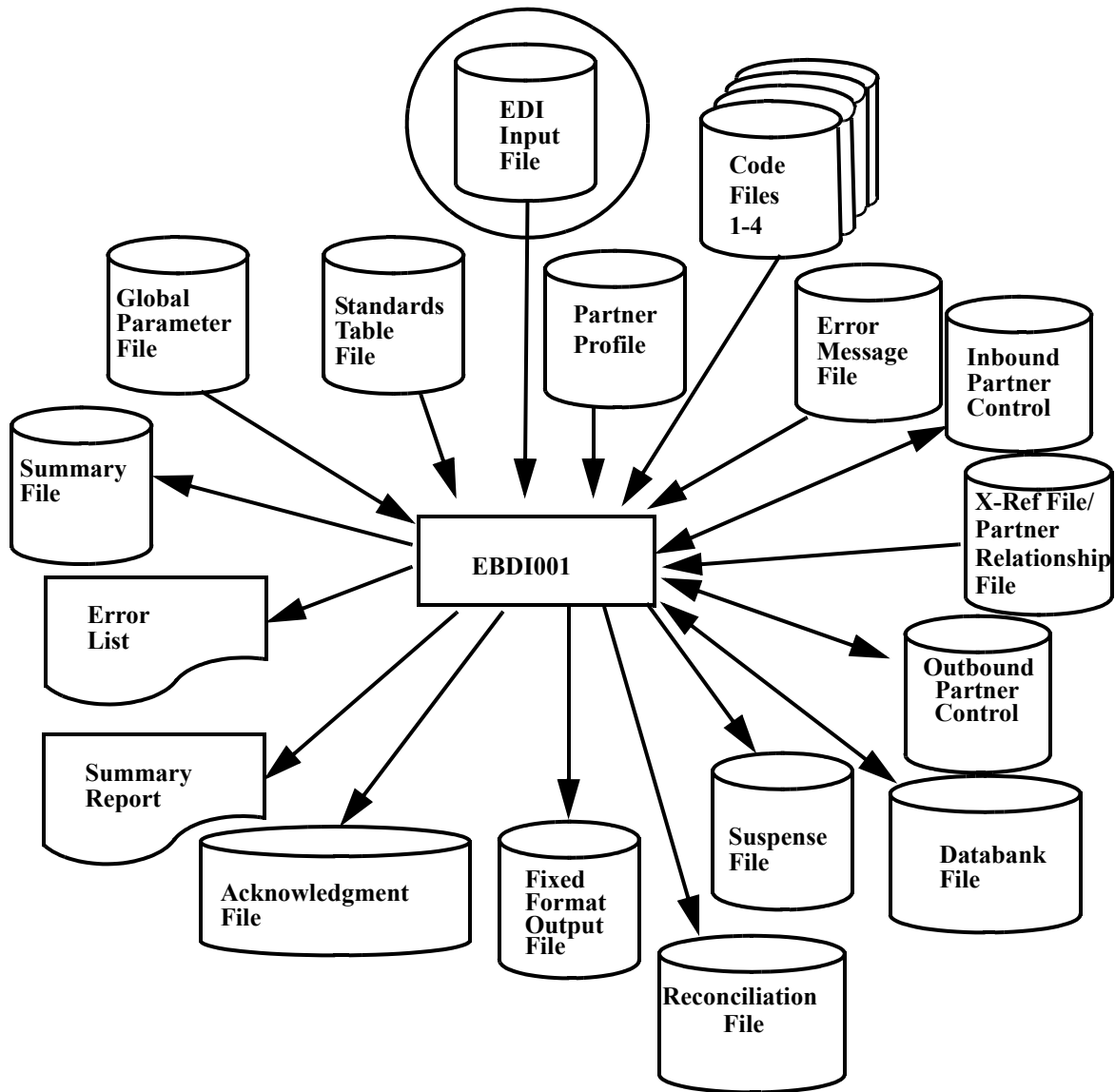
### Gentran:Viewpoint Files Used by Inbound Editor

The following are Gentran:Viewpoint files that are used by the Inbound Editor:

Filename	Description
EDIVT	The Trace ID file is used by Gentran:Viewpoint to store unique identifiers for tracked items.
EDIVP	The partner profile contains the rules (as defined in Extended System Management and Exception Management) used to control the Gentran:Viewpoint system.
EDIVSF	The Staging Files are used to pass data from a Gentran:Viewpoint file to Gentran:Basic. Staging files contain system messages and control/status information.

*Inbound Edit and Reformat Process*

Figure 2.6 illustrates the inbound edit and reformat process.



**Figure 2.6 Figure 2.1 EBDI001 — Inbound Edit and Reformat**

*Inbound Summary Report Data File*

The following table describes the Inbound Summary and Report Data file.

Field Content	COBOL Picture	Format	Length	Description
Summary Start Date	9(08)	ZD	08	Start of Editor run. Year, Month, Day.
Summary Start Time	9(08)	ZD	08	Editor Start time. Hours, Minutes, Seconds.
Summary End Date	9(08)	ZD	08	End of Editor run. Year, Month, Day.
Summary End Time	9(08)	ZD	08	Editor end time. Hours, minutes, Seconds.
Interchanges In	9(15)	ZD	15	Total number of interchanges read.
Groups In	9(15)	ZD	15	Total number of groups read.
Transactions In	9(15)	ZD	15	Total number transactions read.
Segments In	9(15)	ZD	15	Total number of segments read.
Records In	9(15)	ZD	15	Total number of records read.
Characters In	9(15)	ZD	15	Total number of characters read.
Characters Out	9(15)	ZD	15	Total number of characters written.
Characters Suspended	9(15)	ZD	15	Total number of characters suspended. <b>Note:</b> This field is not used by the Inbound Editor. It is used only by the Outbound Editor.
Interchanges Out	9(15)	ZD	15	Total number of interchanges written.
Interchanges Rejected	9(15)	ZD	15	Total number of interchanges rejected.
Interchanges Suspended	9(15)	ZD	15	Total number of interchanges suspended.
Groups Out	9(15)	ZD	15	Total number of groups written.
Groups Rejected	9(15)	ZD	15	Total number of groups rejected.
Groups Suspended	9(15)	ZD	15	Total number of groups suspended.
Transactions Out	9(15)	ZD	15	Total number of transactions written.
Transactions Rejected	9(15)	ZD	15	Total number of transactions rejected.

Field Content	COBOL Picture	Format	Length	Description
Transactions Suspended	9(15)	ZD	15	Total number of transactions suspended.
Segments Out	9(15)	ZD	15	Total number of segments written.
Segments Rejected	9(15)	ZD	15	Total number of segments rejected.
Segments Suspended	9(15)	ZD	15	Total number of segments suspended.
Databank Interchg Out	9(15)	ZD	15	Number of interchanges stored on databank.
Databank Interchg Rej	9(15)	ZD	15	Rejected interchanges stored on databank.
Databank Groups Out	9(15)	ZD	15	Number of groups stored on databank.
Databank Groups Rej	9(15)	ZD	15	Rejected groups stored on databank.
Databank Trans Out	9(15)	ZD	15	Number of transactions stored on databank.
Databank Trans Rej	9(15)	ZD	15	Rejected transactions stored on databank.
Databank Segments Out	9(15)	ZD	15	Number of segments stored on databank.
Databank Segments Rej	9(15)	ZD	15	Rejected segments stored on databank.
Databank Chars Out	9(15)	ZD	15	Number of characters stored on databank.
Databank Chars Rej	9(15)	ZD	15	Rejected characters stored on databank.
Databank Records Out	9(15)	ZD	15	Number of records stored on databank.
Databank Records Rej	9(15)	ZD	15	Rejected records stored on databank.
Acknowledgment Information Segment Generated	9(15)	ZD	15	Total acknowledgment segments generated.

Field Content	COBOL Picture	Format	Length	Description
Non-EDI-Segments	9(15)	ZD	15	Total non-EDI segments suspended.
Reject Pass-Thru Record	9(15)	ZD	15	Error rejection pass-thru records written.
Direction Records Written	9(15)	ZD	15	Direction pass-thru records written.
Pass-thru Records Written	9(15)	ZD	15	Total pass-thru records written.
Records Written	9(15)	ZD	15	Total records written.
Errors Written	9(15)	ZD	15	Error records written.
Ack Reconciliation Records	9(15)	ZD	15	Total acknowledgment reconciliation records generated.
Packages Written	9(15)	ZD	15	Total number of packages written.
Filler	X(3)	AN	3	Not used

**Note:** A COBOL copybook has been provided in the source utility product library. The member is SUMTWS. This copybook can be used by either Editor. Fields in the copybook may be shared or restricted to either Editor.

## Inbound Editor Reports

This section provides an explanation, as well as field descriptions, of the two reports produced by the Inbound Editor (EBDI001) program. The Error (SYS010) and Summary (SYS006) reports contain the following:

- *Error Report.* This report displays errors detected while processing incoming data through the Inbound Editor. Interchange, group, and transaction separate the errors.
- *Summary Report.* This report provides a record of the processing options in effect for editing incoming data through the Inbound Editor and the control counts. You will find these sections on the Summary Report:
  1. Runtime Global Parameter Overrides
  2. System Configuration Options
  3. Global Parameter List
  4. Global Parameter List – Version Conversion (optional)
  5. Global Parameter List – Implementation Convention (IC) Conversion (optional)
  6. Summary Control Counts Processing Incoming Data

Figure 2.7 through Figure 2.9 provide samples of the Error and Summary reports.

Error Report

```

EBDI001 RUN 12/01/2005 TIME 12:00 ERRORS ENCOUNTERED PROCESSING INCOMING DATA PAGE 1
ERROR RECORD SEG ELT/COMP
NUMBER NUMBER ID NBR INFORMATION ERROR MESSAGE VER = 00300 AGY = X

INTERCHANGE = VENDOR-1
TUTORIAL - XYZ COMPUTER COMPAN
ABC COMPANY
EDI SENDER 121212121 -
EDI RECEIVER 987654321 -
GROUP = VENDOR-1
TRANSACTION = INV01
689 15 SE TYPE = IN ENV = 000000007 VER = 003030 AGY = X
*** DOCUMENT AUDIT ID AUDIT ID = INV01 TEST TYPE = 810 ENV = 000070001 VER = 003030 AGY = X
TRANSACTION = INV02 DUPLICATE DOCUMENT ADDED TO DATA BANK.
689 28 SE TEST TYPE = 810 ENV = 000070002 VER = 003030 AGY = X
*** DOCUMENT AUDIT ID AUDIT ID = INV02 DUPLICATE DOCUMENT ADDED TO DATA BANK.
689 41 SE TEST TYPE = 810 ENV = 000070003 VER = 003030 AGY = X
*** DOCUMENT AUDIT ID AUDIT ID = INV03 DUPLICATE DOCUMENT ADDED TO DATA BANK.
689 54 SE TEST TYPE = 810 ENV = 000070004 VER = 003030 AGY = X
*** DOCUMENT AUDIT ID AUDIT ID = INV04 DUPLICATE DOCUMENT ADDED TO DATA BANK.
689 67 SE TEST TYPE = 810 ENV = 000070005 VER = 003030 AGY = X
*** DOCUMENT AUDIT ID AUDIT ID = INV05 DUPLICATE DOCUMENT ADDED TO DATA BANK.
689 79 SE TEST TYPE = 810 ENV = 000070006 VER = 003030 AGY = X
*** DOCUMENT AUDIT ID AUDIT ID = INV06 DUPLICATE DOCUMENT ADDED TO DATA BANK.
689 79 SE DUPLICATE DOCUMENT ADDED TO DATA BANK.
EBDI001 RUN 12/01/2005 TIME 8:29 ERRORS ENCOUNTERED PROCESSING INCOMING DATA PAGE 2
ERROR RECORD SEG ELT/COMP
NUMBER NUMBER ID NBR INFORMATION ERROR MESSAGE VER = 00300 AGY = X

689 80 IEA DUPLICATE DOCUMENT ADDED TO DATA BANK.
*** DATABANK ***** RUN # = 000000007

NO ERRORS OCCURRED DURING PROCESSING
PROCESSING ENDED NORMALLY.
* * * END OF REPORT * * *
    
```

Figure 2.7 Inbound Editor Error Report  
(Errors Encountered Processing Incoming Data)

Summary Report

EBDI001	RUN 12/01/2005 TIME 16:07	RUNTIME GLOBAL PARAMETER OVERRIDES	PAGE 1
DETAIL REPORT ALWAYS USE SUBELEMENT SEPARATOR FOR ANSI VERSION 00300 X 00200 X IC INPUT IDENTIFIER 00200 X			
EBDI001	RUN 12/01/2005 TIME 16:07	SYSTEM CONFIGURATION OPTIONS	PAGE 2
INTCHG VERSION = N GROUP VERSION = Y TRANSACTION VERSION = Y TRADING PROFILE MODE PARTNER QUALIFIER MULTIPLE ENVELOPE ENABLED CONCURRENCY DISABLED			
EBDI001	RUN 12/01/2005 TIME 16:07	GLOBAL PARAMETER LIST	PAGE 3
VERIFY PARTNER INTERCHANGE OFF VERIFY PARTNER GROUP OFF VERIFY PARTNER TRANSACTION OFF VERIFY RECEIVER INTERCHANGE OFF VERIFY RECEIVER GROUP OFF ERROR REPORT ALWAYS DETAIL REPORT ALWAYS OUTPUT MESSAGE ON DIRECTED OUTPUT FILES ICS TABLES OFF CODE CHECK ON BG PARTNER YES PARTNER ACKNOWLEDGEMENT PARTNER DATABANK OFF GS SENDER/RECEIVER QUALIFIER SPACES USE SUBELEMENT SEPARATOR FOR ANSI INBOUND EDI INTERCHANGE FULL INBOUND EDI GROUP DIRECTORY INBOUND EDI TRANSACTION DIRECTORY PARTNER SEQUENCE OFF			
EBDI001	RUN 12/01/2005 TIME 16:07	GLOBAL PARAMETER LIST - VERSION CONVERSION	PAGE 4
VERSION 00300 X 00200 X			

Figure 2.8 Inbound Editor Summary Report



```

EBDI001    RUN 12/01/2005 TIME 16:07    GLOBAL PARAMETER LIST - IC CONVERSION    PAGE 5
IC          INPUT IDENTIFIER            00200          X
EBDI001    RUN 12/01/2005 TIME 16:07    SUMMARY CONTROL COUNTS PROCESSING INCOMING DATA    PAGE 6
PROCESSING BEGAN ON 12/01/2005 AT 4:07 PM.

OVERALL INPUTS AND OUTPUTS SUMMARY

INTERCHANGES READ ----- 1
GROUPS READ ----- 1
TRANSACTIONS READ ----- 6
SEGMENTS READ ----- 226
RECORDS READ ----- 80
CHARACTERS READ ----- 6,392
INTERCHANGES WRITTEN ----- 1
INTERCHANGES REJECTED ----- 0
INTERCHANGES SUSPENDED ----- 0
GROUPS WRITTEN ----- 1
GROUPS REJECTED ----- 0
GROUPS SUSPENDED ----- 0
TRANSACTIONS WRITTEN ----- 6
TRANSACTIONS REJECTED ----- 0
TRANSACTIONS SUSPENDED ----- 0
SEGMENTS WRITTEN ----- 226
PACKAGES WRITTEN ----- 0
CHARACTERS WRITTEN (EXPANDED OUTPUT) ----- 8,435
MISCELLANEOUS OUTPUT SUMMARY
SEGMENTS SUSPENDED ----- 0
NON-EDI RECORDS SUSPENDED ----- 0
ERROR RECORDS WRITTEN ----- 0
TOTAL PASS-THRU WRITTEN ----- 0
DIRECTION PASS-THRU WRITTEN ----- 0
ERROR REJECTION PASS-THRU WRITTEN ----- 0
TOTAL RECORDS WRITTEN ----- 244
GENERATED ACKNOWLEDGEMENT SUMMARY
TOTAL ACK. INFORMATION GENERATED ----- 11

PROCESSING ENDED ON 12/01/2005 AT 4:07 PM.

-----
DATABANK OUTPUT SUMMARY
INTERCHANGES STORED ON DATA BANK ----- 1
REJECTED INTERCHANGES ON DATA BANK ----- 0
GROUPS STORED ON DATA BANK ----- 1
REJECTED GROUPS ON DATA BANK ----- 0
TRANSACTION STORED ON DATA BANK ----- 6
REJECTED TRANSACTIONS ON DATA BANK ----- 0
SEGMENTS STORED ON DATA BANK ----- 226
CHARACTERS STORED ON DATA BANK ----- 6,400
RECORDS STORED ON DATA BANK ----- 80
    
```

Figure 2.9 Continue – Inbound Editor Summary Report

## *Field Definitions*

### **Errors Encountered Processing Incoming Data**

Error Numbr:

This field displays the error code number associated with the error message.

Record Number:

This field displays the application record number.

Seg ID:

This field displays the EDI segment identifier.

Elt/Comp Nbr:

This field displays the EDI element/component number.

Information:

This field displays descriptive information identifying additional information about the error message.

Error Message:

This field displays descriptive information identifying the error message.

### **Runtime Global Parameter Overrides**

This section of the report displays Runtime global parameter overrides for this execution of the Inbound Editor, specified via Global Parameter file (SYS030) in the JCL. If you did not specify Runtime global parameters, this section will be blank.

### **System Configuration Options**

This section of the report displays the settings from the Configuration (EDICFG) file used by the Inbound Editor.

### **Global Parameter List**

This section of the report displays all the parameter options in effect for editing incoming data. Only those parameter options used are displayed.

### **Global Parameter List – Version Conversion (optional)**

This section of the report displays specified version conversion parameters for this execution of the Inbound Editor. If you did not specify version conversion parameters, the Editor will not produce this section.

### **Global Parameter List – Implementation Convention (IC) Conversion (optional)**

This section of the report displays specified IC conversion parameters for this execution of the Inbound Editor. If you did not specify IC conversion parameters, the Editor will not produce this section.

**Summary Control Counts Processing Oncoming Data**

(Message):

This field displays a text message indicating the date (in MM/DD/YYYY format) and the time (in HH:MM format) that processing began.

**Overall Inputs and Outputs Summary**

Interchanges Read:

This field displays the total number of interchanges read. Interchange control segments (i.e., ISA, BG, ICS, GS, UNA/UNB, SCH/STX) increment this total.

Groups Read:

This field displays the total number of groups read. Group control segments (i.e., GS, UNG, and BAT) increment this total.

Transactions Read:

This field displays the total number of transactions read. Transaction control segments (i.e., ST, UNH, MHD) increment this total.

Segments Read:

This field displays the total number of segments read. This field is based on number of segment terminators within the EDI data for each segment including interchange Headers.

Records Read:

This field displays the total number of records read. This field is a 255-byte outbound data segment physical record count. Each data segment is terminated.

Characters Read:

This field displays the total number of characters read. Providing that all interchange Headers have terminators, this count details all characters between first segment character and terminator.

Interchanges Written:

This field displays the total number of interchanges written. Interchange control Trailer segments (i.e., IEA, EG, ICE, GE, UNZ, END) increment this total, which includes sequential and databank output.

Interchanges Rejected:

This field displays the total number of interchanges rejected. Interchange control Trailer segments (i.e., IEA, EG, ICE, GE, UNZ, END) increment this total based on partner rejection.

Interchanges Suspended:

This field displays the total number of interchanges suspended. This field represents the occurrence of each interchange suspension as detected by the Inbound Editor.

Groups Written:

This field displays the total number of groups written. Group control Trailer segments (i.e., GE, UNE, EOB) increment this total, which includes both sequential and databank output.

**Groups Rejected:**

This field displays the total number of groups rejected. Group control Trailer segments (i.e., GE, UNE, EOB) increment this total based on group rejection.

**Groups Suspended:**

This field displays the total number of groups suspended. This field represents the occurrence of each group suspension as detected by the Inbound Editor.

**Transactions Written:**

This field displays the total number of transactions written. Transaction control Trailer segments (i.e., SE, UNT, MTR) increment this total, which includes both sequential and databank output.

**Transactions Rejected:**

This field displays the total number of rejected transactions. Transaction control Trailer segments (i.e., SE, UNT, MTR) increment this total based on transaction rejection.

**Transactions Suspended:**

This field displays the total number of transactions suspended. This field represents the occurrence of each transaction suspension as detected by the Inbound Editor.

**Segments Written:**

This field displays the total number of segments written on the expanded sequential output. This field is based on the number of segment terminators within the EDI data for each segment including interchange Header.

**Packages Written:**

This field displays the total number of EDIFACT packages within the wrapped output files.

**Characters Written (Expanded Output):**

This field displays actual characters written to the expanded output file. Direction markers, error rejection markers, error records, and pass-thru records are not included in this field. Trailing spaces are not included. Actual characters, numbers, and delimiters are included. This does not include characters written to suspension. This is considered normal expanded output.

**Characters Suspended:**

This field displays characters written to suspension file. This does not include spaces between EDI data segments.

**Miscellaneous Output Summary****Segments Suspended:**

This field displays EDI data segment suspended with segment terminators.

**Note:** Original envelopes are copied to suspense at the group and transaction level.

**Non-EDI Records Suspended:**

This field displays non-EDI records detected by the Editor between interchanges. Unconditional suspense may produce records in the suspense file. The records could be network Header information and inter-connect information. This information has been

detected by the Inbound Editor and does not contain interchange Headers (ISA, ICS, BG, ICS, GS, UNA/UNB, SCH/STX).

**Error Records Written:**

This field displays the number of Gentran error records written to the expanded output file. Global parameter OUTPUT MESSAGE OFF controls the use of error records in the expanded output file.

**Total Pass-Thru Written:**

This field displays the count of any records received with the pass-thru indicator '###' starting at column 1 of the input record.

**Direction Pass-Thru Written:**

This field displays the number of direction record markers within the expanded output file. Global parameter DIRECTED OUTPUT FILES controls the use of direction record markers within the expanded output file.

**Error Rejection Pass-Thru Written:**

This field displays the number of error rejection markers within the expanded output file. Global parameter PARTNER ERROR CONTROL controls the use of error rejection markers within the expanded output file.

**Total Records Written:**

This field displays total records within the expanded output file. Expanded segments, error records, direction markers, pass-thru records, and error rejection markers are included.

### **Generated Acknowledgment Summary**

**Total Ack. Information Generated:**

This field displays the total number of generated acknowledgments written to the functional Acknowledgment file. Generated segments and/or CNTL records are included. This file is used by EBDI011A – Acknowledgment Envelope generator, and EBDI002 – Inbound Editor.

**Total Ack. Reconciliation Generated:**

This field represents the total number of generated DATA ACKNOWLEDGEMENT RECONCILIATION records written within the DAR reconciliation file. These records represent incoming acknowledgment information. Acknowledgment documents information (CONTRL, 997, 999) is placed in the file. This file is used by EDID510 – Acknowledgment Reconciliation monitor.

### **Databank Output Summary**

**Interchanges Stored on Data Bank:**

This field displays the total number of interchanges stored on the data bank. Interchange control Trailer segments (IEA, EG, ICE, GE, UNZ, END) increment this total. This field represents the actual number of interchange directory records written to the inbound directory file.

**Rejected Interchanges on Databank:**

This field displays the total number of rejected interchanges on the databank. Interchange control Trailer segments (IEA, EG, ICE, GE, UNZ, END) increment this total based on partner rejection.

**Groups Stored on Databank:**

This field displays the total number of groups stored on the databank. Group control Trailer segments (GE, UNE, EOB) increment this total. This field represents the actual number of group directory records written to the inbound directory file.

**Rejected Groups on Databank:**

This field displays the total number of rejected groups on the databank. Group control Trailer segments (GE, UNE, EOB) increment this total based on group rejection.

**Transactions Stored on Databank:**

This field displays the total number of transactions stored on the databank. Transaction control Trailer segments (SE, UNT, MTR) increment this total. This field represents the actual number of transaction directory records written to the inbound directory file.

**Rejected Transactions on Databank:**

This field displays the total number of rejected transactions on the databank. Transaction control Trailer segments (SE, UNT, MTR) increment this total based on transaction rejection.

**Segments Stored on Databank:**

This field displays the total number of segments stored on the databank.

**Characters Stored on Databank:**

This field displays the number of compressed characters written. Actual number is Records Stored on Databank multiplied by 80.

**Records Stored on Databank:**

This field displays the total number of records stored on the databank. The records are 80-byte wrapped from the Editor and logically placed within the message store file.

---

## Inbound Split

## EBDI015

The Inbound Split program is used to either separate data into distinct disk files or sort data.

Documents may be separated (split) by Functional Group ID to assemble similar documents such as Purchase Orders and Purchase Order Corrections for application processing. Similar results can be accomplished by separating documents by transaction set, where transactions are aligned with your application requirements.

After splitting your documents by group, Functional ID, or Transaction Set ID, it is possible to split further by data content, if needed.

See the program description for the Data Router (EBDI095) program in this guide for more information on splitting by data content (e.g., purchase order number).

The Inbound Split program does not generate Direction (###DIR) records or Rejection (###REJ) records. These records are created by the Inbound Editor using both the Global Parameters file and the partner profile.

See the topic “Inbound Editor” in this section for more detail on creating both Rejection and Direction records.

Along with the Direction and Rejection Control records, the WRITE DIRECTION RECORD is the available parameter used for controlling the Split program. This parameter is only effective if the input has Direction records. Otherwise, it is ignored by the Split program.

### Inbound Split Processing

Processing begins with the reading of the control card file which contains control parameters to be used during processing.

#### Write Direction Records

This parameter instructs the Split program to include the direction record (###DIR) with its respective transactions. The normal process would delete the direction records after the transaction records have been written.

**Open Files SYSXXX**            where XXX stands for 001-099 files.

This parameter is used by the Split program to identify which of the 99 different split files you want opened each run, regardless of whether there are ###DIR records. If ###DIR record indicates a file not set up with an open parameter, the Split program opens and closes all split files indicated by direction records

#### Suppress Plus Records

This parameter instructs the Split program to drop the Plus records (###PLUS) and not write them to the slide file. If any ###PLUS records are found and this parameter is not used, the Plus records are written to the slide file.

After the parameters are stored, the transaction begins. The Split program looks for either direction (###DIR) or rejection (###REJ) records to guide its processing.

**Direction Records (####DIR)**

The direction control record tells the Split program to copy the transactions to specific output files. Each direction record contains up to four numbers that correspond to the last three digits of the output split files. The split files have a DDNAME of SYSxxx where the xxx can be a value of 001 to 099. These numbers instruct the Split program to write the transactions to their proper output files. A copy of the transactions is written to each file indicated on the direction record. Should any other value be found, the Split program generates an error report and write the transactions to the default Slide file.

**Rejection Records (####REJ)**

The rejection control record tells the Split program all transactions from the start of rejection (####REJS) until it finds an end of reject marker (####REJE) are to be written to the reject file.

**PLUS Records (####PLUS)**

Plus records are not used by the Split program. These records are sent with the inbound EDI data by trading partners using Gentran:Plus. These records are used by the Inbound Editor and then passed along. If Plus records are found in the data, the Split program writes them to the slide file unless the SUPPRESS PLUS RECORDS parameter is used.

If no control records are found, the Split program writes the transactions to the default Slide file.

**Logical Documents**

A TRADACOMS document is composed of at least three individual messages: Header, Detail, and Summary. These individual documents are treated as one *logical* document by the Inbound Editor. This means that only one ####DIR and/or one ####REJ record is generated for each TRADACOMS document. The Inbound Split program has been enhanced to properly process TRADACOMS logical documents. The Split program holds the ####DIR and/or ####REJ records throughout the entire logical document.

**Error Report**

An error report is generated when the Split program detects a problem with the control records. It contains the sequence number of the record in error along with an error number and description. If an error is detected, the program generates an error message and terminates.

**Files Used**

The following files are used by EBDI015 program:

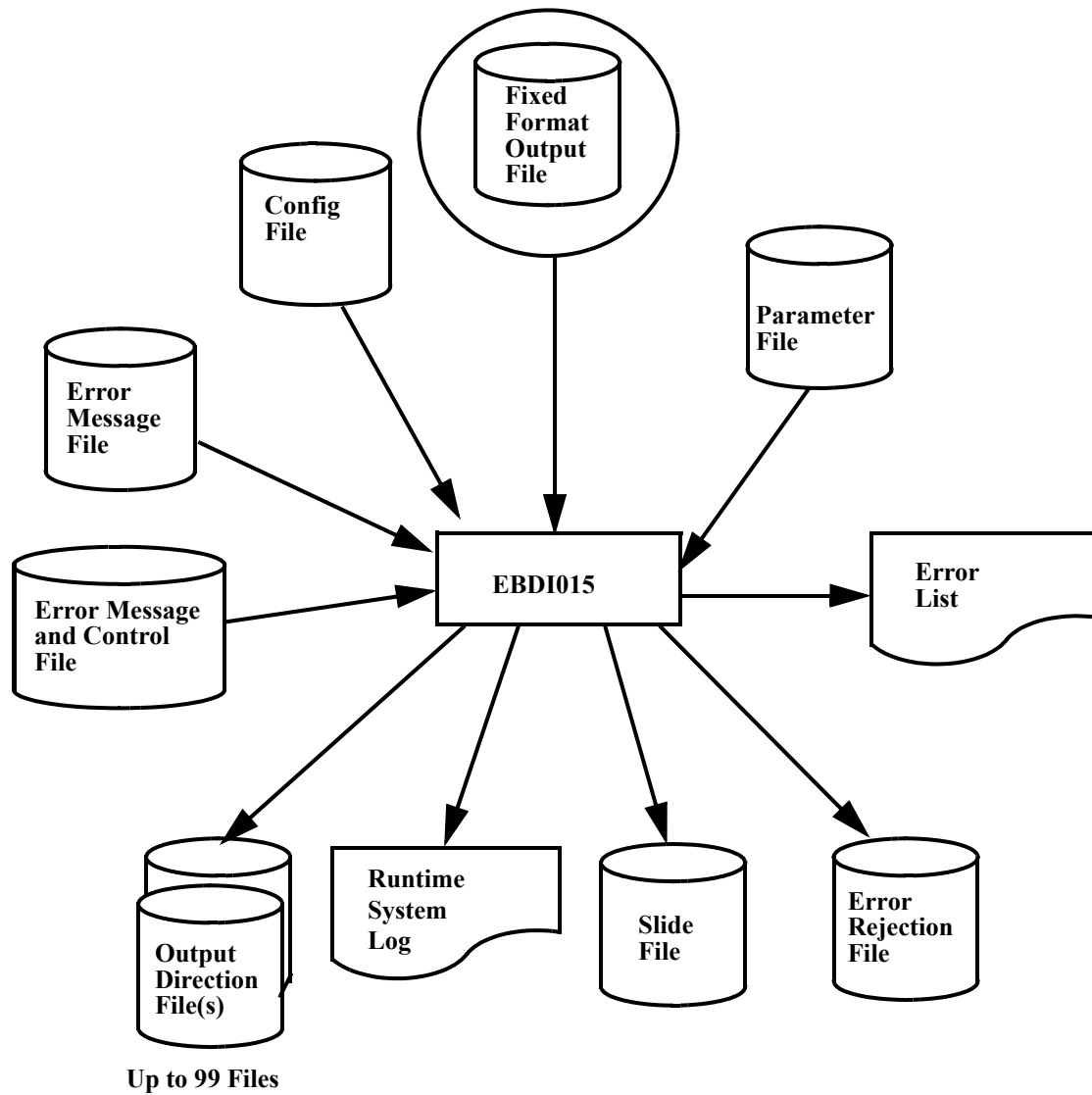
Filename	Description
SYS000	Slide File (30001-character variable input) This is the default file for the Split program. It is used when no direction records are found.
SYS001 thru SYS099	Output Direction Files (30001-character variable output) These files are used by the Split program when direction records are present. The three-digit codes on the Direction records correspond to the last three digits of the SYSXXX DDNAME.



Filename	Description
SYS160	Input File (30001-character variable input) This file contains the edited data from your trading partner with imbedded Direction records and reject records, if requested.
SYS163	Parameter File (80-character input) This file contains input parameters to the Split process. The valid parameters are: WRITE DIRECTION RECORDS Indicating that all direction records are to be copied with their data to split file(s). OPEN FILE SYSxxx This parameter indicates that the named split file is opened every time the Split program is run. SUPPRESS PLUS RECORDS Indicates that ###PLUS records should not be written to the slide file.
SYS166	Error Report This report contains any errors detected during processing.
SYS167	Reject File (30001-character variable output) This file contains all data marked with reject records from the Editors.
ERRCTL	Error Message and Control File (Input) This file contains descriptions and control flags for every error number in the Gentran:Basic system.
EDICFG	Gentran:Basic System Configuration File This file contains records to control System-level features and options.

*Inbound Split Process*

Figure 2.10 illustrates the Inbound Split process.



**Figure 2.10 EBDI015 — Inbound Split Process**

## Inbound Split Processing Parameters

### *WRITE DIRECTION RECORDS Parameter*

The following table describes the WRITE DIRECTION RECORDS parameter.

Positions	Field Content	COBOL Picture	Format	Length	Description
1 – 23	Write Direction Records	X(23)	AN	23	Indicates direction marker records for input file are to be written to output file. Others are dropped. Valid value: WRITE DIRECTION RECORDS.
24 – 80	Filler	X(57)	AN	57	Not used

### *OPEN FILE SYSxxx Parameter*

The following table describes the OPEN FILE SYSxxx parameter.

Positions	Field Content	COBOL Picture	Format	Length	Description
1 – 16	Open File SYSxxx	X(16)	AN	16	Indicates that the split file is opened every time the Split program is run (xxx stands for split file 001-099).
17 – 80	Filler	X(63)	AN	63	Not used

### *SUPPRESS PLUS RECORDS Parameter*

The following table describes the SUPPRESS PLUS RECORDS parameter.

Positions	Field Content	COBOL Picture	Format	Length	Description
1 – 21	Suppress Plus records	X(21)	AN	21	Indicates that ###PLUS records should not be written to the slide file. Valid value: SUPPRESS PLUS RECORDS
22 – 80	Filler	X(59)	AN	59	Not used

## Runtime System Log List

Figure 2.11 is an example of the Runtime System Log list.

```
RUNTIME SYSTEM LOG LIST
*****
PROGRAM EBDI015 COMPILED 12/01/2005 16.47.47
VERSION 6.4 GENTRAN:BASIC 12/01/2005
CURRENT DATE IS 12/01/2005
TIME STARTED IS 12:00:00
*****

*****
RECORDS READ = 00003296
DIRECTION RECORDS READ = 00000057
RECORDS WRITTEN = 00003244
DIRECTION RECORDS PROCESSED = 00000047
*****
REJECT RECORDS WRITTEN = 00000000
NON-SPLIT RECORDS WRITTEN = 00000201
RECORDS WRITTEN TO SPLIT FILE 0001 = 00000001
RECORDS WRITTEN TO SPLIT FILE 0004 = 00000002
RECORDS WRITTEN TO SPLIT FILE 0005 = 00000054
RECORDS WRITTEN TO SPLIT FILE 0007 = 00000911
RECORDS WRITTEN TO SPLIT FILE 0008 = 00000813
RECORDS WRITTEN TO SPLIT FILE 0025 = 00002013
RECORDS WRITTEN TO SPLIT FILE 0099 = 00000060
*****NORMAL END OF JOB*****
```

**Figure 2.11 Runtime System Log List**

## Miscellaneous Output Summary

### Records Read:

This field displays the total number of inbound records read.

### Direction Records Read:

This field displays the total number of direction markers (###DIR) read. Direction markers determine split file processing.

**Note:** Direction Records Read count is included in Records Read.

### Records Written:

This field displays the number of records written at least once to the split and non-split output files. If the WRITE DIRECTION RECORDS parameter is active, this field includes counts for direction markers.

### Direction Records Processed:

This field displays the total number of direction markers (###DIR) processed.

**Note:** Direction Records Processed count is included in Records Written.

### Reject Records Written:

Total number of rejected records. This count includes ###REJ start and end control records.

### Non-Split Records Written:

Total number of records written with no splitting (no direction markers from input).

### Records Written to Split File xxxx:

Total number of records written to a specific split file indicated by 'xxxx' value.

---

## Inbound Data Mapping

**EBDI041**

The Inbound Data Mapping (EBDI041) program is the last step in the inbound translation process. This program is responsible for reading the edited EDI data and translating it into an application format to be processed further by a user application.

### Processing Overview

The Inbound Mapper reads the EDI data file and translates it into application data. To perform this translation, it is necessary to instruct the Mapper as to which application definition to process. After loading the application definition, the Mapper determines the correct EDI Transaction Definition to use to translate the data, based on partner profile options.

Optionally, information concerning each application document and the data records contained in each document can be stored in the Inbound Application Databank. Summary information about each document is stored in the databank directory, while the physical application data is stored in the databank message store. Selection of the Databank feature can be specified at runtime via a parameter option or on the partner profile.

Databank information is stored so that it can be accessed by document within partner (the sender) records. Multi-divisional organizations have the option of storing this data by document within user and partner (the sender and receiver) records. Because you will need to perform additional processing when using this option, we recommend that you use this option only when required by your organization.

The application definition to generate is presented to the Inbound Mapper on the parameter record. This can be done in one of two ways. If the data in the input EDI data file is consistent, meaning that it all can be translated to one application definition, the name of that application definition can be provided directly on the parameter record (SYS001). This method is sufficient in the majority of situations.

If, however, the data in the EDI Data file is to be translated to more than one application definition, partner profile Data Separation records that define the application definition to process, must be established at the transaction level. To process multiple application definitions, it is necessary to define the application to process on the first input parameter record (SYS001, columns 1-10) as `##INSTREAM`. If the Databank Message Store feature is not used, all of the resulting application data is written to one physical output file. Data set information (fixed or variable record format, record length) is determined from the first input parameter record. Use of the `##INSTREAM` option might degrade performance considerably as the same application definition might be loaded many times in one run.

Once the application definition is loaded, the EDI segments that are defined in the Header section of the definition are read and loaded into memory.

The user (only if user/partner databanking is specified) and the Partner Definitions are loaded from information provided in the input file by the Inbound Editor.

Based on the Partner Definitions the Transaction Definition that is used to translate the data is selected and loaded. If the Transaction Definition cannot be determined, the EDI document is suspended, and the document is written to the EDI suspense file.

The Header EDI information that is held in memory is used to generate the Header level application data that is defined in the application definition. If the Databank Message store option has been selected, the Header application data records are stored.

The Detail section of the document is processed one line item at a time. That is each iteration of the defined Detail loop is loaded into memory and mapped to the application definition, one at a time. If it is impossible for the Mapper to load an entire line item into memory at once, an error message is issued and processing is terminated. When this happens, the line item must be defined in “sections.” This feature is explained in detail later in this chapter. If the Databank Message store option has been selected, the Detail application data records are stored.

The Summary section of the document is read into memory and the Summary application data is written. If the Databank Message store option has been selected, the Summary application data records are stored.

If the Databank feature has been selected, the Directory entry is written for the application document.

This process is repeated until all EDI documents have been translated.

An audit trail and summary reports are produced. A return code is set that can be tested for in the job stream.

### **Processing Detail – User (Receiver) Determination**

The term *user* in inbound processing refers to the receiver of the application data. It is only used when processing in Relationship mode which is the *user* portion of the User/Partner ID. This is specified in Record 0 of the Configuration file. The Editor will resolve User/Partner ID from the inbound EDI data using the partner Relationship file and pass it to the Inbound Mapper.

### **Processing Detail – Partner (Sender) Determination**

The term *partner* in inbound processing is the sender of the EDI data. This is the *partner* portion of the relationship User/Partner ID. The Editor will resolve User/Partner ID from the inbound EDI data using the Partner Relationship file and pass it to the Inbound Mapper.

The partner is determined by the Inbound Editor program and is passed to the Inbound Mapper.

### **Processing Detail – Transaction Determination**

The Inbound Mapper attempts to select the Transaction Definition (Map) to use to map the data to the application file based on the application definition and selected partner profile.

After the application definition is loaded, a list of all maps associated with that application are loaded into memory. The two types of maps are:

- General – These maps are not associated to any specific partner.
- Partner-specific – The maps are only selected when their name is defined on the partner Transaction record (partner profile) in the inbound Translation ID field.

The selection process is outlined below:

1. The Mapper attempts to find a partner-specific map.
2. The list of all available maps (e.g., all maps that have been defined to translate EDI data to a specific application) is scanned for the first partner-specific map.
3. If one is found, the Mapper accesses the partner profile using the Transaction ID defined for the map as the key.

**Note:** For TRADACOMS users, the combination of Transaction ID and release character is used for the partner profile lookup. If a record for that particular release does not exist, the Transaction ID is used as the sole key.

For all users, the version portion of the Transaction ID is not used during map selection.

The Transaction ID that is defined for the map is compared to the transaction set identifier contained in the EDI data. If the above are the same and the name that identifies the map matches the Translation ID Inbound field (passed from the Inbound Editor from the partner's Transaction record), the map is selected. If the map is not selected, the Mapper looks for the next partner-specific map defined.

If the system could not select a partner-specific map, an attempt is made to select a general map. The list of available maps being held in memory is scanned for the first general map. If one is found, the version contained in the EDI data is compared to the version that is defined for the map. If these match, the map is selected. If the map is not selected, the Mapper looks for the next general map defined. If map selection fails, the EDI document is suspended.

## Processing Detail – Sectioning

Sectioning improves system performance and makes processing easier by enabling you to subdivide the Detail section of an Application and Transaction Definition into as many as nine sections. Each of these sections is processed as a discrete unit of work.

To truly understand sectioning, it is necessary to understand just how the mapping programs process data. To provide the greatest flexibility, the Mappers normally process data as follows:

1. Data is read into memory until it is determined that the Mapper has loaded one complete unit of work. A *unit of work* is defined as either the entire Header section of a document, one line item of a document (a *line item* is one iteration of the Detail section, from one occurrence of the first Detail record to the next), or the entire Summary section. All the data contained in a unit of work cannot exceed 1 million characters.

**Note:** If the character limit is exceeded, an error message is issued and processing terminates. If this error is encountered, the Detail area of the document must be sectioned using the online Mapping subsystem.



2. All output data defined to correspond to the unit of work of the input data is created.
3. All input data for the appropriate section is available for output.

This scenario works well for the majority of the business documents as defined in the EDI Standards. However, certain Application or Transaction Definitions are defined with nested looping structures that cause one iteration of the Detail section of a document to contain many records or segments.

When the Detail area of the document is sectioned, the following restrictions apply:

- No mapping is allowed between sections (e.g., you cannot move data from the Header area directly to the Detail area). To move data from a higher section to a lower section, the data must be mapped in its own section, saved into a constant, then loaded from the constant to the new destination. Data cannot be mapped from a lower section to a higher section.
- No attempt is made to map data to a section unless some input is received that is in that section. For example, you cannot generate an outbound segment that is composed of all literals, unless an input record is received that is in the corresponding section.
- All data defined for a section must be contiguous both when the data is read and when the data is written.

## Processing Detail – Enveloping

### *Map Envelope Elements to Application*

It is possible to map envelope information directly to the application from the EDI envelope data elements. All of the beginning envelope data elements are loaded into reserved word constants named for the specific element (e.g., ISA02).

See the Glossary for a complete list of all reserved word constants.

Be advised that all of the application fields that are defined to receive the data from these envelope elements should be text fields and should be of the correct length to receive the EDI element. No data type conversions or zero filling are performed. Data is truncated on the right, if necessary.

### *Test/Production Determination*

The Test/Production status of a document can be mapped to the application using specific reserved word constants.

The test/production status of the map definition is loaded into the reserved word constant – PROD-TEST-FLAG.

The test/production status of the transaction set as defined on the partner profile is loaded into the reserved word constant – PART-TEST-FLAG.

The test/production status of the received interchange (ISA and UNB) is loaded into reserved envelope constants (ISA and UNB, respectively) named as the elements are to the standards. Any of the reserved word constants may be used to load an application field.

## Processing Option – Instream Parameters

If ##INSTREAM is indicated in columns 1-10 on the mapping parameter record (SYS001) as the application definition to process, the Mapping program expects that a valid application definition be established on a Transaction Level Data Separation record (partner profile) for each transaction set to be processed. The application definition to process is passed to the Inbound Mapper from the Inbound Editor.

If this option is used, normal processing would be to store all of the application data on the Inbound Application Databank Message Store and use the Databank Extract program to pull the data from the store at a scheduled time.

It is possible to write multiple applications to a sequential file; however, you must be sure to implement the process to differentiate one application format from another.

If the Inbound Mapper is to produce the sequential output Application Data file, file information must be specified on the Mapping Parameter record (SYS001). See the Mapping Parameters Layout later in this section for further information. Failure to provide this information results in termination of the run. If the Inbound Application Message Store is used and the sequential output Application Data file is not written, file information is obtained from the Application Data ID, when the File Type parameter (column 28) is set to spaces. File information specified on the parameter record is used.

If the system does not find the application definition, a GT639 error is generated (Unable to find application data ID specified in parameters) and processing ends immediately with a return code of 16. If the Mapper finds a return code of less than 08 on the Error Control file for error number 639, the Mapper suspends the bad application data and continues processing any remaining applications. You must change the return code for error number 639 to a value less than 08 to enable this feature. You can change the return code on the Message Maintenance screen. Return code 1 is used for inbound processing and return code 2 is used for outbound processing.

## Processing Option – Databanking

### *Application Databank Facility*

Application Databank provides the facility so that Application data can be tracked and stored as it is processed by the Inbound Mapping program.

The tracking of the data is accomplished using the Databank Directory. The storing of data uses the Databank Message Store. Databanking can be performed at two levels or turned off completely.

Directory Only	Only the databank directory entry is created. Document tracking is performed but data storage is not.
Full	The databank directory and message store entries are created. Document tracking and data storage are performed.

If Partner Databank Level is specified, the Inbound Application Databank Level field on the Transaction record of the partner profile determines the level of databanking.

The databank level for the run is set using a combination of two switches on the Mapping Parameter record: the Databank Configuration switch (column 36) and the Databank Processing Level switch (column 37). These two switches are used in conjunction to determine how Databanking is to be performed.

The Databank Configuration switch, column 36 of the Mapping Parameter record (SYS001), tells the Mapper how to initialize the databank, so it should be set to the highest level that might be requested.

The Databank Processing Level Switch, column 37 of the Mapping Parameter record (SYS001), identifies the lowest level of databanking processing required. Partner requests at a higher level are performed. If there is a request for a databank operation at a higher level than the configuration setting, an error is generated and processing terminates.

### Examples

When the Databank Configuration switch is set to **Full** and the Databank Processing Level switch is set to **Full**, databank processing is performed for all documents in the run.

When the Databank Configuration switch is set to **Directory** only and the Databank Processing level switch is set to **No** databank, no databanking is performed for any documents in the run.

### Message Store Vs. Sequential Output File

The Databank Message Store facility can replace the sequential application output file (SYS003) produced by the Inbound Mapper. If the Inbound Application Databank Extract (EDID405) program is used to retrieve data from the Message Store and produce the input to the application process, there may be no need for the Inbound Mapper to write the application file. To stop the sequential application file from being written, column 43 of the Mapping Parameter record (SYS001) should be set to **1**. The application records is then written to the Message Store only. The generation of the sequential application file can be stopped *only* if FULL databanking is used.

## Processing Option – User Exits

### Overview

Gentran:Basic can handle exceptional situations that regular mapping cannot resolve. Examples of two such situations are accessing files outside of the system and accessing nonstandard data conversions.

**Note:** To ensure minimal impact on current Gentran:Basic customers, a mapping parameter is required to instruct the mapping programs to use the new call block. This parameter is in column 55 of the Mapping Parameter record and should be set to **2** if you want to use the new call block.

**Caution:** By using user exits, you are circumventing all of the built-in edits of the system. Results that are returned by the exit are mapped into the data as instructed.

### *Version 1 User Exits*

User exits before the release of Gentran:Basic Release 5.5 are known as version 1 user exits. Element and segment level user exits for version 1 are described in the paragraphs that follow.

#### **Element Level User Exits (version 1)**

User exits can be invoked in any of these times during the mapping process:

- At the start of the mapping run after the Gentran:Basic files are opened. This exit, called the Begin Process User Exit Routine, is specified on the parameter record. This routine opens any needed files and performs any preliminary processing.
- At the end of the mapping run after Gentran:Basic files are closed. This exit, called the End Process User Exit Routine, is specified on the parameter record. This routine closes any open files and performs any end-of-run processing.
- As each element is mapped. By specifying a user exit routine at the element level on the Extended Element Mapping screen, the exit is called each time the mapping for that element is invoked. If conditions that have been set up on the mapping of the element are not satisfied, the exit is not invoked.

All user exits at the element level use the element level call block record layout to communicate with the main module. Note that all of the reserved constants are available to the exit; the reserved constants should be modified only if they are not used elsewhere in the mapping.

See the Element Level User Exit Record Layout (version 1) in Record Layouts later in this section.

#### **Segment Level User Exits (version 1)**

Segment level user exits give additional processing power to the Mapping Integration System. The segment level user exits are invoked after the specified segment is read (inbound), or before the specified segment is written (outbound). The segment is passed to the user exit, and can be modified as necessary.

See the Segment Level User Exit Record Layout (version 1) in Record Layouts later in this section.

### *Version 2 User Exit*

Version 2 has an expanded call block that passes more information to the user exit. A version 2 user exit must be used for the Application Record Exit.

To take advantage of the version 2 exit or the expanded call block, you must indicate to the Mapper via the EXIT-VERSION parameter on the first parameter record that you want to use version 2 of the user exits. If you have existing user exits and do not want to take advantage of the version 2 features, you may want to indicate that you want to process with version 1 of the user exits.

**Note:** You *cannot* mix version 1 and version 2 user exits.

The version 2 user exit consists of a common call block that is used for all user exit types. These user exit types are:

- **Start Process Exit.** This exit is used at the start of the mapping run after the Gentran:Basic files are opened. This exit, called the Begin Process User Exit Routine, is specified on the parameter record. This routine opens any needed files and performs any preliminary processing.
- **Application Record Exit.** This exit is used just after an application record is read during Outbound Mapping and just before an application record is written during Inbound Mapping. This user exit, called the Application Record Exit, is similar to the current segment level user exit. This user exit provides you with exit points at all input and output points in the processing flow. The application record is passed to the user exit and can be modified as necessary.
- **Segment Exit.** Segment level user exits give additional processing power to the Mapping Integration System. The segment level user exits are invoked after the specified segment is read (inbound), or before the specified segment is written (outbound). The segment is passed to the user exit, and can be modified as necessary.
- **Element Exit.** By specifying a user exit routine at the element level on the Extended Element Mapping screen, the exit is called each time the mapping for that element is invoked. If conditions that have been set up on the mapping of the element are not satisfied, the exit is not invoked.
- **End Process Exit.** This exit is used at the end of the mapping run after Gentran:Basic files are closed. This exit, called the End Process User Exit Routine, is specified on the parameter record. This routine closes any open files and performs any end-of-run processing.

This common call block includes all of the functionality provided in version 1 user exit.

See the All User Exits Record Layout (version 2) in Record Layouts later in this section.

**Note:** The Mapper clears only the areas that are valid for that type of record. For example, if application record level processing is taking place, segment level processing fields are not cleared during execution

## Processing Option – EDI Data Print

Optionally, the first 125 characters of each EDI segment read and each Application record generated can be printed. This option is selected by entering **Y** in column 11 of the Mapping Parameter record in the Mapping Parameters file (SYS001). The User/Partner IDs and names can also be printed when they are loaded. The print option is selected by loading a **1** in column 42 of the parameter record (SYS001). These features can be extremely useful when first setting up and testing a map. These print options are selected with a runtime parameter. Both are also available when using the Instream parameter option.

See Parameters File Layout – Inbound later in this section for additional information.

## Processing Option – Return Code Setting

The Mapper can be instructed to issue a return code at the end of the mapping process. To turn on this feature, enter **Y** in column 34 of the parameter record (SYS001). This return code can be interrogated by the job stream and conditional processing can be performed based on the value of the return code.

Return codes are located on the error message and control file. These codes can be modified online to meet your installation standards, through the (error) Message Maintenance subsystem.

See Chapter 5 in the *Gentran:Basic for zSeries Release 6.4 User's Guide* for additional information.

## Processing Option – NTE Segment Processing

Default processing of the floating NTE segment requires that the Transaction Definition define each place in the EDI document that an NTE segment might be received. Each of these definitions are processed independently.

This can become inconvenient for organizations that are dealing with a multitude of trading partners that might send NTE segments anywhere.

To simplify the mapping process, the NTE float switch in column 35 of the parameter record (SYS001) can be set to **Y** to instruct the Mapping program to treat all NTE segments received within a section of the document the same way. Using this option, the NTE segment must be defined to the map only once per section.

## Processing Option – Alternate Real Application Format

Default processing maps numeric data into application real (R) fields as follows:

In the first example, the number 1.1 is received in the EDI data and is to be mapped to a five character application real field. The resulting data in the application field is 1.100

In the second example, the number -1.1 is received in the EDI data and is to be mapped to a five character application real field. The resulting data in the application field is -1.10

By setting the Alternate Application Real Switch in column 33 of the parameter record to **Y**, the Mapper can be instructed to right justify these numbers and leave the first character reserved to indicate positive or negative. The first character is a blank if positive and a minus sign if negative.

In the first example, when the number 1.1 is received in the EDI data and is to be mapped to a five character application real field and the Alternate Application Real Switch is set to **Y**, the resulting data in the application field is “<space>01.1”.

In the second example, when the number -1.1 is received in the EDI data and is to be mapped to a five character application real field and the Alternate Application Real Switch is set to **Y**, the resulting data in the application field is -01.1.

## Processing Option – Viewpoint Processing

The Inbound Mapping program automatically performs all actions required for Gentran:Viewpoint Exception Management and Tracking Management monitoring within Gentran:Basic.

For Viewpoint Tracking Management, if tracking is to continue after the Inbound Mapper by a user written application Tracking program, the Inbound Mapping program can be instructed to include required information for the user Tracking program.

For Gentran:Viewpoint Tracking Management users who want to continue tracking data into their user applications systems, the Inbound Mapping program passes required information at the end of each document.

## Processing Option – Verify Map Version

During normal processing, if partner-specific maps are used, the Inbound Mapper will unconditionally use the transaction map specified on the partner profile to translate the data from an EDI format to an application format.

By setting the Verify Map Version option in column 77 of the parameter record to **Y**, the Mapper can be instructed to verify that the Version ID on the incoming document matches the Version ID on the partner-specific map. If the Version IDs do not match, the document will be suspended.

## Processing Option – First Segment Mandatory

The Inbound Mapper uses a default set of logic when processing input segments to select which segment in the map should be selected for mapping. Based on the complexity of the looping structure of a transaction set, the Inbound Mapper may not consistently select the correct segment in the map. This parameter allows the Inbound Mapper to invoke alternate logic to aid in segment selection.

This parameter is set by specifying a value in column 52 in the first record of the parameter record (SYS001):

- Specify **Y** to indicate to the Inbound Mapper that the first segment in every loop received is mandatory and is present in the incoming EDI data. This is often used with X12 837 documents.
- Specify **X** to indicate to the Inbound Mapper that the first segment in every loop received is mandatory and is present in the incoming EDI data and, because sectioning is used in the map, extended search logic needs to be invoked to select the correct segment.
- Specify **L** to indicate to the Inbound Mapper that the transaction you are processing has deeply nested loops with multiple loops beginning with the same segment ID. The logic invoked in the Inbound Mapper is different than if Y or X is specified.

**Note:** There is a special mapping requirement if “L” is specified in this field. You must use Loop-End segments in the map to explicitly define the nesting of the loops. For every loop in the map, there must be a corresponding Loop-End segment.

For more information about how to use Loop-End segments, see “Segments Screen (EDIM504)” in Chapter 5 of the *Gentran:Basic User’s Guide*.

Specifying “L” in column 52 also provides you with more flexibility when using sectioning. While it is normally bad practice to create new sections in a map that start with the same segment ID, this is not a restriction using this parameter.

An EDIFACT DELFOR message is a good example where using the “L” option may be beneficial.

## Files Used

### Input

The following files are used as input:

Filename	Description
SYS001	Parameters File This file contains the parameters that drive the Inbound Mapping process.
ERRCTL	Error Message and Control File This file contains descriptions of every error in the Gentran:Basic system.
SYS004	EDI Data This file contains the EDI data after it has been processed by the Inbound Editor program. It may have optionally been processed by the inbound Split program. For Gentran:Structure: This data has been processed by the Gentran:Structure Preprocessor program.
SYS020	Application Data Definition Contains the definition of the Application Data file.
SYS021	Application Record Definition Contains the definition of the records that are part of the Application Data file.
SYS022	Application Field Definition Contains the definition of the fields that are part of the Application Data file.
SYS023	Application/Transaction Cross Reference Provides a cross reference to all Transaction Definitions that refer to the Application Data Definition.
SYS030	Code Translation File Translates ID type fields from the application value to standard value.
SYS031	Code Validation File Used to validate EDI data values.



Filename	Description
SYS032	Data Translation File Translates trading partner data values to the application data values.
SYS040	Transaction Definition Contains the definition of the transaction to be mapped.
SYS041	Transaction Segment Definition Contains the definition of the segments to be mapped for this Transaction.
SYS042	Transaction Element Definition Contains the definition of the elements to be mapped for this Transaction.

### Output

The following files are used as output:

Filename	Description
SYS003	Application Data File This file contains the result of the inbound mapping process. The data is in the format defined by the application definition. The restrictions of the definition of this file are: Variable Block/Fixed Block Format Longest record no longer than 5000 bytes
SYS005	Error Report This file lists all of the errors encountered while mapping the EDI data to the application.
SYS006	Summary Report This file contains summary information for the mapping run. Information includes processing parameters, control counts, and performance information.
SYS007	Summary Report in Data Format This file contains the summary report control counts in data format.
SYS099	EDI Suspense File This file contains all EDI documents that were not mapped because a transaction map could not be selected.

### Databank

The following are databank files:

Filename	Description
EDICFG	The Configuration File. This file contains records to control System-level features and options.

Filename	Description
EDIIAA	Inbound Application Databank Directory. This file contains status and control information for outbound application data and maintains records for controlling data recovery.
EDIIAS	Inbound Application Databank Message Store. This file contains application data. Data on the message store file can be viewed and optionally edited using CICS Databank Maintenance.
EDIIEL	Inbound EDI Databank Link File This file contains control information used to connect each EDI transaction to the same document on the Inbound Application Databank.

### *Gentran:Viewpoint (for Gentran:Viewpoint Customers Only)*

The following are Gentran:Viewpoint files:

Filename	Description
EDIVP	The partner profile contains the rules that are used to control the Gentran:Viewpoint system as defined in Extended System Management and Exception Management.
EDIVSF	The Staging file is used to pass data from the Gentran:Viewpoint enabled Inbound Mapper to Gentran:Viewpoint. Staging files contain error messages issued by the Inbound Mapper as well as control/status information.
EDIVT	The Trace ID file is used by Gentran:Viewpoint Tracking Management to maintain unique identifiers for tracked items.

### *Parameters File Layout – Inbound*

The following tables describe the inbound parameters file layout.

#### **Record 1**

Positions	Field Content	COBOL Picture	Format	Length	Description
001 – 010	Application Data ID	X(10)	AN	10	The name assigned to the definition of the Application Data file, or ##INSTREAM to specify application parameters.
011 – 011	EDI Data Print Switch	X(01)	AN	01	<b>Y</b> – Instructs the Mapper to print the first 125 characters of each record read and each record generated.
012 – 012	Forced Abend Switch	X(01)	AN	01	<b>Y</b> – Instructs the Mapper to abend if any errors are encountered during execution.

Positions	Field Content	COBOL Picture	Format	Length	Description
013 – 019	Start Process User Exit	X(07)	AN	07	The name of a user written program to be executed after all Gentran:Basic files have been opened.
020 – 026	End Process User Exit	X(07)	AN	07	The name of a user written program to be executed after all Gentran:Basic files have been closed.
027 – 027	Application Decimal Notation	X(01)	AN	01	A comma (,) instructs the Mapper to interpret a comma as a decimal point in the application file.
028 – 028	Application File Type	X(01)	AN	01	If using Instream application parameters, enter <b>V</b> in this field to indicate the application file is variable; enter <b>F</b> in this field to indicate a fixed format.
029 – 032	Application File Length	9(04)	N	04	If using Instream application parameters, code the maximum record length of the application file.
033 – 033	Alternate Application Real Switch	X(01)	AN	01	<b>Y</b> – Instructs the Mapper to right justify application real numbers.
034 – 034	Return Code Switch	X(01)	AN	01	If = <b>Y</b> , generate return code.
035 – 035	NTE Float Switch	X(01)	AN	01	<b>Y</b> – instructs the Mapper to process all NTE segments received in a section with one definition.
036 – 036	Databank Configuration Switch	X(01)	AN	01	<b>0</b> – No Databank <b>1</b> – Full Databank <b>2</b> – Directory Only Databank.
037 – 037	Databank Processing Level Switch	X(01)	AN	01	<b>0</b> – No Databank <b>1</b> – Full Databank <b>2</b> – Directory Only Databank. <b>3</b> – Partner Databank Level
038 – 039	Filler	X(02)	AN	02	Not used
040 – 040	Not Used	X(01)	AN	01	Not used
041 – 041	Not Used	X(01)	AN	01	Not used
042 – 042	Partner Print Switch	X(01)	AN	01	<b>0</b> – Do not print user/partner information when partner is loaded <b>1</b> – Print partner information when partner is loaded.
043 – 043	Write to Output File Switch	X(01)	AN	01	<b>0</b> – Write application to an output file in addition to the message store if requested. <b>1</b> – Do not write to the output file.
044-044	Document Tracking Switch	X(01)	AN	01	<b>1</b> – Print an entry on the Audit Trail for each document in error in the run. <b>2</b> – Print an entry on the Audit Trail for each document processed in the run.

Positions	Field Content	COBOL Picture	Format	Length	Description
045 – 045	Support Quote Switch	X(01)	AN	01	<b>Y</b> – Execute additional program logic to allow a single quote in reserved word constants and literal constants.
046 – 046	Input Format Type	X(01)	AN	01	Used for Gentran:Structure only. <b>F</b> – Input to this program was generated by the inbound Structure pre-processor. <b>V</b> or <b>blank</b> – Input to this program was generated by the Inbound Editor (default).
047 – 051	Filler	X(05)	AN	05	Not used
052 – 052	Mandatory Segment	X(01)	AN	1	Used to invoke alternate map segment selection logic: <b>Y</b> – First segment in every loop received is mandatory and present in the incoming EDI data. This is often used with X12 837 documents. <b>X</b> – Same as when Y is specified but includes extended search capabilities when sectioning is used in the map. <b>L</b> – Used for transactions that have deeply nested loops with multiple looks beginning with the same segment ID.
053 – 054	Filler	X(02)	AN	02	Not used
055 – 055	User Exit Version	X(01)	AN	01	<b>1</b> – Use the version 1 call blocks for user exits. <b>2</b> – Use the version 2 call block for user exits.
056 – 062	Filler	X(10)	AN	10	Not used
063 – 063	Viewpoint User Tracking (Gentran: Viewpoint Tracking Management only)	X(01)	AN	01	<b>0</b> – Viewpoint tracking is not performed by a user application tracking program. <b>1</b> – Viewpoint tracking is performed by a user application tracking program.
064 – 076	Filler	X(13)	AN	13	Not used
077 – 077	Verify Map Version	X(01)	AN	01	<b>Y</b> – Instructs the Mapper to verify the Version ID used to compliance check the incoming document matches the version ID found on the selected partner-specific map. If the Versions IDs do not match, the document will be suspended.
078 – 080	Filler	X(03)	AN	03	Not used

*User Exit Call Blocks*

The following tables describe User Exit call blocks.

**User Exit Call Block Layout (Element Level – version 1)**

<b>Positions</b>	<b>Field Content</b>	<b>COBOL Picture</b>	<b>Format</b>	<b>Length</b>	<b>Description</b>
001 – 001	Processing	X(01)	AN	01	This field is set by the calling program to indicate where in the process the module was called: <b>S</b> – start of process <b>P</b> – during process at the element level <b>E</b> – end of process.
002 – 081	User Defined Area	X(80)	AN	80	This field can be defined for any purpose. Its intent is to provide a method of communicating between different exits and subsequent invocations of the same exit in different documents. This field is not used by the mapping process.
082 – 161	Field To Be Mapped	X(80)	AN	80	On entry to the exit, this field contains the result of the mapping up to this point. The user exit can modify this field causing the data contained in this field to be mapped.
162 – 163	Length of Field To be Mapped	S9(04) COMP	BI	02	On entry to the exit, this field contains the length of the field to be mapped. If the exit modifies the field to be mapped, it also should set this length field to correspond.
164 – 165	Standard Minimum Length	S9(04) COMP	BI	02	This field contains the minimum length of the element to be mapped as defined by the standard. It should not be modified by the user exit.
166 – 167	Standard Maximum Length	S9(04) COMP	BI	02	This field contains the maximum length of the element to be mapped as defined by the standard. It should not be modified by the user exit.
168 – 169	Standard Data Type	X(02)	AN	02	This field contains the data type of the element to be mapped as defined by the standard. It should not be modified by the user exit.
170 – 4569	Constants 01 thru 55 (55 occurrences)	X(80)	AN	4400	These fields contain the 55 reserved constants that can be referred to on the mapping screens. The user exit can use these fields to pass data within the same document.

Positions	Field Content	COBOL Picture	Format	Length	Description
4570 – 4769	Hash Totals 01 thru 20 (20 occurrences)	S9(12) V9(6) COMP-3	PD	200	These fields contain the 20 reserved hash totals that can be referred to on the mapping screens. The user exit can use these fields to pass data within the same document.
4770 – 4970	Accumulators 01 thru 20 (20 occurrences)	S9(18) COMP-3	PD	200	These fields contain the 20 reserved accumulators that can be referred to on the mapping screens. The user exit can use these fields to pass data within the same document.

## User Exit Call Block Layout (Segment Level – version 1)

Positions	Field Content	COBOL Picture	Format	Length	Description
1 – 256	Segment	X(256)	AN	256	The segment (inbound), or segment to be written (outbound).
257 – 4656	Constants 01 thru 55 (55 occurrences)	X(80)	AN	4400	These fields contain the 55 reserved constants that can be referred to on the mapping screens. The user exit can use these fields to pass data within the same document.
4657 – 4856	Hash Totals 01 thru 20 (20 occurrences)	S9(12) V9(6) COMP-3	PD	200	These fields contain the 20 reserved hash totals that can be referred to on the mapping screens. The user exit can use these fields to pass data within the same document.
4857 – 5056	Accumulators 01 thru 20 (20 occurrences)	S9(18) COMP-3	PD	200	These fields contain the 20 reserved accumulators that can be referred to on the mapping screens. The user exit can use these fields to pass data within the same document.

## All User Exits (version 2)

Positions	Field Content	COBOL Picture	Format	Length	Description
1 – 80	USER-EXIT-USER-AREA	X(80)	AN	80	Used by the user exit module to communicate information to other user exit modules and subsequent invocations of the same module. It is never initialized or accessed by the mapping program, so any data that is placed in the field by any exit module remains in the field until it is removed or modified by the user.
81 – 380	USER-EXIT-FIELD	X(300)	AN	300	Valid only for element level exits. This field contains the data that is mapped to the target field or data element, and it can be modified by the exit module. The value in this field is mapped to the target field. If this field is modified, the next field, USER-EXIT-FIELD-LENGTH, should also be modified to reflect the length of the field to be mapped.
381 – 382	USER-EXIT-FIELD-LENGTH	S9(04)	BI	2	Valid only for element level exits. This field contains the length of the data that is mapped to the target field or data element. It should be modified by the exit module if the previous field, USER-EXIT-FIELD, is modified.
383 – 383	USER-EXIT-FIELD-FLAG	X(01)	AN	01	Must be equal to <b>Y</b> if any changes to have been made to the user exit field.
384 – 619	USER-EXIT-REFERENCE-AREA	Group		236	For reference purposes only. <b>Caution:</b> If you modify the data in these fields, unpredictable results can occur. If you must map the content of any of these fields, its value should be moved to one of the 99 available constants and mapped from there.

Positions	Field Content	COBOL Picture	Format	Length	Description
384 – 384	PROCOPT	X(01)	AN	01	Indicates where in the process the user exit was invoked. Valid values: <b>S</b> – Starts process exit. This exit is invoked at the beginning of the mapping process. <b>E</b> – Ends process exit. This exit is invoked at the end of the mapping process. <b>P</b> – Element level exit. This exit is invoked just before a target application field (inbound) or target data element (outbound) is loaded. <b>B</b> – Segment level exit. This exit is invoked just after a segment is read (inbound) or just before a segment is written (outbound). <b>R</b> – Application record level exit. This exit is invoked just after an application record is read (outbound) or just before an application record is written (inbound). <b>X</b> – Envelope generation level exit. This exit is invoked only during the outbound process, just before the envelopes are written to the EDI output file.
385 – 386	FIELD-MIN	S9(04)	BI	02	Valid only for element level exits. This field contains the minimum length of the intended target field of data element.
387 – 388	FIELD-MAX	S9(04)	BI	02	Valid only for element level exits. This field contains the maximum length of the intended target field of data element.
389 – 390	FIELD-TYPE	X(02)	AN	02	Valid only for element level exits. This field contains the data type of the intended target field of data element.



Positions	Field Content	COBOL Picture	Format	Length	Description
391 – 429	USER-INFO	Group		39	Fields containing information about the user. Valid only if the Trading Relationship mode is USER/PARTNER. If the relationship mode is USER/PARTNER, these fields always contain valid information during element level exit processing, outbound segment level exit processing, and inbound application record level processing. These fields contain valid information during inbound segment level processing and outbound application record processing only after the entire Header portion of the input has been loaded. <b>Caution:</b> When using these types of exits, carefully review the fields before updating to avoid inadvertent modifications.
391 – 425	USER-ID	X(35)	AN	35	The resolved User ID used to translate the document. The USER-ID will only contain valid values if trading mode on the Configuration file is set to <b>M</b> for mixed. If the trading mode is set to <b>R</b> (relationship) on the Configuration file, only the first 15 characters of USER-ID will be used. The USER-ID-QUAL field will be spaces.
426 – 429	USER-ID-QUAL	X(04)	AN	04	The resolved User ID Qualifier used to translate the document. The USER-ID-QUAL will only contain valid values if trading mode on the Configuration file is set to <b>M</b> for mixed. If the trading mode is set to <b>R</b> (relationship) on the Configuration file, only the first 15 characters of USER-ID will be used. The USER-ID-QUAL field will be spaces.

Positions	Field Content	COBOL Picture	Format	Length	Description
430 – 468	PARTNER-INFO	Group		39	Fields that contain information about the trading partner. These fields always contain valid information during element level exit processing, outbound segment level exit processing, and inbound application record level processing, and contain valid information during inbound segment level processing and outbound application record processing only after the entire Header portion of the input has been loaded. <b>Caution:</b> When using these types of exits, carefully review the fields before updating to avoid inadvertent modifications.
430 – 464	PARTNER-ID	X(35)	AN	35	The resolved Partner ID used to translate the document. The PARTNER-ID will only contain valid values if trading mode on the Configuration file is set to <b>M</b> for the Mixed mode. If the trading mode is set to <b>R</b> (relationship) on the Configuration file, only the first 15 characters of PARTNER-ID will be used. The PARTNER-ID-QUAL field will be spaces.
465 – 468	PARTNER-ID-QUAL	X(04)	AN	04	The resolved Partner ID Qualifier used to translate the document. The PARTNER-ID-QUAL will only contain valid values if trading mode on the Configuration file is set to <b>M</b> for mixed. If the trading mode is set to <b>R</b> (relationship) on the Configuration file, only the first 15 characters of PARTNER-ID will be used. The PARTNER-ID-QUAL field will be spaces.

Positions	Field Content	COBOL Picture	Format	Length	Description
469 – 488	INTERCHANGE-REF	X(20)	AN	20	The value of the Interchange Reference (control number) field. This field is valid only during inbound processing. It always contains valid information during element level exit processing and inbound application record level processing, and contains valid information during inbound segment level processing only after the entire Header portion of the input has been loaded. <b>Caution:</b> When using these types of exits, carefully review the fields before updating to avoid inadvertent modifications.
489 – 508	GROUP-REF	X(20)	AN	20	The value of the Functional Group Reference (control number) field. This field is valid only during inbound processing. It always contains valid information during element level exit processing and inbound application record level processing, and contains valid information during inbound segment level processing only after the entire Header portion of the input has been loaded. <b>Caution:</b> When using these types of exits, carefully review the fields before updating to avoid inadvertent modifications.
509 – 528	TRANSACTION-REF	X(20)	AN	20	The value of the Transaction (message) Reference (control number) field. This field is valid only during inbound processing. It always contains valid information during element level exit processing and inbound application record level processing, and contains valid information during segment level processing only after the entire Header portion of the input has been loaded. <b>Caution:</b> When using these types of exits, carefully review the fields before updating to avoid inadvertent modifications.

Positions	Field Content	COBOL Picture	Format	Length	Description
529 – 548	USER-REF	X(20)	AN	20	The value of the User Reference field. This field is valid during inbound and outbound processing. It always contains valid information during element level exit processing, outbound segment level processing, and inbound application record level processing, and contains valid information during inbound segment level processing and outbound application record level processing only after the entire Header portion of the input has been loaded. <b>Caution:</b> When using these types of exits, carefully review the fields before updating to avoid inadvertent modifications.
549 – 572	PLUS-BATCH-ID	X(24)	AN	24	The value of the Gentran:Plus Batch ID field. This field is valid only during outbound processing. It contains valid data only if the batch ID is mapped to the ##ADD card. It always contains valid information during element level exit processing and outbound segment level processing, and contains valid information during outbound application record level processing only after the entire Header portion of the input has been loaded and the interchange envelopes have been generated. <b>Caution:</b> When using these types of exits, carefully review the fields before updating to avoid inadvertent modifications.
573 – 586	PLUS-BATCH-NUMBER	X(14)	AN	14	The value of the Gentran:Plus® batch number. This field is valid only during inbound processing. It is passed to the Inbound Mapping program by the Inbound Editor for variable-format standards. It contains valid information during inbound segment level processing only after the entire Header portion of the input has been loaded. <b>Caution:</b> When using these types of exits, carefully review the fields before updating to avoid inadvertent modifications.

Positions	Field Content	COBOL Picture	Format	Length	Description
587 – 592	FUNCTIONAL-GROUP	X(06)	AN	06	<p>The value of the functional group for the document to be translated. This field is valid during inbound and outbound processing. It always contains valid information during element level exit processing, outbound segment level processing, and inbound application record level processing, and contains valid information during inbound segment level processing and outbound application record level processing only after the entire Header portion of the input has been loaded.</p> <p><b>Caution:</b> When using these types of exits, carefully review the fields before updating to avoid inadvertent modifications.</p>
593 – 598	TRANSACTION-SET	X(06)	AN	06	<p>The value of the Transaction Set ID of the document to be translated. This field is valid during inbound and outbound processing. It always contains valid information during element level exit processing, outbound segment level processing, and inbound application record level processing, and contains valid information during inbound segment level processing and outbound application record level processing only after the entire Header portion of the input has been loaded.</p> <p><b>Caution:</b> When using these types of exits, carefully review the fields before updating to avoid inadvertent modifications.</p>

Positions	Field Content	COBOL Picture	Format	Length	Description
599 – 608	TRANSACTION-MAP	X(10)	AN	10	The name of the transaction map used to translate the document. This field is valid during inbound and outbound processing. It always contains valid information during element level exit processing, outbound segment level processing, and inbound application record level processing, and contains valid information during inbound segment level processing and outbound application record level processing only after the entire Header portion of the input has been loaded. <b>Caution:</b> When using these types of exits, carefully review the fields before updating to avoid inadvertent modifications.
609 – 618	APPLICATION-DEF	X(10)	AN	10	The name of the application definition used to translate the document. This field is valid during inbound and outbound processing. It always contains valid information during element level exit processing, outbound segment level processing, and inbound application record level processing, and contains valid information during inbound segment level processing and outbound application record level processing only after the entire Header portion of the input has been loaded. <b>Caution:</b> When using these types of exits, carefully review the fields before updating to avoid inadvertent modifications.
619 – 619	DIRECTION	X(01)	AN	01	Indicates the Mapper used to invoke the exit. Valid Values: <b>I</b> – Inbound Mapper <b>O</b> – Outbound Mapper
620 – 637	USER-EXIT-SOURCE-INFO	Group		18	Fields that describe the source of the mapping information. The information contained in these fields varies depending on the type of exit and the direction of the translation.

Positions	Field Content	COBOL Picture	Format	Length	Description
620 – 629	USER-EXIT-SRC-REC-ID	X(10)	AN	10	Valid values: Inbound segment level=Contains the Segment ID of the segment where the exit was invoked. Inbound record level=Not valid. Outbound segment level=Not valid. Outbound record level=Contains the record ID of the record where the exit was invoked. Inbound element level=Contains the Segment ID of the segment that contains the element to be mapped from. Outbound element level=Contains the record ID of the record that contains the field to be mapped from.
630 – 633	USER-EXIT-SRC-REC-SEQ	S9(05)	BI	04	Valid values: Inbound segment level=Contains the segment sequence number of the segment where the exit was invoked. Inbound record level=Not valid. Outbound segment level=Not valid. Outbound record level=Contains the record sequence number of the record where the exit was invoked. Inbound element level=Contains the segment sequence number of the segment that contains the element to be mapped from. Outbound element level=Contains the record sequence of the record that contains the field to be mapped from.
634 – 637	USER-EXIT-SRC-FLD-SEQ	S9(05)	BI	04	Valid values: Inbound segment level=Not valid. Inbound record level=Not valid. Outbound segment level=Not valid. Outbound record level=Not valid. Inbound element level=Contains the element sequence number of the element to be mapped from. Outbound element level=Contains the field sequence of the field to be mapped from.
638 – 655	USER-EXIT-TARGET-INFO	Group		18	Fields that describe the target of the mapping information. The information contained in these fields varies depending on the type of exit and the direction of the translation.

Positions	Field Content	COBOL Picture	Format	Length	Description
638 – 647	USER-EXIT-TGT-REC-ID	X(10)	AN	10	Valid values: Outbound segment level=Contains the Segment ID of the segment where the exit was invoked. Outbound record level=Not valid. Inbound segment level=Not valid. Inbound record level=Contains the record ID of the record where the exit was invoked. Outbound element level=Contains the Segment ID of the segment that contains the element in which to map. Inbound element level=Contains the Record ID of the record that contains the field in which to map.
648 – 651	USER-EXIT-TGT-REC-SEQ	S9(05)	BI	04	Valid values: Outbound segment level=Contains the segment sequence number of the segment where the exit was invoked. Outbound record level=Not valid. Inbound segment level=Not valid. Inbound record level=Contains the record sequence number of the record where the exit was invoked. Outbound element level=Contains the segment sequence number of the segment that contains the element in which to map. Inbound element level=Contains the record sequence of the record that contains the field in which to map.
652 – 655	USER-EXIT-TGT-FLD-SEQ	S9(05)	BI	04	Valid values: Outbound segment level=Not valid. Outbound record level=Not valid. Outbound segment level=Not valid. Outbound record level=Not valid. Outbound element level=Contains the element sequence number of the element in which to map. Inbound element level=Contains the field sequence of the field in which to map.
656 – 954	USER-EXIT-EXAMINER-AREA	Group		299	Not used.
656 – 656	XMR-PROCESS-TYPE	X(01)	AN	01	Not used.



Positions	Field Content	COBOL Picture	Format	Length	Description
657 – 668	XMR-MAPPING-STATUS-FIELDS	Group		12	Not used.
	XMR-MAPPING-STATUS (6 occurrences)	X(02)	AN	02	
669 – 908	XMR-MAPPING-CONSTANT-FIELDS	Group		240	Not used.
	XMR-MAPPING-CONSTANT (12 occurrences)	X(20)	AN	20	
909 – 953	XMR-REFERENCE	X(45)	AN	45	Not used.
954 – 954	XMR-CHANGED-FLAG	X(01)	AN	01	Not used.
955 – 1258	USER-EXIT-ERROR-HANDLING	Group		304	Data that is returned to the Mappers from an application user exit.
955 – 955	USER-EXIT-ERR-FLAG	X(01)	AN	01	Field set by the user to the value <b>Y</b> to indicate that an error occurred in the user exit.
956 – 957	USER-EXIT-MAP-STATUS	X(02)	AN	02	For future use.
958 – 959	USER-EXIT-VAL-STATUS	X(02)	AN	02	Status that the Mapper terminates with at end of processing. Valid values: <b>04</b> and <b>16</b> . Status 16 terminates the mapping process.
960 – 965	USER-EXIT-MSG-NUMBER	X(06)	AN	06	Message number that is displayed on the mapping report for the error that occurred in your user exit. This number begins with an alphanumeric character that can be distinguished from other Gentran:Basic errors.
966 – 1085	USER-EXIT-MSG-TXT1	X(120)	AN	120	Message that is displayed in the mapping report for the error that occurred in your user exit.
967 – 1205	USER-EXIT-MSG-TXT2	x(120)	AN	120	For future use.
1206 – 1206	USER-EXIT-ACTION1	X(01)	AN	01	For future use.
1207 – 1207	USER-EXIT-ACTION2	X(01)	AN	01	For future use.

Positions	Field Content	COBOL Picture	Format	Length	Description
1208 – 1208	USER-EXIT-ACTION3	X(01)	AN	01	For future use.
1209 – 1258	USER-EXIT-ERR-FILLER	X(50)	AN	50	For future use.
1259 – 34028	USER-EXIT-RECORD-AREA	Group		32770	Two fields used for the segment level and application record level exits to contain the segment or record that the exit is associated with. Valid values: Inbound segment level=Contains the input segment information just after the segment was read. Inbound record level=Contains the output application record information just before the record is written. Outbound segment level=Contains the output segment information just before the segment is written. Outbound record level=Contains the input application record information just after the record was read.
1259 – 34025	RECORD	X(32767)	AN	32767	Contains the actual record or segment data. This field can be modified by the user. If this field is modified, the UE-RECORD-LENGTH field should be modified and the length of the record changed. The UE-RECORD-CHANGED-FLAG should be set to <b>Y</b> to instruct the Mapping program that the record was changed.
34026 – 34027	RECORD-LENGTH	S9(04)	BI	02	The number of characters of valid data that are contained in the UE-RECORD field.
34028 – 34028	RECORD-CHANGED-FLAG	X(01)	AN	01	Set by the user to <b>Y</b> to indicate that a change was made to the UEW-RECORD data.
34029 – 56601	USER-EXIT-ENVELOPE-AREA	Group		22573	Valid data on return from an outbound envelope level exit.
34029 – 36578	VARIABLE-ENVELOPE (10 occurrences)	X(255)	AN	2550	If variable envelopes are generated, ten fields that contain the envelopes to be written. The exit can modify this area. If the number of envelopes to be written is changed, the new count should be set in the UE-ENVELOPE-COUNT variable. If there are any envelope changes the UE-ENVELOPE-CHANGED-FLAG should be set to <b>Y</b> .

Positions	Field Content	COBOL Picture	Format	Length	Description
36579 – 56578	FIXED- ENVELOPE (10 occurrences)	X(2000)	AN	20,000	If fixed-format (user-defined) envelopes are generated, ten fields that contain the envelopes to be written. The exit can modify this area. If the number of envelopes to be written is changed, the new count should be set in the UE-ENVELOPE-COUNT variable. If the length of any of the envelopes to be written is changed, the new length should be set in the appropriate occurrence of the UE-ENVELOPE-LENGTH variable. If there are any envelope changes, the UE-ENVELOPE-CHANGED-FLAG should be set to Y.
56579 – 56598	ENVELOPE- LENGTH (10 occurrences)	S9(04)	BI	20	Occurrences of this variable should be modified to reflect any length changes of fixed-format, user-defined envelopes that are written.
56599 – 56600	ENVELOPE- COUNT	S9(04)	BI	02	Determines how many envelopes are written.
56601 – 56601	ENVELOPE- CHANGED- RECORD	X(01)	AN	01	Set by the user to Y to indicate that a change was made to the UE-VARIABLE-ENVELOPE or UE-FIXED-ENVELOPE data.
56602 – 64521	CONSTANT-AREA	Group		7920	The 99 mapping constants that are available to the mapping system. These values can be tested or modified by the exit module. Modified constants are eligible for subsequent mapping.
56602 – 64521	CONSTANT (99 occurrences)	X(80)	AN	7920	The 99 mapping constants that are available to the mapping system. These values can be tested or modified by the exit module. Modified constants are eligible for subsequent mapping.
64522 – 64721	HASH-TOTAL- AREA	Group		200	The 20 mapping hash totals that are available to the mapping system. These values can be tested or modified by the exit module. Modified hash totals are eligible for subsequent mapping.
64522 – 64721	HASH-TOTAL (20 occurrences)	S9(18)	PD	10	The 20 mapping hash totals that are available to the mapping system. These values can be tested or modified by the exit module. Modified hash totals are eligible for subsequent mapping.

Positions	Field Content	COBOL Picture	Format	Length	Description
64722 – 64921	ACCUMULATOR-AREA	Group		200	The 20 mapping accumulators that are available to the mapping system. These values can be tested or modified by the exit module. Modified accumulators are eligible for subsequent mapping.
64722 – 64921	ACCUMULATOR	S9(18)	PD	10	The 20 mapping accumulators that are available to the mapping system. These values can be tested or modified by the exit module. Modified accumulators are eligible for subsequent mapping.
64922 – 64931	TOTAL-ITEMS	S9(18)	PD	10	The value of the reserved word constant TOTAL-ITEMS, which contains the value of the number of line items read (inbound) or written (outbound) at the current point in the process.
64932 – 64941	TOTAL-SEGMENTS	S9(18)	PD	10	The value of the reserved word constant TOTAL-SEGMENTS, which contains the value of the number of segments read (inbound) or written (outbound) at the current point in the process.

### *Inbound Control Totals File*

The following table describes the Inbound Control Totals file.

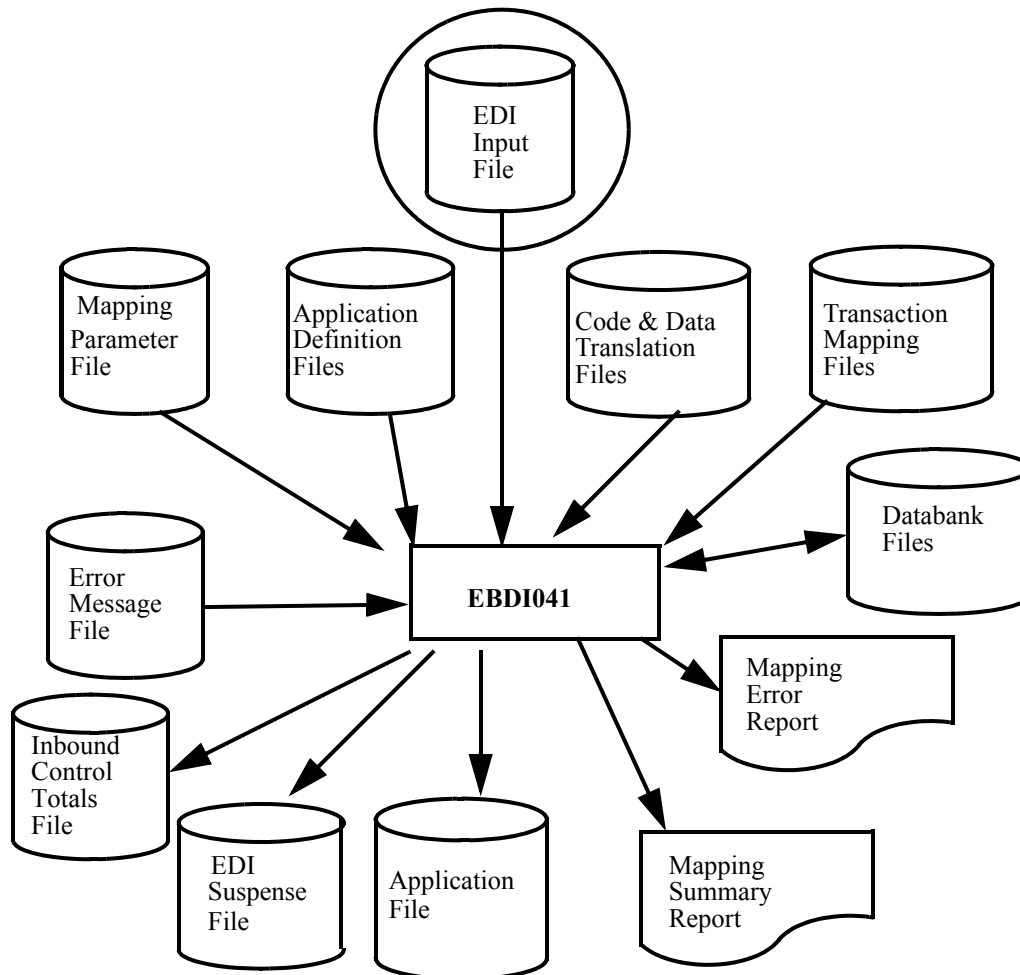
Position	Field Content	COBOL Picture	Format	Length	Description
001 – 008	Summary Start Date	9(08)	YYYYMMDD	8	The name assigned to the definition of the file. Format is Year, Month, and Day.
009 – 016	Summary Start Time	9(08)	HHMMSShh	8	This field defines the Summary Start Time. Format is Hour, Minutes, and Seconds.
017 – 024	Summary End Date	9(08)	YYYYMMDD	8	This field defines the Summary End Date. Format is Year, Month, and Day.
018 – 032	Summary End Time	9(08)	HHMMSShh	8	This field defines the Summary End Time.
033 – 047	Summary Interchanges Input	9(15)	N	15	This field defines the Summary Interchanges in the input file.
048 – 062	Summary Groups Input	9(15)	N	15	This field defines the Summary Groups in the input file.
063 – 077	Summary Transactions Input	9(15)	N	15	This field defines the Summary Transactions in the input file.
078 – 092	Summary Segments Input	9(15)	N	15	This field defines the Summary Segments in the input file.

Position	Field Content	COBOL Picture	Format	Length	Description
093 – 107	Summary Characters Input	9(15)	N	15	This field defines the Summary Characters in the input file.
108 – 122	Summary Databank Documents Output	9(15)	N	15	This field indicates the Summary Databank Documents in the output file.
109 – 137	Summary Databank Records Output	9(15)	N	15	This field indicates the Summary Databank Records in the output file.
138 – 152	Summary Application Documents Output	9(15)	N	15	This field indicates the Summary Application Documents in the output file.
153 – 167	Summary Application Records Output	9(15)	N	15	This field indicates the Summary Application Records in the output file.
168 – 182	Summary Application Characters Out	9(15)	N	15	This field indicates the Summary Application characters in the output file.
183 – 197	Summary Documents Suspended	9(15)	N	15	This field indicates Summary Documents suspended by the file.
198 – 212	Summary Records Suspended	9(15)	N	15	This field indicates the Summary Records suspended by the file.
213 – 227	Summary Characters Suspended	9(15)	N	15	The Summary characters suspended by the file.
228 – 242	Summary Applications Processed	9(15)	N	15	The Summary Applications processed by the file.
243 – 257	Summary Maps Processed	9(15)	N	15	The Summary Maps processed by the file.
258 – 272	Summary Partners Processed	9(15)	N	15	The Summary Partners processed by the file.
273 – 300	Filler	X(28)	AN	28	Not used

**Note:** A COBOL copybook has been provided in the source utility product library. The member is SUMMIWS.

### Inbound Mapping Process

Figure 2.12 illustrates the inbound mapping process.



**Figure 2.12 EBDI041 — Inbound Mapping**

### Inbound Mapper Audit Trail Report

The Inbound Mapper Audit Trail Report is generated when executing the Inbound Mapper (EBDI041) program. This report indicates errors encountered, processing options, summary control counts, and a performance summary for mapping incoming data.

#### Example Reports

Figure 2.13 through Figure 2.15 illustrate the Inbound Mapper Audit Trail Report (Errors Encountered Mapping Incoming Data), the Inbound Mapper Audit Trail Report (Processing Options for Mapping Incoming Data), and the Inbound Mapper Audit Trail Report (Summary Control Counts Mapping Incoming Data).

```

EBDI041  RUN 12/01/2005 TIME 15:42  ERRORS ENCOUNTERED MAPPING INCOMING DATA  PAGE 1
ERROR **RECORD**  FIELD SEG ELE
NUMBER  NBR ID  SEQ # ID SEQ INFORMATION  ERROR MESSAGE
INTERCHANGE: VENDOR-1          QUAL:          CONTROL NO: 000000005
GROUP      : VENDOR-1          QUAL:          CONTROL NO: 000000007
TRANSACTION: 810                CONTROL NO: 000070001
785      20 7      100          ** WARNING ** MANDATORY FIELD/ELEMENT MISSING.
689      20 7      100          DUPLICATE DOCUMENT ADDED TO DATA BANK. /INVFILE /INV01
TRANSACTION: 810                DATABANK KEY: VENDOR-1
785      40 7      100          CONTROL NO: 000070002
689      40 7      100          ** WARNING ** MANDATORY FIELD/ELEMENT MISSING.
689      40 7      100          DUPLICATE DOCUMENT ADDED TO DATA BANK. /INVFILE /INV02
DATABANK KEY: VENDOR-1

PROCESSING ENDED WITH ERRORS - PROCESSING COUNTS BELOW
EDI RECORDS READ ----- 244
EDI RECORDS SUSPENDED ----- 0
APPLICATION RECORDS WRITTEN ----- 120
RETURN-CODE FOR MAPPING ----- 4
    
```

**Figure 2.13 Inbound Mapper Audit Trail Report  
(Errors Encountered Mapping Incoming Data)**

```

EBDI041      RUN 12/01/2005 TIME 13:45      PROCESSING OPTIONS FOR MAPPING INCOMING DATA      PAGE 1
APPLICATION TO PROCESS-----INVFILE
DEBUGGING OPTION (INPUT/OUTPUT PRINT)---Y
ABEND PROGRAM ON SERIOUS ERROR-----N
USER EXIT VERSION SUPPORTED-----1
APPLICATION DECIMAL INDICATOR IS-----
RIGHT JUSTIFY ALL APPLICATION REALS-----N
HANDLE FLOATING NOTES WITHIN A SECTION--Y
DATABANK PROCESSING CONFIGURATION-----DIRECTORY AND MESSAGE STORE
DATABANK PROCESSING LEVEL-----DRIECTORY AND MESSAGE STORE
DIRECTORY POSTING OPTION-----POST SENDER ONLY
PARTNER PROFILE MODE-----PARTNER/QUALIFIER MODE
PRINT PARTNER NAME -----Y
WRITE APPLICATION RECORDS-----Y
BUSINESS DOCUMENT TRACKING-----N
SUPPORT SINGLE QUOTE -----N
VERIFY PARTNER SPECIFIC MAP VERSION-----N
    
```

**Figure 2.14 Inbound Mapper Audit Trail Report  
(Processing Options for Mapping Incoming Data)**



EBDI041 RUN 12/01/2005 TIME 16:10 SUMMARY CONTROL COUNTS MAPPING INCOMING DATA PAGE 1

PROCESSING BEGAN ON 12/01/2005 AT 4:10 PM.

INTERCHANGES READ	-----1
GROUPS READ	-----1
TRANSACTIONS READ	-----6
SEGMENTS READ	-----210
CHARACTERS READ	-----62,464
DOCUMENTS STORED ON DATA BANK	-----6
RECORDS STORED ON DATA BANK	-----120
APPLICATION DOCUMENTS WRITTEN	-----6
APPLICATION RECORDS WRITTEN	-----120
APPLICATION CHARACTERS WRITTEN	-----9,600
DOCUMENTS SUSPENDED	-----0
RECORDS SUSPENDED	-----0
CHARACTERS SUSPENDED	-----0
NUMBER OF APPLICATIONS PROCESSED	-----1
NUMBER OF MAP DEFINITIONS PROCESSED	-----1
NUMBER OF TRADING PARTNERS PROCESSED	-----1

PROCESSING ENDED ON 12/01/2005 AT 4:10 PM.

**Figure 2.15 Inbound Mapper Audit Trail Report  
(Summary Control Counts Mapping Incoming Data)**

## *Field Definitions*

### **Errors Encountered Mapping Incoming Data**

Error Numbr:

This field displays the error code number associated with the error message.

Record Nbr:

This field displays the application record number.

Record ID:

This field displays the application record identifier.

Field Seq #:

This field displays the field sequence number.

Seg ID:

This field displays the EDI segment identifier.

Ele Seq:

This field displays the EDI element sequence number.

Information:

This field displays descriptive information identifying additional information about the error message.

Error Message:

This field displays descriptive information identifying the error message.

(Message):

This field displays a message regarding the error or the statement “no errors occurred during processing.”

EDI Records Read:

This field displays the total number of EDI records read.

EDI Records Suspended:

This field displays the total number of EDI records suspended.

Application Records Written:

This field displays the total number of application records written.

Return–Code for Mapping:

This field displays the return code for mapping.

### **Processing Options for Mapping Incoming Data**

This section of the report displays all the parameter options in effect for mapping incoming data. Only those parameter options used are displayed.

**Summary Control Counts Mapping Incoming Data****(Message):**

This field displays a text message indicating the date (in MM/DD/YYYY format) and the time (in HH:MM format) that processing began.

**Interchanges Read:**

This field displays the total number of interchanges read.

**Groups Read:**

This field displays the total number of groups read.

**Transactions Read:**

This field displays the total number of transactions read.

**Segments Read:**

This field displays the total number of segments read.

**Characters Read:**

This field displays the total number of characters read.

**Documents Stored on Data Bank:**

This field displays the total number of documents stored on the Data Bank.

**Records Stored on Data Bank:**

This field displays the total number of records stored on the Data Bank.

**Application Documents Written:**

This field displays the total number of application documents written.

**Application Records Written:**

This field displays the total number of application records written.

**Application Characters Written:**

This field displays the total number of application characters written.

**EDI Documents Suspended:**

This field displays the total number of EDI documents suspended.

**EDI Records Suspended:**

This field displays the total number of EDI records suspended.

**EDI Characters Suspended:**

This field displays the total number of EDI characters suspended.

**Number of Applications Processed:**

This field displays the total number of applications processed.

**Number of Map Definitions Processed:**

This field displays the total number of map definitions processed.

**Number of Trading Partners Processed:**

This field displays the total number of trading partners processed.



# Outbound Processing Programs

## Overview

This chapter contains detailed descriptions of Gentran:Basic outbound processing programs. Its purpose is to enable you to familiarize yourself with the system, its terminology, and functional requirements.

This chapter contains the following topics:

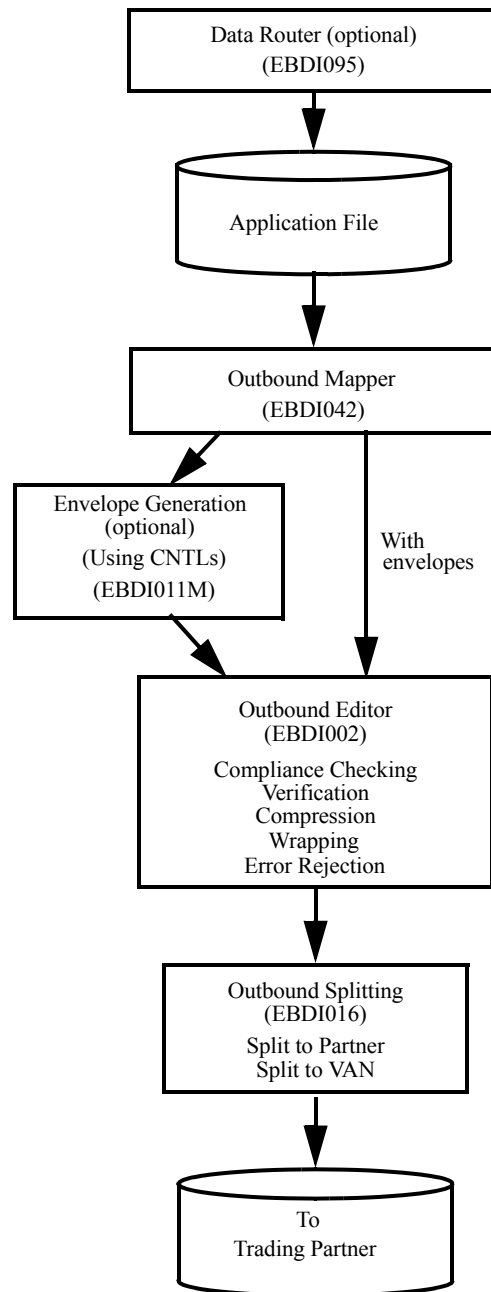
Topic	Page
<b>System Flow</b> .....	<b>3-3</b>
<b>Outbound Data Mapping EBDI042</b> .....	<b>3-5</b>
Processing Overview.....	3-5
Processing Detail – User (Sender) and Partner (Receiver) Determination.....	3-8
Processing Detail – Multiple Envelope ID .....	3-10
Processing Detail – Blank Partner Error Parameter.....	3-10
Processing Detail – Transaction Definition Map Determination.....	3-10
Processing Detail – Sectioning .....	3-11
Processing Detail – Enveloping .....	3-12
Processing Option – Instream Application Parameters.....	3-17
Processing Option – Databanking.....	3-18
Processing Option – Partner Read Sequence .....	3-19
Processing Option – User Exits .....	3-19
Processing Option – Map Blank Sub-Fields.....	3-21
Processing Option – Pass EDI IDs to Outbound Editor.....	3-21
Processing Option – EDI Data Print .....	3-22
Processing Option – Return Code Setting.....	3-22
Processing Option – \$TCR Transmission Control Record .....	3-22
Processing Option – Gentran:Plus .....	3-23
Processing Option – Truncated Sender’s Reference (TRADACOMS) (Gentran:Plus only).....	3-23
Processing Option – Gentran:Viewpoint.....	3-23
Files Used.....	3-24
Record Layouts .....	3-29
Enveloping Parameter Records (SYS000).....	3-33
Outbound Mapper Audit Trail Report.....	3-57

<b>Outbound Envelope Generation EBDI011M, EBDI011A.....</b>	<b>3-63</b>
Envelope Generation.....	3-63
Selectable Feature – TDDC Numbering.....	3-63
Envelope Processing.....	3-65
Files Used.....	3-69
Envelope Processing Parameters – SYS000.....	3-70
<b>Outbound Editor EBDI002.....</b>	<b>3-77</b>
Global Parameters.....	3-77
Sequence of Processing.....	3-79
Global Parameters Processing.....	3-80
Compliance Checking to Standards.....	3-80
Document Compliance.....	3-84
Error Rejection Processing.....	3-85
Error Rejection with the Databank.....	3-88
Error Rejection Numbers.....	3-89
Processing Methods.....	3-90
Interchange Envelope Processing (Standard Enveloping).....	3-91
Global Parameter Overrides.....	3-93
Group Envelope Processing (Standard Enveloping).....	3-94
Transaction Envelope Processing (Standard Enveloping).....	3-95
Object Envelope Processing (Standard Enveloping).....	3-95
Segment Processing (Standard Enveloping).....	3-96
Trailer Envelope Processing (Standard Enveloping).....	3-96
Control Record Processing (Deferred Enveloping).....	3-96
Segment Processing (Deferred Enveloping).....	3-97
Databank Processing.....	3-98
Viewpoint Processing.....	3-99
Output Processing.....	3-99
Network Tracking.....	3-103
Databank Network Extract.....	3-104
Files Used.....	3-105
Outbound Editor Reports.....	3-111
<b>Outbound Split EBDI016.....</b>	<b>3-119</b>
Outbound Split Processing.....	3-119
Files Used.....	3-120
Error Report.....	3-121
Outbound Split Process.....	3-122
Outbound Split Processing Parameters.....	3-123
Runtime System Log List.....	3-124
Miscellaneous Output Summary.....	3-124

## System Flow

The section illustrates various system flows, depending on how your system is set up. Several variations of the outbound processing exist. Descriptions of the processes are shown on the following flowcharts.

Figure 3.1 illustrates the outbound processing flow with mapping and with no databank message store. Use the optional programs according to your own processing needs. For detailed information on the programs illustrated, see the appropriate sections in this guide.



**Figure 3.1 Outbound Processing – With Mapping – No Databank Message Store**

Figure 3.2 illustrates the outbound processing flow, with mapping and with a databank message store. Use the optional programs according to your own processing needs. For detailed information on the programs illustrated, see the appropriate sections in this guide.

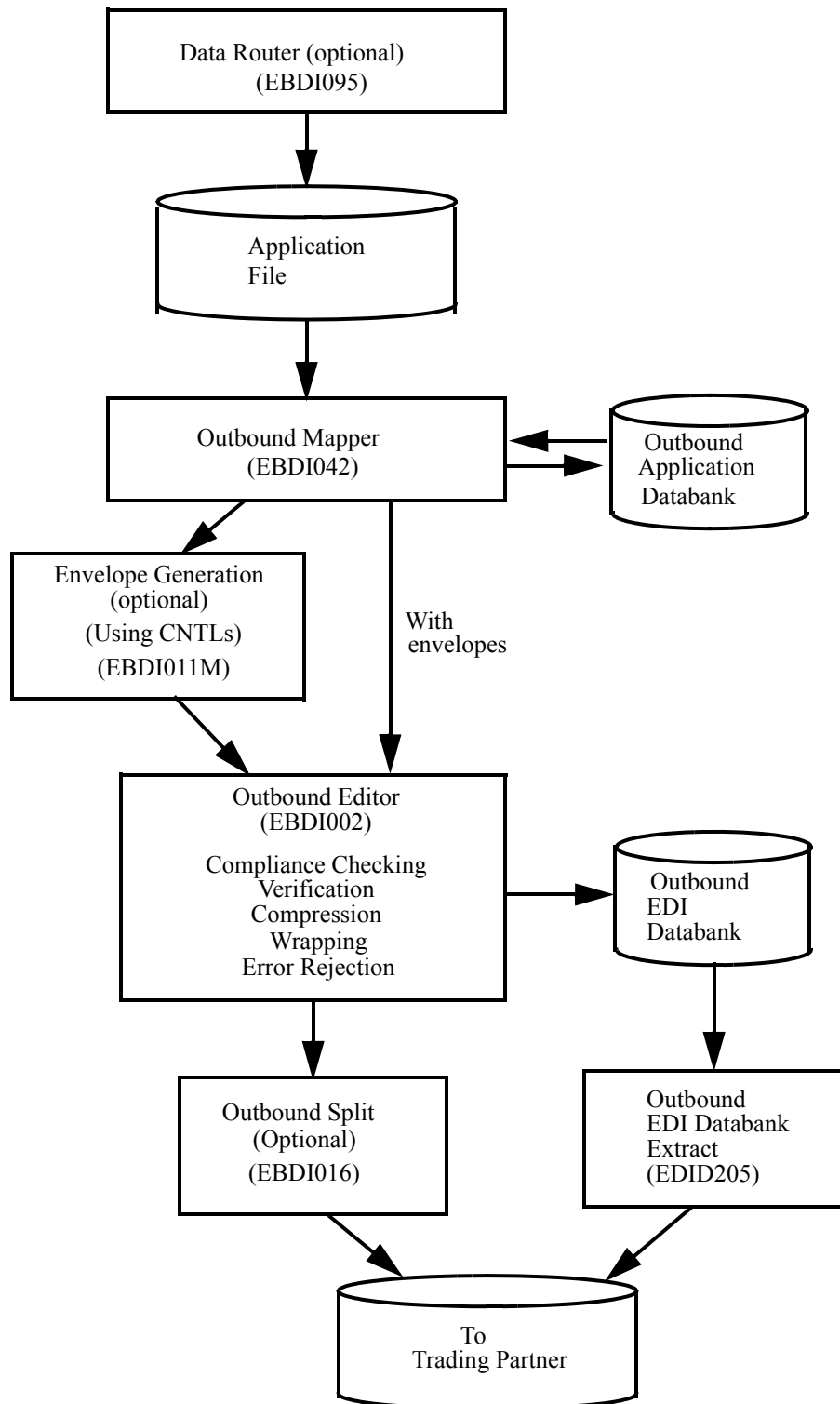


Figure 3.2 Outbound Processing – With Mapping – With Databank Message Store



---

## Outbound Data Mapping

**EBDI042**

The Outbound Data Mapping (EBDI042) program is the first step in the outbound translation process. This program reads the information in the Application Data file and translates it into EDI format for further processing. Definitions established through the Mapping Maintenance subsystem in Gentran:Basic will translate the Application Data file.

### Processing Overview

#### *Step 1: Select and Load the Application Definition*

For consistent data, an application data input file described by a single application definition, the input parameter record will directly provide the name of the application definition. This method of specifying the application definition to use is sufficient in the majority of situations.

If, however, the data in the Application Data file needs to be processed by more than one application definition, you must use Instream Parameter records (i.e., ###PARM records) to indicate which application definition to use. The application definition then needs placed in the data file immediately preceding the data described by the application definition, which you specified on the Instream Parameter record.

**Note:** When using the Instream Parameter option, you can avoid performance problems by sorting data by application before submitting it to the Outbound Mapping program.

If the program cannot select the application definition, a fatal error will occur and processing stops unless you use Instream Parameter records. When you use Instream Parameter records, the return code value on the Error Control file will mandate the result of an inconsistent application definition. Possible results include:

- Generate a fatal error
- Suspend the document(s), and then writes it to the application Suspense file.

After the selection of the application definition, all information pertaining to that application definition loads from the online mapping files into memory.

#### *Step 2: Load Header Section*

After loading the application definition, the program reads all Application Data records defined in the Header section of the definition, and then loads them into memory.

If you have selected the Databank Message Store option, the program stores the Header Application Data records at this time.

### Step 3: Load Partner Profiles

The Outbound Mapping program loads partner profiles depending on the trading profile mode selected from the Configuration file.

- *Partner/Qualifier.* The program attempts to read the partner profile for the specified Partner ID (receiver). If the partner is not found, the application document is suspended and written to the application Suspense file.
- *Relationship.* The program attempts to read the partner profile for the specified user/partner (sender/receiver) combination. If the user/partner combination is not found, the application document is suspended and written to the application Suspense file.
- *Mixed.* The program attempts to read the partner profile for the specified Partner ID (receiver). If the Partner ID is not found, the application document is suspended and written to the application Suspense file. Next, the program attempts to read the partner profile for the specified User ID (sender). If the user is not found, an error is generated but processing continues.

### Step 4: Select and Load Transaction Definition (Map)

Based on both the partner profile definition and the application data, the program selects the transaction definition used to translate the data. If the program cannot determine the transaction definition, the application document is suspended and written to the application Suspense file.

After selection of the transaction definition, all of the information pertaining to that transaction definition loads from the online mapping files into memory.

### Step 5: Envelope / CNTL Record Generation

Runtime parameters determine the type of envelope processing used by the Outbound Mapper. The program will do one of the following:

- Generate beginning envelopes required for the selected transaction and defined the partner profile.
- Generate CNTL records to be used by the Envelope Generation program or the Outbound Editor (i.e., if Deferred Enveloping is used).
- Pass through application generated CNTL records to be used by the Envelope Generation program or the Outbound Editor.

### Step 6: Process the Header Section

For the Header application data held in memory, the Outbound Mapper generates the EDI data defined in the Header section of the transaction definition.

### Step 7: Process the Detail Section

The Mapper processes the Detail section of the document (i.e., each iteration of the defined detail loop loads into memory, and then is mapped to the transaction definition, one at a time). If the Mapper cannot load an entire line item into memory at once, an error message is issued and processing terminates. If this error occurs, you must *section* the line item.

See the topic “Processing Detail –Sectioning” in this section for more information about how to section a document.

If you have selected the Databank Message Store option, the program stores the Detail Application Data records at this time.

### ***Step 8: Process the Summary Section***

The program reads the Summary section of the document, loads it into memory, and then writes the summary EDI data.

If you have selected the Databank Message Store option, the program stores the Summary Application Data records at this time.

### ***Step 9: Generate Envelopes***

If the Runtime parameters indicate that the Mapper should generate envelopes, the program writes the required ending envelopes at this time.

If you have selected the Databank feature, the program writes the directory entry for the application document.

### ***Step 10: Process All Documents***

The program repeats Steps 1 through 9 until translation completes for all Application Documents.

If you have selected the Databank reprocessing option, the Databank retrieves all selected application documents and processes them through the Mapper.

### ***Step 12: Summary Reports***

The Outbound Mapper produces an audit trail and summary reports. A return code is set that can be tested for in the job stream.

## Processing Detail – User (Sender) and Partner (Receiver) Determination

The term *user* in outbound processing refers to the sender, or the originator of the application data. Often, a user has multiple divisions within its company. Depending on the requirements of a user, you may need to separate information and track by divisions. Consideration for the user occurs only when the trading profile mode on the Configuration files is set to **Relationship** or **Mixed**.

**Note:** The term *partner* in outbound processing refers to the receiver of EDI data.

### *Relationship Mode*

Specify the user using one of these three methods:

- On the second parameter record in the Mapping Parameters file (SYS001). The User ID is specified in columns 40 through 54.
- On a CNTL record generated by the application. The User ID must be placed in columns 5 through 19 of the CNTL record.

**Note:** A value other than spaces on the CNTL record will override any value found on the SYS001 parameter record.

- In the application data. To access user information from the application data, the User Field Name must be defined on the Application Partner Reference screen.

**Note:** The User ID is pulled from the application data only if the User ID on the SYS001 parameter record and the CNTL record (if used) are spaces.

Specify the partner using one of these three methods:

- On the second parameter record in the Mapping Parameters file (SYS001). The Partner ID is specified in columns 1 through 15.
- On a CNTL record generated by the application. The Partner ID must be placed in columns 20 through 34.

**Note:** A value other than spaces on the CNTL record will override any value found on the SYS001 parameter record.

- In the application data. To access partner information from the application data, the Partner Field Name must be defined on the Application Partner Reference screen.

**Note:** The Partner ID is pulled from the application data only if the Partner ID on the SYS001 parameter record and the CNTL record (if used) are spaces.

The User ID and Partner ID are combined to build the key used to access the partner profile. If the user/partner combination can not be found on the partner profile, the application document is suspended.

### *Partner/Qualifier and Mixed Mode*

On the Configuration file, be sure to set the trading profile mode to **Mixed** if you want to specify a user. Then, specify the user using one of these two methods:

- On the second parameter record in the Mapping Parameters file (SYS001). The User ID is specified in columns 40 through 74 and the User ID Qualifier is specified in columns 75 to 78, if needed.
- In the application data. To access user information from the application data, the User Application fields must be defined on the Application Partner Reference screen.

**Note:** The User ID is pulled from the application data only if the User ID on the SYS001 parameter record is spaces.

After determining a User ID, the partner profile is accessed using user information as a key. If necessary, the partner profile cross-reference is accessed. If the ID for the user cannot be found on the partner profile, an error is generated, yet processing continues.

Specify the Partner using one of these three methods:

- On the second parameter record in the Mapping Parameters file (SYS001). The Partner ID is specified in columns 1 through 35 and the Partner ID/Qualifier is specified in columns 36 through 39, if needed.
- On a CNTL record generated by the application. The Partner ID must be placed in columns 5 through 39 and the Partner ID Qualifier in columns 40 through 43, if needed.

**Note:** A value other than spaces on the CNTL record will override any value found on the SYS001 parameter record.

- In the application data. To access partner information from the application data, the Partner Application fields must be defined on the Application Partner Reference screen.

**Note:** The Partner ID is pulled from the application data only if the Partner ID on the SYS001 parameter record and the CNTL record (if used) are spaces.

After determining a Partner ID, the partner profile is accessed using partner information as a key. If necessary, the partner profile cross-reference is accessed. If the Partner ID cannot be found on the partner profile, the application document is suspended.

At this point, (no matter which trading profile mode is being utilized) the partner profile has been selected at the *control* (or *interchange*) level to be used to process the outbound document. The next step is to determine which group and transaction level partner profile records should be used for processing. The partner group and transaction records are used to determine the selection of the transaction definition (map) to process. The group record that is used is the first group record that has a Group ID equal to one of the Functional Group IDs on the application definition. The group records are accessed in the sequence that they are defined to the application. If no specific group records are located, the default group record is used. If the CNTL pass-through option is in effect,

as with the interchange level, the partner's group and transaction information is loaded based on the CNTL records.

**Note:** The *Multiple Envelope IDs* and *Partner Version* options impacts which partner profile is selected. Please see the Tips and Techniques chapter in the *User's Guide* for more information on using these two features.

### Processing Detail – Multiple Envelope ID

The Multiple Envelope ID feature allows a single partner profile to have multiple EDI standards (i.e., X12, EDIFACT, TRADACOMS) defined under it. This eliminates the need to create multiple partner profiles for a single trading partner in the event that the partner uses more than one EDI standard. This feature is turned on globally in the Configuration file. Refer to the Tips and Techniques chapter in the *User's Guide* for more information on using this feature.

### Processing Detail – Blank Partner Error Parameter

When the Partner ID is passed to the Mapper through the application data and the application field that holds the Partner ID is blank, the Mapper uses the Partner ID from the previous document. A parameter named Blank Partner Error is provided to prevent the Mapper from using the partner from the previous document. The Blank Partner Error parameter is located on the first parameter record in the Mapping Parameters file (SYS001) in column 64. If column 64 is set to **Y** or **1**, the Mapper displays an error message indicating that the partner is not found on the partner profile and suspends the application data.

### Processing Detail – Transaction Definition Map Determination

Once the partner group and transaction records have been selected, the process to select a transaction definition map begins. The selection of the transaction definition map is based on the application definition and the selected partner group and transaction records. When the application definition is loaded, a list of all maps associated with that application are loaded into memory. All transaction maps linked to the application definition are evaluated until a transaction map is selected. Partner-specific maps are evaluated first and then general maps are evaluated, until a match is found.

General Map	These maps are not associated to any specific partner.
Partner-Specific Map	These maps are only selected after their name is defined on the specific Partner Transaction record (located on the partner profile) in the Outbound Translation ID field.

**Note:** For the "UNO," Object Header segment, the Transaction Set ID on the partner profile must be set to "UNO" to use a partner-specific map.

The selection process is outlined below:

1. First, the Mapper attempts to find a partner-specific map. The list of all available maps is scanned for the first partner-specific map. If one is found, the Mapper accesses the partner profile using the Transaction ID defined for the map as the key.

**Note:** For TRADACOMS users, the combination of a Transaction ID and release character is used for the partner profile lookup. If a record for that particular release does not exist, the Transaction ID is used as the exclusive key.  
For all users, the version portion of the Transaction ID is not used during map selection.

2. If a partner Transaction record is found and the map name specified in the Translation ID Outbound field on the partner record matches the partner-specific map name, and the value of the data contained in the Application Select fields (if used) is one of those defined for this map, the map is selected. If no match is found, the Mapper continues this process of reviewing all partner-specific maps until a map is selected or there are no more partner-specific maps associated with the application definition being processed.
3. If the system could not select a partner-specific map, an attempt is made to select a general map. The list of all available maps is scanned for the first general map. If one is found, the compliance version on the functional group record of the partner profile is compared to the version that is defined for the map. If these match and the value of the data contained in the Application Select field (if used) is one of those defined for the map, the map is selected. If the map is not selected, the entire step is repeated until all general maps have been reviewed. If map selection fails, the application document is suspended.

## Processing Detail – Sectioning

The option exists to divide the Detail portion of a document, such as a document where the defined Detail area contains deeply nested loops, into multiple parts or *sections*. The mapping program attempts to load one iteration of the Detail (commonly called a line item) into memory at once. The reserved area in the program to hold this information is limited to 1 million characters. If the document exceeds the limit, the program issues an error message and terminates processing. When you encounter an error, you must section the Detail area of the document by using the online Mapping Facility.

**Note:** To learn more about the Sectioning feature, refer to the Mapping Subsystem Chapter in the *User's Guide*.

Before sectioning the Detail level of a document, carefully consider the following restrictions:

- You cannot perform mapping between sections (e.g., Header to Detail). You can, however, move the data from a higher level to a lower level by saving the desired application data fields to reserved word constants. **Data cannot be mapped from a lower section to a higher section.**
- No attempt is made to map data to a section 1 through 9 unless some input is received that is in that section. This means that you cannot generate an outbound segment that is composed of all literals, unless an input record is received that is in the corresponding section. This can cause problems for existing maps where LS and LE segments were being generated and there was no corresponding input.
- All data defined for a section must be contiguous both when the data is read and when the data is written.

## Processing Detail – Enveloping

The Outbound Data Mapping program produces EDI output with or without the EDI control segments (envelopes). Generating envelopes is controlled at run-time by means of a parameter setting in column 80 on the second Mapping Parameter record in the Mapping Parameter file (SYS001). If envelopes are generated by the mapping program, the resulting data is ready to be processed by the Outbound Editing (EBDI002) program. If envelopes are not generated by the mapping program, the resulting EDI data can be further processed by the Envelope Generation (EBDI011M) program or can be passed into the Editor, using the Deferred Enveloping (see the Error Rejection with the Databank topic in this chapter for more information). Performance of the outbound process is better if the Outbound Mapper generates envelopes.

### *Map Envelope Elements from Application*

On the Application Envelope Definition screen, you can define application fields that are mapped directly to the EDI envelope data elements. Access the Application Envelope Definition screen, to determine which fields you can define.

Before defining application fields, carefully consider the following:

- Defined application fields that need to be moved to the envelopes must be text fields. Gentran does not move blank application fields. The application field must have a data type of AN (alphanumeric).
- Gentran truncates the data on the right when text in the application field exceeds the character length of the field before moving this information directly to the EDI element.
- Gentran does not perform data type conversion or zero filling. The field should be defined with the correct length.

The one exception to this rule is the test/production indicator. Only a **T** or **P** value should be provided. All others are ignored.

**Note:** These values are only loaded when the Partner ID changes and a new interchange envelope is generated.

To use this option, the Mapper must generate the envelopes.

### *Determining Test/Production Status*

Determining the test/production status of an interchange is crucial if ISA, UNB, or STX interchanges are being generated. Each of these interchanges contains a data element that indicates the test/production status of the data within that interchange. GS only, BG, and ICS interchanges do not have a test/production indicator.

- ISA has a test/production indicator that contains the value **T** or **P** to indicate the status of the interchange.
- UNB has a test/production indicator that contains the value **1** for test or **0** for production to indicate the status of the interchange.



- STX has an Application Reference data element that normally contains the message identifier of the first message in the interchange (e.g., ORDHDR). If an interchange is in test status, characters four through six are replaced with the characters TES. Processing a batch of test orders causes this Application Reference data element to contain the characters ORDTES.

The following tests are performed by the Outbound Mapper to determine what should be mapped to the interchange to indicate the test/production status.

#### **ASC X12 Interchanges (ISA)**

- If CNTL records are being passed from the application, and **T** or **P** is coded on the CNTL record in column 59, that status is mapped to the interchange. Only the interchange level CNTL record is used for this determination.

*See the CNTL record layout in this section for additional information.*

- If the test or production indicator is mapped from the application, and the application field being mapped equals **T** or **P**, that status is mapped to the interchange.
- If the test/production status of the map, the partner profile Control record, or the partner profile Transaction record of the first message in the interchange equals **T**, the interchange is mapped with a status of **T**. If all of the above are coded with **P**, the interchange is determined to be in production. Be advised that the partner profile Transaction record that is queried is the specific record defined for the transaction set, if it exists. If the specific transaction record does not exist, the default record is used.

#### **EDIFACT Interchanges (UNB)**

- If CNTL records are being passed from the application, and **T** or **P** is coded on the CNTL record in column 59, the status is mapped to the interchange.

**Note:** The value **1** indicates a test status, while **0** indicates production status. Only the interchange level CNTL record is used for this determination.

- If the test or production indicator is mapped from the application, and the application field being mapped equals **T** or **P**, the status is mapped to the interchange.

**Note:** The value **1** indicates a test status, while **0** indicates production status.

- If the test/production status of the map equals **T**, the partner profile Control record status equals **1**, or the partner profile Transaction record (of the first message in the interchange) equals **T**, the interchange is mapped with a status of **1**. If all of the above are coded with a production status, the interchange is determined to be in production, and **0** is mapped. Be advised that the partner profile Transaction record that is queried is the specific record defined for the message, if it exists. If the specific transaction record does not exist, the default record is used.

**TRADACOMS Interchanges (STX)**

- If CNTL records are being passed from the application, and **T** or **P** is coded on the CNTL record in column 59, that status is mapped to the interchange.
  - Note:** Characters four through six changes to TES for test interchanges to indicate a test status. Otherwise, production status results. Only the interchange level CNTL record is used for this determination.
- If the test or production indicator is mapped from the application, and the application field being mapped equals **T** or a **P**, that status is mapped to the interchange.
  - Note:** Characters four through six changes to TES for test interchanges to indicate a test status. Otherwise, production status results.
- If the test/production status of the map equals **T** or the partner profile Transaction record of the first message in the interchange equals **T**, the interchange is mapped with a status to indicate test. If all of the above are coded with a production status, the interchange is determined to be in production. Be advised that the partner profile Transaction record that is used is the specific record (including release character) defined for the message, if it exists.
  - Note:** Proper envelope generation for TRADACOMS requires that the specific partner profile Transaction records for each message to be generated must be defined. For this reason, the default transaction record is not used.

**Control Number Generation**

To improve performance, the Mapper generates the envelopes needed for the EDI data. This eliminates the need for a separate envelope generation step.

The envelopes are generated with the information provided on the interchange control, group, and transaction records of the trading partner. Optionally, some of this data is generated by the application, and overrides the partner defaults.

*See the topic “Map Envelope Elements From Application” on page 3-12 of this guide for more details.*

Control numbers at all three levels can be provided in the Application file or on CNTL records. These numbers are moved directly to the envelopes.

Control numbers can be automatically determined by the partner records at the appropriate levels. Control numbers are automatically incremented by one at the interchange, group, and transaction level.

### *CNTL Record Generation*

The Outbound Mapper generates CNTL records instead of envelopes for the EDI documents if column 80 on the second Mapping Parameter record in the Mapping Parameter file contains the value **Y**. The resulting data and CNTL records must be processed through the Envelope Generation (EBDI011M) program before being edited.

This option should only be used if absolutely required for the implementation of the outbound process. Additional overhead is incurred in the subsequent envelope generation step.

The Mapper generates an I-Type (interchange) CNTL record and a G-Type (group) CNTL record each time the partner changes, and a T-Type (transaction) CNTL record for each document in the Application file.

### *CNTL Record Pass-Thru*

A CNTL record Pass-Thru option is provided in the outbound mapping process. The application may provide CNTL records in the application data where envelopes should be generated.

If the application being mapped contains object (binary) data, then CNTL record pass-thru is required. The CNTL record that precedes the UNO Header record must contain the type code **O**.

If column 79 on the second Mapping Parameter record in the Mapping Parameter file (SYS001) is set to **Y**, the user must generate all of the CNTL records in the application that are needed to generate the required EDI envelopes.

**Note:** The following option is currently supported for the ANA TRADACOMS Standard only.

If column 79 is set to **I**, the user must generate only the I level CNTL records in the application that are needed to generate interchange envelopes. The Mapper generates the G- and T-level CNTL records as required.

If different Partner IDs are needed to produce the interchange, group, and transaction envelopes, the application (with the appropriate Partner ID at each level) must generate CNTL records, which are passed through the Outbound Mapper (EBDI042) program and processed by the Envelope Generation (EBDI011M) program. The Outbound Mapper uses the Partner ID from the I-level CNTL record to generate all envelopes (interchange, group, and transaction). The Envelope Generator uses the Partner ID from each CNTL record to generate the corresponding envelope (e.g., the Partner ID on the G-level CNTL record is used to produce the group envelope).

The CNTL Pass-Thru option instructs the Mapper only on the format of the input Application file. Either outbound enveloping option can be used by setting the Envelope Generation switch in column 80 of the second Mapping Parameter record in the Mapping Parameter file (SYS001).

### *TDCC Numbering*

TDCC Standards (i.e., UCS, WINS, AIR, MOTOR and RAIL) allow for special numbering of the transaction segments. These control numbers are a combination of the low-order characters in the group control number and an incrementing number for each transaction within the group.

**User UCS Numbers**

The USE UCS NUMBERS Runtime parameter generates special numbers for every document with a specified EDI version. The following table shows how transaction control numbers are generated for different version types:

<b>Version Type</b>	<b>How Control Numbers are Generated</b>
Rail, Motor, Air	A nine-digit number is generated with the first five digits of the number resulting from the last five digits of group control number. The last four digits of the number are incremented by one within the group.
Wins	An eight-digit number is generated with the first five digits of the number resulting from the last five digits of group control number. The last three digits of the number are incremented by one within the group.
UCS	An 8- or 9-digit number is generated with the first five digits of the number resulting from the last five digits of the group control number. The last three or four digits are incremented by one within the group. <b>Note:</b> For UCS versions 003050UCS, 003060UCS, 003070UCS, 004010UCS and 004020UCS, a 9-digit transaction control number is generated. For all other UCS versions, an 8-digit transaction control number is generated.
All other versions	A nine-digit number is generated with the first six digits of the number resulting from the last six digits of group control number. The last three digits of the number are incremented by one within the group.

Because of the consolidation of the TDCC Standard within ANSI, the following special numbering parameters are provided at the transaction set level for all EDI versions processed or for specific EDI versions. For example, special numbering may be required for an 875 document for ANSI version 003050, but not for an 875 document for ANSI version 003060.

**Use 5/4 Numbers**

The USE 5/4 NUMBERS Runtime parameter generates special numbers for every document that has the specified transaction set identifier for all EDI versions processed or specific EDI versions as follows:

A nine-digit number is generated with the first five digits of the number coming from the last five digits of group control number. The last four digits of the number is incremented by one within the group.

**Use 5/3 Numbers**

The USE 5/3 NUMBERS causes special numbers to be generated for every document that has the specified transaction set identifier for all EDI versions processed or specific EDI versions as:

An eight-digit number is generated with the first five digits of the number coming from the last five digits of group control number. The last three digits of the number are incremented by one within the group.

**Use 6/3 Numbers**

The USE 6/3 NUMBERS Runtime parameter generates special numbers for every document that has the specified transaction set identifier for all EDI versions processed or specific EDI versions as follows:

A nine-digit number is generated with the first six digits of the number coming from the last six digits of group control number. The last three digits of the number are incremented by one within the group.

*See the Enveloping Parameter Records (SYS000) program in this guide for record layouts.*

**For “RAIL” Documents Only**

The Outbound Mapper uses the partner profile to determine whether to generate a Rail ISA interchange. If the interchange envelope to be generated is an ISA and the Authorization Information Qualifier ISA01 contains 04, the Mapper generates a Rail ISA. Both methods for generating envelopes from the Outbound Mapper support the Rail ISA.

If the ISA01 contains 04 on the partner profile, the Mapper generates an ISA02 value in the format of SWxxx, where xxx represents the transaction type (e.g., 426, 850). If the Mapper is to generate CNTL records, the Swxxx value is passed on the CNTL Control record for input into the Mapper Envelope Generation (EBDI011M) program.

**GS Envelopes – Generating the GS05**

It is possible to generate the time element in the GS segment (i.e., GS05) with varying lengths. The default is to generate GS envelopes with a four-digit GS05 time for all versions. Using the parameter GENERATE MAX LENGTH GS TIME in the SYS000 parameter record, the GS envelope record will be generated with the GS05 time field at the maximum length allowed according to the standard. If this parameter is used, ANSI versions 003020 through 003030 will have a six-digit GS time field and version after 003030 will have an eight-digit GS time field. Versions prior to 003020 will have a four-digit GS time field.

**Processing Option – Instream Application Parameters**

If only one application is to be processed during the outbound mapping run, the application can be specified in columns 1 to 10 on the first mapping parameter record on the mapping parameter file (SYS001).

If using Instream Parameters, the Application Data ID specified in columns 1 to 10 on the first parameter record in the Mapping Parameters file (SYS001) equals ##INSTREAM. The Mapper then assumes that instream parameter records are placed in the Application file to instruct the Mapper to which application definition to load. This feature enables you to process different application definitions using one pass of the Mapper.

*See the topic “Mapping Parameter Records (SYS001)” later in this section for the ##INSTREAM (SYS001) record layout.*

An instream parameter record (###PARM) must be the first record on the Application file. Additional parameter records should be placed before each new application document type if a new application definition is to be loaded.

See the topic “Instream Parameter Record Layout” later in this chapter for the ####PARM instream parameter records.

If the system does not find the application definition, a GT639 error is generated (Unable to find application data ID specified in parameters) and processing ends immediately with a return code of 16. If the Mapper finds a return code of less than 08 on the Error Control file for error number 639, the Mapper suspends the bad application data and continues processing any remaining applications. You must change the return code for error number 639 to a value less than 08 to enable this feature. You can change the return code on the Message Maintenance screen. Return code 1 is used for inbound processing and return code 2 is used for outbound processing.

## Processing Option – Databanking

### *Application Databank Facility*

Application Databank provides the facility so that application data can be tracked and stored as it is processed by the Outbound Mapping program.

Data tracking is accomplished using the Databank Directory; data storage is accomplished using the Databank Message Store. Databanking can be performed at two levels, or turned off completely.

Directory Only	Only the databank directory entry is created. Document tracking is performed but data storage is not.
Full	The databank directory and message store entries are created. Document tracking and data storage are performed.

The databank level for the run is set using a combination of two switches on the first Mapping Parameter record: the Databank Configuration switch and the Databank Processing Level switch.

The Databank Configuration switch tells the Mapper how to initialize the databank, so it should be set to the highest level that might be requested. The Databank Processing Level identifies to the Mapper the lowest level of databank processing required. Partner requests at a higher level are performed. If there is a request for a databank operation at a higher level than the configuration setting, an error is generated and processing terminates.

The Databank Configuration Switch, column 36 of the first Mapping Parameter record in the Mapping Parameter file (SYS001), tells the Mapper how to initialize the databank. This Configuration switch must be set to the most detailed level of databanking that might be requested. The Databank Processing Level Switch, column 37 of the first Mapping Parameter record in the Mapping Parameter file (SYS001), identifies the level of databanking to be used for the run. If the Databank Processing Level Switch requests a databanking level more detailed than the configuration setting, an error is generated and processing terminates.

### Examples

When the Databank Configuration switch is set to **Full** and the Databank Processing Level switch is set to **Full**, databank processing is performed for all documents in the run.

When the Databank Configuration switch is set to **Directory Only** and the Databank Processing Level switch is set to **No Databank**, no databanking is performed for any documents in the run.

If the Databank Processing Level Switch indicates Partner Databank Level, the Outbound Application Databank Level field on the Transaction record of the partner profile determines the level of databanking.

**Note:** Full databanking cannot be specified by partner profile.  
To use the Message Store feature, the Databank Processing Level Switch must specify Full databanking.

Using 'Full' Outbound Application Databanking provides you with the ability to edit the data in an application document and to mark selected application documents for reprocessing via the online Databank screens. The selected data is reprocessed by the first outbound mapping run that has the Runtime parameter for Databank reprocessing active. The Databank Reprocess Switch is in column 38 of the first Mapping Parameter record in the Mapping Parameter file (SYS001).

Online maintenance for Application documents, such as documents marked for deletion, is also applied to the outbound application databank files when the Databank Reprocess Switch is active. You must then run the Outbound Application Maintenance (EDID101) program to physically delete a document from the databank file.

### Processing Option – Partner Read Sequence

When processing in the Partner/Qualifier mode and using the Partner Cross Reference file, you can improve efficiency by instructing the Mapper to read the Partner Cross Reference file first to lookup partner and user information. Default processing looks first to the partner profile and accesses the Partner Cross Reference file only if the partner is not found. Select this option if the majority of the partner profile accesses are made through the Partner Cross Reference file.

The feature is selected through the use of the Partner Read Switch in column 39 on the first input parameter record in the Mapping Parameter Input file (SYS001).

### Processing Option – User Exits

#### *Overview*

Gentran:Basic can handle exceptional situations that regular mapping cannot resolve. Examples of two such situations are accessing files outside of the system and accessing nonstandard data conversions.

**Caution:** By using user exits, you are circumventing all of the built-in edits of the system. Results that are returned by the exit are mapped into the data as instructed.

#### *Version 2 User Exit*

Version 2 has an expanded call block that passes more information to the user exit. The following user exits are available:

- Application Record Exit
- Outbound Envelope User Exit

To take advantage of the version 2 exits or the expanded call blocks, you must indicate to the Mapper, by using the EXIT-VERSION parameter on the first parameter record, that you want to

use version 2 of the user exits. If you have existing user exits and do not want to use these features, you can indicate that you want to process with version 1.

**Note:** Do not mix version 1 and version 2 user exits.

The version 2 user exit consists of a common call block that is used for all user exit types. These user exit types are:

- **Start Process Exit**  
This exit is used at the start of the mapping run after the Gentran:Basic files are opened. This exit, called the Begin Process User Exit Routine, is specified on the parameter record. This routine opens any needed files and performs any preliminary processing.
- **Application Record Exit**  
This exit is used just after an application record is read during Outbound Mapping and just before an application record is written during Inbound Mapping. This user exit, called the Application Record Exit, is similar to the current segment level user exit. This user exit provides you with exit points at all input and output points in the processing flow. The application record is passed to the user exit and can be modified as necessary.
- **Segment Exit**  
Segment level user exits give additional processing power to the Mapping Integration System. The segment level user exits are invoked after the specified segment is read (inbound), or before the specified segment is written (outbound). The segment is passed to the user exit, and can be modified as necessary.
- **Element Exit**  
By specifying a user exit routine at the element level on the Extended Element Mapping screen, the exit is called each time the mapping for that element is invoked. If conditions that have been set up on the mapping of the element are not satisfied, the exit is not invoked.
- **Outbound Envelope User Exit**  
As the interchange, group, and transaction envelopes are created in the Outbound Mapper, a call is made to this exit before the envelopes are written out. This provides the capability for the user program to change fields on the envelopes before they are written. Calls are made for each group of envelopes being processed (e.g., Call 1 is ISA, GS, and ST; then Call 2 is SE and ST; then Call 3 is SE, GE, and IEA).
- **End Process Exit**  
This exit is used at the end of the mapping run after Gentran:Basic files are closed. This exit, called the End Process User Exit Routine, is specified on the parameter record. This routine closes any open files and performs any end-of-run processing.

This common call block includes all of the functionality provided in version 5.02.

**Note:** The Mapper clears only the areas that are valid for that type of record. For example, if application record level processing is taking place, segment level processing fields are not cleared during execution



### *Version 1 User Exits (Prior to Gentran:Basic 5.5)*

The user exits prior to the release of Gentran:Basic 5.5 are version 1 user exits. Element and segment level user exits for version 1 are described in the following section.

#### **Element Level User Exits (version 1)**

User exits can be invoked in any of these times during the mapping process:

- At the start of the mapping run after the Gentran:Basic files are opened. This exit, called the Begin Process User Exit Routine, is specified on the parameter record. This routine opens any needed files and performs any preliminary processing.
- At the end of the mapping run after Gentran:Basic files are closed. This exit, called the End Process User Exit Routine, is specified on the parameter record. This routine closes any open files and performs any end-of-run processing.
- As each element is mapped. By specifying a user exit routine at the element level on the Extended Element Mapping screen, the exit is called each time the mapping for that element is invoked. If conditions that have been set up on the mapping of the element are not satisfied, the exit is not invoked.

All user exits at the element level use the element level call block record layout to communicate with the main module.

**Note:** All reserved constants are available to the exit; the reserved constants should be modified only if they are not used elsewhere in the mapping.

*See the User Exit Call Block Layout (Element Level – version 1) on page 3-42.*

#### **Segment Level User Exits (version 1)**

Segment level user exits give additional processing power to the Mapping Integration System. Segment level user exits are invoked after the specified segment is read (inbound), or before the specified segment is written (outbound). The segment is passed to the user exit, and can be modified as necessary.

*See the User Exit Call Block Layout (Segment Level – version 1) on page 3-43.*

#### **Processing Option – Map Blank Sub-Fields**

Mapping blank sub-fields allows sub-fields that contain spaces to be mapped and holds a place in the element if a subsequent sub-field for the element contains non-space data. Trailing spaces are compressed. This feature is selected through the Map Blank Sub-Fields switch in column 54 in the first Mapping Parameter record of the Mapping Parameters file (SYS001).

#### **Processing Option – Pass EDI IDs to Outbound Editor**

**Note:** This option is valid only if the value **P** (Partner/Qualifier) or **M** (Mixed) is specified as the trading partner mode, on the Configuration file. Do not specify this option unless necessary.

By default, the Outbound Mapper passes the resolved Partner ID that generates the envelopes to the Outbound Editor. The Outbound Editor uses these IDs (without cross-references) to determine proper error rejection and data separation requirements for the trading partner.

Prior to release 5.0, the Outbound Editor would determine data separation and error rejection based on the Receiver ID and Qualifier fields at the interchange and group level. As a result, some users established trading partner entries for these ID/Qualifier combinations and did not use cross-references. Data separation and error rejection were established under these partner entries as opposed to the partner entry used to generate the envelope.

Users who have implemented in this manner (with EDI IDs), should select the Pass EDI IDs option, to continue to process with no modifications to the partner profile.

The feature is selected through the Partner Read Switch in column 39 on the first input parameter record in the Mapping Parameter Input file (SYS001).

### Processing Option – EDI Data Print

Optionally, the first 125 characters of each Application record read and each EDI segment generated can be printed. This option is selected by entering **Y** in column 11 in the first Mapping Parameter record in the Mapping Parameters file (SYS001). The User/Partner IDs and names can also be printed when they are loaded. This option is set to **ON** by entering **1** in column 42 in the first Mapping Parameter record in the Mapping Parameter file (SYS001). These features can be extremely useful when first setting up and testing a map. These print options are selected with a Runtime parameter. Both are also available when using the instream parameter option.

*See Instream Parameter Record Layout – Outbound later in this section for further details.*

### Processing Option – Return Code Setting

The Mapper can be instructed to issue a return code at the end of the mapping process. To turn on this feature, enter **Y** in column 34 of the first Mapping Parameter record in the Mapping Parameters file (SYS001). This return code can be interrogated by the job stream and conditional processing can be performed based on the value of the return code.

Return codes are located on the error message and control file. These codes can be modified online to meet your installation standards, through the (error) Message Maintenance subsystem.

### Processing Option – \$TCR Transmission Control Record

This option enables you to generate user-defined transmission control information by defining the \$TCR User-Defined Record ID for your outbound partner and using the TCR Envelope parameter in the Enveloping Parameter file (SYS000). The user-defined transmission Control record information can be generated to accompany the outbound interchange.

*See Enveloping Parameter Records (SYS000) later in this section for additional details.*

### **Processing Option – Gentran:Plus**

If the Gentran:Plus communications system is implemented, the PLUS Runtime envelope parameter must be activated to generate the required \$\$ADD records that are needed to place data on the communications queues. This feature is selected by using the PLUS parameter record in the Enveloping Parameters input file (SYS000).

*See Enveloping Parameter Records (SYS000) later in this section for additional details.*

### **Processing Option – Truncated Sender's Reference (TRADACOMS) (Gentran:Plus only)**

The Sender's Reference (SNRF) for the TRADACOMS standard and the interchange control reference UNB05 for EDIFACT contain a 14-character default data element. Certain VANS require that only a six-character SNRF or control reference be used. This option is activated by the Runtime SNRF envelope parameter. This feature is selected by using the SNRF parameter record in the Enveloping Parameters input file (SYS000).

*See Enveloping Parameter Records (SYS000) later in this section for additional details.*

### **Processing Option – Gentran:Viewpoint**

The Outbound Mapping program performs all actions required for Gentran:Viewpoint Exception Management and Tracking Management within Gentran:Basic when the Configuration file indicates installation of Viewpoint.

For Viewpoint Tracking Management, the Outbound Mapper accepts Viewpoint User Tracking Records in the Application Data file. Viewpoint User Tracking Records are required if you want to extend the monitoring capabilities of Tracking Management system into application systems and processes.

The Instream Viewpoint User Tracking record should be generated by the user application tracking program as the last record for a document. The purpose of the Instream Viewpoint User Tracking record is to assign a unique Trace ID to every document for which Viewpoint Tracking Management is performed.

*See Instream Viewpoint User Tracking Record later in this section for the record layout.*

If Viewpoint Tracking Management is implemented and a document is found without an associated Instream Viewpoint Tracking record, the system assigns a Trace ID for each document.

The Outbound Mapper passes all Trace IDs to the Outbound Editor in the Mapped Application Data file.

## Files Used

### Input

The following files are used as input:

Filename	Description
SYS000	Envelope Parameters File This file contains the parameters to determine how envelope generation is to be performed. This file is used to generate user-defined transmission control record information (TCR parameter option). For Gentran:Plus customers only: This file is also used to input the Gentran:Plus parameter records.
SYS001	Parameters File This file contains the parameters that drive the Outbound Mapping process.
ERRCTL	Error Message and Control File This file contains descriptions of every error in the Gentran:Basic system. Error rejection is not supported by the mapping function.
SYS003	Application Data File This file contains the Application Data to be read. The requirements of this file are: The file can be either in variable block format or fixed block format. The longest record can be no longer than 5000 bytes. There must be a record type code from 1 to 10 bytes in length that uniquely identifies each type of record in the file. A record type must identify the start of the Header section, which may occur only one (1) time per document. If Detail records are used, a record type that identifies the start of the Detail Section. If Summary records are used, a record type must identify the start of the Summary Section. This record may occur only one (1) time per document. The data must contain the proper CNTL records before each group of Header records if the CNTL Pass-Thru option is used. The data must contain ###PARM records before each different application if you are using the ##INSTREAM option. Viewpoint users — The data must contain the ###VPTM record as the last record of a document for Viewpoint Tracking to occur.
SYS020	Application Data Definition-Contains the definition of the Application Data file.
SYS021	Application Record Definition Contains the definition of the records that are part of the Application Data file.
SYS022	Application Field Definition Contains the definition of the fields that are part of the Application Data file.

Filename	Description
SYS023	Application/Transaction Cross Reference Provides a cross-reference to all transaction definitions that refer to the application data definition.
SYS030	Code Translation File Used to translate ID type fields from an application value to a standard value.
SYS031	Code Validation File Used to validate application data values.
SYS032	Data Translation File Used to translate application data values to one or more trading partner's data values.
SYS040	Transaction Definition Contains the definition of the transaction to be mapped.
SYS041	Transaction Segment Definition Contains the definition of the segments to be mapped for this Transaction.
SYS042	Transaction Element Definition Contains the definition of the elements to be mapped for this transaction.
SYS090	Partner Profile This file used to load specific information concerning the sender and receiver. The information used is at the interchange, group, and transaction level.
SYS091	Outbound Partner Control File This file is used to generate the interchange, group, and transaction control numbers for the EDI envelopes.
SYS095	Partner Cross Reference File This file is used to cross-reference the partner identification given on the Control records or defined by the application to the partner profile. <b>Note:</b> This file is accessed only when processing in the Partner/Qualifier mode or Mixed mode.
EDIPREL	Partner Relationship File This file is used to cross-reference EDI IDs to User/Partner IDs. <b>Note:</b> This file is accessed only when processing in the Relationship mode.

*Output*

The following files are used as output:

<b>Filename</b>	<b>Description</b>
SYS004	Mapped Application Data in EDI Format This file contains the application data in EDI format with optional CNTL records to be used for envelope generation. This file is passed to the EBDI011M program for envelope generation or to the EBDI002 program for editing (if envelopes generated by Mapper).
SYS005	Error Report This file lists all of the errors encountered while mapping the application data to the standard.
SYS006	Summary Report This file contains summary information for the mapping run. Information includes processing parameters, control counts, and performance information.
SYS007	Summary Report in Data Format This file contains the summary report control counts in data format.
SYS098	Fixed Format Output File (For Gentran:Structure customers only.)
SYS099	Application Suspense File This file contains all application documents that were not mapped because a transaction map could not be selected.

*Databank*

The following are databank files:

<b>Filename</b>	<b>Description</b>
EDICFG	Configuration File This file contains records to control system-level features and options.
EDIOAA	Outbound Application Databank Directory This file contains status and control information for outbound application data and maintains records for controlling data recovery.
EDIOAS	Outbound Application Databank Message Store This file contains application data in the same format as it is received from the application and the data on the message store file can be viewed and optionally edited using CICS Databank Maintenance.
EDIOACA	Outbound Application Databank Change Audit This file contains detailed audit information of all online updates which are applied to the Outbound Application Databank.

Filename	Description
EDIQ093	Outbound Application Databank Transaction Queue This file contains transactions between the CICS and batch environments for the Outbound Application Databank created through Databank Maintenance.
EDIRW	Reprocess Work File These files are used by the Outbound Mapper and the Inbound Editor for processing Databank Maintenance Reset action requests. The files are temporary and are only required during the job step.
EDIAUD3	Databank Update Report

### *Gentran:Viewpoint Users Only*

The following are Gentran:Viewpoint files:

Filename	Description
EDIVT	The Trace ID file is used by Gentran:Viewpoint Tracking Management to store unique identifiers for tracked items.
EDIVP	The partner profile contains the rules that are used to control the Gentran:Basic system as defined in the Extended System Management and Exception Management.
EDIVSF	The Staging files are used to pass data from a Gentran:Viewpoint enabled system to Gentran:Basic. Staging files contain system messages issued in the Gentran:Viewpoint enabled system, as well as control/status information.

Outbound Mapping Process

Figure 3.3 illustrates the outbound mapping process.

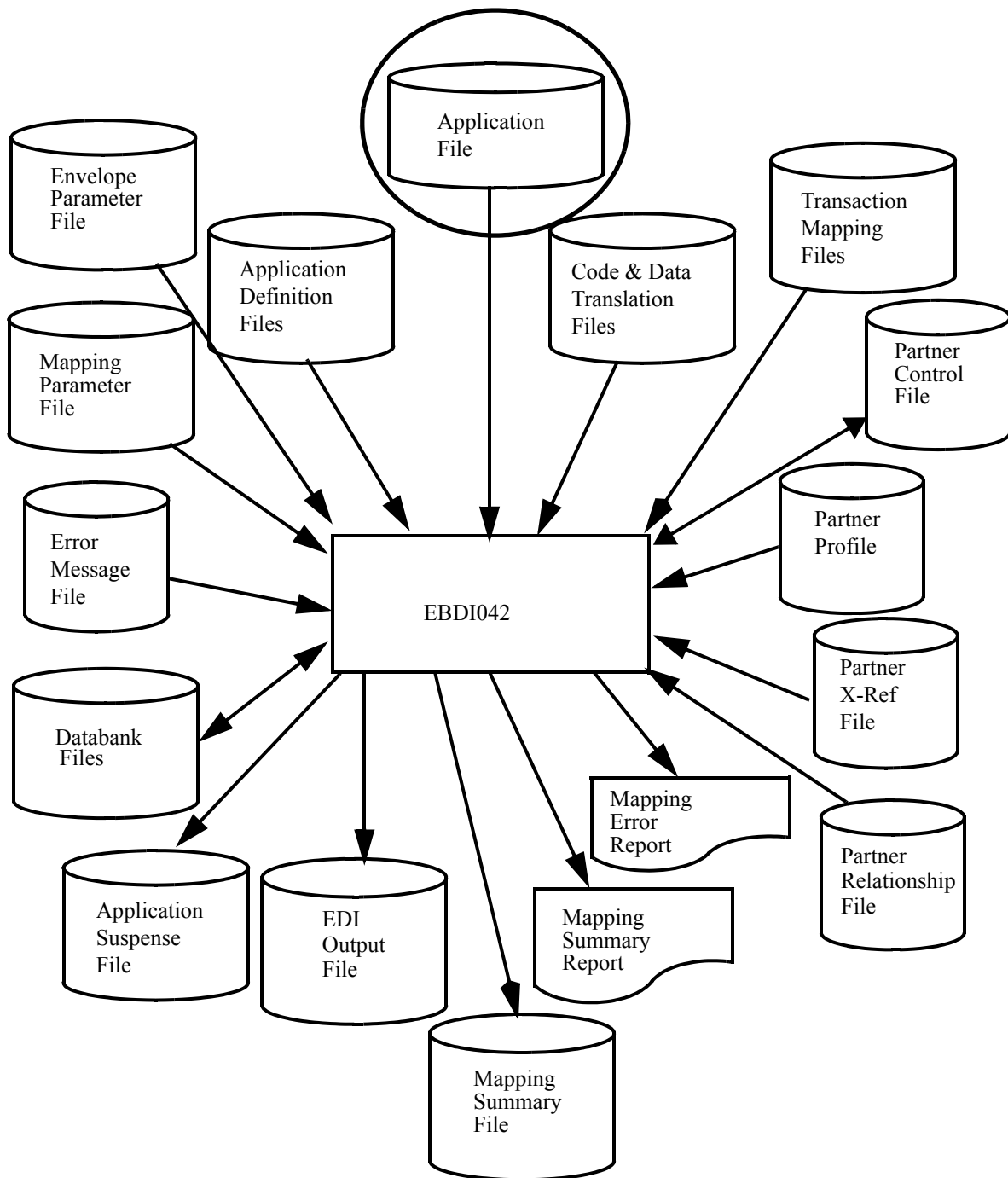


Figure 3.3 Program EBDI042 – Outbound Mapping



## Record Layouts

### Mapping Parameter Records (SYS001)

These record layouts are used by the Outbound Mapper to drive the outbound mapping process. The file is in a sequential format. The records are a fixed 80-byte length. Many of the options set in the parameter records have been described earlier in the processing section of this chapter.

### Mapping Parameter Record Layout #1

The following table describes Mapping Parameter Record Layout #1.

Positions	Field Content	COBOL Picture	Format	Length	Description
001 – 010	Application Data ID	X(10)	AN	10	The name assigned to the definition of the Application Data file, or ##INSTREAM to specify application parameters.
011 – 011	EDI Data Print Switch	X(01)	AN	01	<b>Y</b> – Instructs the Mapper to print the first 125 characters of each record read and each record generated. <b>Note:</b> When using ##INSTREAM, a “Y” in this field will override the value found in position 18 of the instream ###PARM record.
012 – 012	Forced Abend Switch	X(01)	AN	01	<b>Y</b> -Instructs the Mapper to abend if any fatal errors are encountered during execution.
013 – 019	Start Process User Exit	X(07)	AN	07	The name of a user-written program to be executed after all Gentran:Basic files have been opened.
020 – 026	End Process User Exit	X(07)	AN	07	The name of a user-written program to be executed after all Gentran:Basic files have been closed.
027 – 027	Application Decimal Notation	X(01)	AN	01	A comma (,) instructs the Mapper to interpret a comma as a decimal point in the Application file.
028 – 028	Application File Type	X(01)	AN	01	If instream application parameters are used <b>V</b> – Indicate the Application file is variable <b>F</b> – Indicate a fixed format.
029 – 032	Application File Length	9(04)	ZD	04	If instream application parameters are used, code the maximum record length of the Application file. For variable records, add 4 bytes.
033 – 033	Not used				Used for inbound only
034 – 034	Return Code Switch	X(01)	AN	01	If = <b>Y</b> , generate a return code.
035 – 035	Not used				Used for inbound only

Positions	Field Content	COBOL Picture	Format	Length	Description
036 – 036	Databank Configuration Switch	X(01)	AN	01	0 – No Databank 1 – Full Databank 2 – Directory Only Databank
037 – 037	Databank Processing Level Switch	X(01)	AN	01	0 – No Databank 1 – Full Databank 2 – Directory Only Databank. 3 – Partner Databank Level
038 – 038	Databank Reprocess Switch	X(01)	AN	01	N – No reprocessing Y – Reprocess data from application Databank.
039 – 039	Partner Read Switch	X(01)	AN	01	0 – Read partner profile first, then check cross-reference 1 – Read cross-reference first, then get partner 2 – Read partner profile first, but pass the EDI IDs to the Outbound Editor 3 – Read cross-reference first, but pass the EDI IDs to the to the Outbound Editor
040 – 040	Not used	X(01)	AN	01	Not used.
041 – 041	Not used				Internal use only
042 – 042	Partner Print Switch	X(01)	AN	01	0 – Do not print user/partner information when partner is loaded 1 – Print partner information when partner is loaded.
043 – 043	Filler	X(01)	AN	01	Used for inbound only
044 – 044	Document Tracking Switch	X(01)	AN	01	1 – Print an entry on the Audit Trail for each document in error in the run. 2 – Print an entry on the Audit Trail for each document processed in the run.
045 – 045	Support Quote Switch	X(01)	AN	01	Y – Execute additional program logic to allow a single quote in reserved word constants and literal constants.
046 – 046	Standard Type	X(01)	AN	01	Used for Gentran:Structure only. F – Fixed-format standard. V or blank – Variable-format standard (default).
047 – 051	Maximum Length	X(05)	AN	05	Used for Gentran:Structure only. The maximum length of a segment in the defined standard. 32760 is the maximum this value can be for fixed-length files. 32752 is the maximum this value can be for variable-length files.
052 – 052	Not used				Used for inbound only

Positions	Field Content	COBOL Picture	Format	Length	Description
053 – 053	Initialize Numerics	X(01)	AN	01	Used for Gentran:Structure only. <b>Y</b> – Allows non-packed dates and numeric data with types N0-N9 and S0-S9 to be set to 0 if no valid value is mapped to the element.
054 – 054	Map Blank Subfield	X(01)	AN	01	<b>Y</b> – Subfields with spaces as values hold their places in the element. <b>N</b> or <b>blank</b> – (default) Spaces are compressed.
055 – 055	User Exit Version	X(01)	AN	01	<b>1</b> or <b>blank</b> – Use version 1 of the User Exit Call Block <b>2</b> – Use version 2 of the User Exit Call Block
056 – 062	Outbound Envelope User Exit	X(07)	AN	07	The name of a user-written program to execute before writing the outbound envelopes.
063 – 063	Not used				
064 – 064	Blank Partner Error Switch	X(01)	AN	01	Used with application partner enveloping. <b>Y</b> or <b>1</b> – Issue an error message and suspend the application data when the application has a blank Partner ID. <b>blank</b> – (default value) If the application has a blank Partner ID, the partner is processed using the previous Partner ID.
065 – 076	Default Outbound Version	X(12)	AN	12	Default outbound version used for each application document to resolve functional groups and transactions with the partner profile.
077 – 079	Filler	X(02)	AN	02	Not used.
080 – 080	Application Reference Load	X(01)	AN	01	<b>1</b> – Use new logic to load the Application Reference field on STX. <b>0</b> or <b>blank</b> – Use old logic to load the Application Reference field on STX.

**Note:** Determine databanking performance by using the combination of the Databank Configuration switch and Databank Processing Level switch.

*Mapping Parameter Record Layout #2*

The following table describes Mapping Parameter Record Layout #2.

Positions	Field Content	COBOL Picture	Format	Length	Description
001 – 035	Default Partner ID (limited to 15 characters in the Relationship mode)	X(35)	AN	35	Partner ID used in processing all outbound data in this run. Application partner reference is ignored.
036 – 039	Default Partner ID Qualifier (not used in the Relationship mode)	X(04)	AN	04	Partner ID/Qualifier used in processing all outbound data in this run. Application partner reference is ignored.
040 – 074	Default User ID (limited to 15 characters in the Relationship mode)	X(35)	AN	35	User ID used in processing all outbound data in this run. Application User reference is ignored.
075 – 078	Default User ID Qualifier (not used in the Relationship mode)	X(04)	AN	04	User ID/Qualifier used in processing all outbound data in this run. Application User reference is ignored.
079 – 079	CNTL Record Pass Thru Switch	X(01)	AN	01	<b>Y</b> – Instructs the Mapper that the application contains CNTL records at the interchange, group, and transaction level. <b>I</b> – Instructs the Mapper that the application contains CNTL records at the interchange level.
080 – 080	Envelope Generation Switch	X(01)	AN	01	<b>Y</b> – Generate CNTL records. <b>N</b> or <b>blank</b> – Generate envelopes. <b>I</b> – Generate envelopes and always generate a new interchange for every document processed.

*Mapping Parameter Record Layout #3*

The following table describes Mapping Parameter Record Layout #3.

Positions	Field Content	COBOL Picture	Format	Length	Description
001 – 003	Multiple Envelope ID	X(03)	AN	3	Interchange Envelope ID used in key of control, group, and transaction records on the partner profile.
004 – 015	Default Interchange Version	X(12)	AN	12	Default interchange version used for each application document to resolve the control partner profile.
016 – 080	Filler	X(65)	AN	65	Not used.

**Mapping Parameter Record Layout #4**

The following table describes Mapping Parameter Record Layout #4.

Positions	Field Content	COBOL Picture	Format	Length	Description
001 – 001	Filler	X(01)	AN	1	Not used.
002 – 002	Spaces in UNB:07	X(01)	AN	12	Y – Always fill application reference (UNB:07) with spaces.
003 – 003	Structure Databank Level	X(01)	AN	1	0 – No databanking 1 – Full databanking 2 – Directory databanking only
004 – 004	Interchange Version Off	X(01)	AN	1	Y – Do not use interchange version for this run.
005 – 005	Group Version Off	X(01)	AN	1	Y – Do not use group version for this run.
006 – 006	Transaction Version Off	X(01)	AN	1	Y – Do not use transaction version for this run.
007 – 007	Multiple Envelope ID Off	X(01)	AN	1	Y – Do not use multiple envelope ID for this run.
008 – 008	Always Generate Group Envelope	X(01)	AN	1	Y – Always generate group envelope for NCPDP processing only.

**Enveloping Parameter Records (SYS000)**

These record layouts are used during the outbound envelope processing. They specify how control numbers are generated. The file is in a sequential format. The records are a fixed 80-byte length. These records have been explained in detail in the Enveloping section of this chapter.

**UCS NUMBER Record**

The following table describes the UCS NUMBER Record.

Positions	Field Content	COBOL Picture	Format	Length	Description
001 – 020	UCS Number	X(20)	AN	20	Indicates UCS number structure for transaction control number to be used. Valid value: USE UCS NUMBERS FOR.
021 – 032	Version	X(12)	AN	12	Version of the standards that causes UCS numbering of the transactions.
033 – 033	Filler	X(01)	AN	01	Not used.
034 – 036	Agency	X(03)	AN	03	Indicates the numbering structure for standards agency to be used.
037 – 080	Filler	X(44)	AN	44	Not used.

**6/3 NUMBER Record**

The following table describes the 6/3 NUMBER Record.

<b>Positions</b>	<b>Field Content</b>	<b>COBOL Picture</b>	<b>Format</b>	<b>Length</b>	<b>Description</b>
001 – 020	Alternate Record Number	X(20)	AN	20	This parameter generates 6/3 alternate numbering. Valid value: USE 6/3 NUMBERS FOR.
021 – 026	Transaction Set	X(06)	AN	06	Transaction set identifier of the standards used in alternate numbering sequencing.
027 – 027	Filler	X(01)	AN	01	Not used.
028 – 039	Version	X(12)	AN	12	Standards version to be used for alternate numbering.
040 – 040	Filler	X(01)	AN	01	Not used.
041 – 043	Agency	X(03)	AN	03	Standards agency to be used for alternate numbering.
044 – 080	Filler	X(37)	AN	37	Not used.

**5/3 NUMBER Record**

The following table describes the 5/3 NUMBER Record.

<b>Positions</b>	<b>Field Content</b>	<b>COBOL Picture</b>	<b>Format</b>	<b>Length</b>	<b>Description</b>
001 – 020	Alternate Record Number	X(20)	AN	20	This parameter generates 5/3 alternate numbering. Valid value: USE 5/3 NUMBERS FOR.
021 – 026	Transaction Set	X(06)	AN	06	Transaction set identifier of the standards used in alternate numbering sequencing.
027 – 027	Filler	X(01)	AN	01	Not used.
028 – 039	Version	X(12)	AN	12	Standards version to be used for alternate numbering.
040 – 040	Filler	X(01)	AN	01	Not used.
041 – 043	Agency	X(03)	AN	03	Standards agency to be used for alternate numbering.
044 – 080	Filler	X(37)	AN	37	Not used.

*5/4 NUMBER Record*

The following table describes the 5/4 NUMBER Record.

Positions	Field Content	COBOL Picture	Format	Length	Description
001 – 020	Alternate Record Number	X(20)	AN	20	This parameter generates 5/4 alternate numbering. Valid value: USE 5/4 NUMBERS FOR.
021 – 026	Transaction Set	X(06)	AN	06	Transaction set identifier of the standards used in alternate numbering sequencing.
027 – 027	Filler	X(01)	AN	01	Not used.
028 – 039	Version	X(12)	AN	12	Standards version to be used for alternate numbering.
040 – 040	Filler	X(01)	AN	01	Not used.
041 – 043	Agency	X(03)	AN	03	Standards agency to be used for alternate numbering.
044 – 080	Filler	X(37)	AN	37	Not used.

*PLUS Record*

The following table describes the PLUS Record.

Positions	Field Content	COBOL Picture	Format	Length	Description
001 – 004	Gentran:Plus Parameter	X(04)	AN	4	Instructs program to generate \$\$ADD cards for Gentran:Plus. Valid value: PLUS.

*TCR Record*

The following table describes the TCR Record.

Positions	Field Content	COBOL Picture	Format	Length	Description
001 – 004	TCR Parameter	X(04)	AN	04	Instructs the program to generate \$TCR user-defined information. Valid value: TCR.

*GS Time Record*

The following table describes the GS Time Record.

Positions	Field Content	COBOL Picture	Format	Length	Description
001 – 027	GS Time Parameter	X(27)	AN	27	Instructs the program to generate the GS envelope with the GS05 time field having a length equal to the maximum length allowed according to the Standard. Valid value: GENERATE MAX LENGTH GS TIME.

*SNRF Record*

The following table describes the SNRF Record.

Positions	Field Content	COBOL Picture	Format	Length	Description
001 – 004	Six Character Sender's Reference Parameter	X(04)	AN	4	Instructs the program to generate six character SNRF elements for STX interchanges (ANA TRADACOMS) or six character interchange control reference UNB05 for UNB interchanges (EDIFACT). Valid value: SNRF.

*UNB Application Reference*

The following table describes the UNB Application Reference parameter.

Positions	Field Content	COBOL Picture	Format	Length	Description
001 – 021	UNB Application Reference parameter			21	This parameter instructs the program to allow blanks in the UNB07 Application Reference field. Valid value: ALLOW SPACES IN UNB07
022 – 080	filler				not used



*Instream Parameter Record Layout*

Use the following parameter record layout before each different application in instream processing. When used, these records are embedded in the Application file input and must be at least 17 bytes.

Positions	Field Content	COBOL Picture	Format	Length	Description
001 – 007	Instream Parameter Identifier	X(07)	AN	07	Literal constant ####PARM.
008 – 017	Application Data ID	X(10)	AN	10	The name assigned to the definition of the Application Data file.
018 – 018	EDI Data Print Switch	X(01)	AN	01	<b>Y</b> – Instructs the Mapper to print the first 125 characters of each record read and each record generated. <b>Note:</b> A “Y” in position 11 of the first record in the SYS001 will override the EDI Data Print Switch of the instream ####PARM record.
019 – 025	Start Process User Exit	X(07)	AN	07	The name of a user- written program to be executed after all Gentran:Basic files have been opened.
026 – 032	End Process User Exit	X(07)	AN	07	The name of a user-written program to be executed after all Gentran:Basic files have been closed.
033 – 033	CNTL Record Pass Thru Switch	X(01)	AN	01	<b>Y</b> – Instructs the Mapper that the application contains CNTL records at the interchange, group, and transaction level. <b>I</b> – Instructs the Mapper that the application contains CNTL records at the interchange level.
034 – 034	Application Decimal Notation	X(01)	AN	01	A comma (,) instructs the Mapper to interpret a comma as a decimal point in the Application file.
035 – 035	Forced Abend Switch	X(01)	AN	01	<b>Y</b> – Instructs the Mapper to abend if any errors are encountered during execution.
036 – 070	Default Receiver Partner ID (limited to 15 characters in the Relationship mode)	X(35)	AN	35	Partner ID used in processing all outbound data in this run. Application partner reference is ignored.
071 – 074	Default Receiver Partner ID Qualifier (not used in the Relationship mode)	X(04)	AN	04	Partner ID/Qualifier used in processing all outbound data in this run. Application partner reference is ignored.

Positions	Field Content	COBOL Picture	Format	Length	Description
075 – 109	Default Sender User ID (limited to 15 characters in the Relationship mode)	X(35)	AN	35	User ID used in processing data. Application partner reference is ignored.
110 – 113	Default Sender User ID Qualifier (not used in Relationship mode)	X(04)	AN	04	User ID/Qualifier used in processing data. Application partner reference is ignored.
114 – 114	Real Right	X(01)	AN	01	<b>Y</b> – Indicates that real-type numeric fields for fixed format data is justified.
115 – 117	Multiple Envelope ID	X(03)	AN	03	Interchange Envelope ID to be used in the Header Option field on the Control, Group, and Transaction key for the partner lookup.
118 – 120	Filler	X(03)	AN	03	Not used.
121 – 132	Default Standards version	X(12)	AN	12	Standards version to be used in optional version field on partner Group and Transaction key for partner lookup.
133 – 144	Default Interchange Version	X(12)	AN	12	Default interchange version used for each application document to resolve the control partner profile.

*CNTL Record Layout – Partner/Qualifier Mode*

The following table describes the CNTL Record for the Partner/Qualifier mode.

Positions	Field Content	COBOL Picture	Format	Length	Description
1 – 4	Label	X(04)	AN	4	Constant CNTL.
5 – 39	Partner ID	X(35)	AN	35	Partner ID of partner receiving transactions. This field is required on at least the first CNTL record.
40 – 43	Partner Qualifier	X(04)	AN	4	Partner ID/Qualifier of partner receiving transactions. This field is required on at least the first CNTL record.
44 – 44	Record Type	X(01)	AN	1	Valid values: <b>I</b> – Interchange <b>G</b> – Group <b>T</b> – Transaction <b>O</b> – Object
45 – 58	Control Number	X(14)	AN	14	Control Number to be used on the Header/Trailer segments of the level indicated by the previous field. If this field is blank, the appropriate control numbers must be in the partner profile.
59 – 59	Test	X(01)	AN	1	Valid values: <b>T</b> – Test interchange <b>Blank</b> – Production interchange. Meaningful only on the I Interchange Control record.
60 – 65	Transaction Type	X(06)	AN	6	Identifies transaction type of data.
66 – 71	Functional Group	X(06)	AN	6	Identified functional group of following data.
72 – 73	CNTL Version	X(02)	AN	2	Identifies the version of the Control record. Constant <b>V4</b> .
74 – 97	Gentran:Plus Message	X(24)	AN	24	Text to be moved to \$\$ADD Batch ID field. Valid only at interchange level.
98 – 111	Application Reference	X(14)	AN	14	Text to be moved to Application Reference Data Element on UNB and STX interchanges.
112 – 146	Receiver ID	X(35)	AN	35	EDI-Receiver ID Not used.
147 – 150	Receiver ID Qualifier	X(04)	AN	04	EDI-Receiver ID Qualifier Not used.
151 – 154	Interchange Only Switch	X(04)	AN	04	Used in EBDI011A to control TA1 only acknowledgments
155 – 155	Transaction Release	X(1)	AN	1	Message release for TRADACOMS
156 – 160	Delimiters	X(5)	AN	5	Inbound delimiters used for acknowledgment generation
161 – 162	Rail Constant	X(2)	AN	2	FA or SW used for Rail interchanges

Positions	Field Content	COBOL Picture	Format	Length	Description
163 – 168	Rail Transaction	X(06)	AN	6	Transaction ID used for Rail interchanges
169 – 180	Control Version	X(12)	AN	12	Optional version to be used on partner profile Lookup of Group and Transaction records.
181 – 215	UNH Common Access Reference	X(35)	AN	35	Text to be moved to the UNH Common Access Reference (UNH03). Only valid at the transaction level.
216 – 218	Multiple Envelope ID	X(03)	AN	03	Interchange Envelope ID to be used in Header Option field on the Control, Group, and Transaction keys for partner lookup.
219 – 219	Repetition Separator	X(01)	AN	01	Inbound repetition separator used for acknowledgment generation.
220 – ??	Filler	X(??)	AN	??	The length of the Filler field varies depending on the file the Control record is in.

### CNTL Record Layout – Relationship Mode

The following table describes the CNTL Record for the Relationship mode.

Positions	Field Content	COBOL Picture	Format	Length	Description
1 – 4	Label	X(04)	AN	4	Constant CNTL.
5 – 19	User ID	X(15)	AN	15	User ID portion of User/Partner ID. This field is required on at least the first CNTL record.
20 – 34	Partner ID	X(15)	AN	15	Partner ID portion of User/Partner ID. This field is required on at least the first CNTL record.
35 – 43	Filler	X(06)	AN	6	Not used.
44 – 44	Record Type	X(01)	AN	1	Valid values: <b>I</b> – Interchange <b>G</b> – Group <b>T</b> – Transaction <b>O</b> – Object
45 – 58	Control Number	X(14)	AN	14	Control number to be used on the Header/Trailer segments of the level indicated by the previous field. If this field is blank, the appropriate control numbers must be in the partner profile.
59 – 59	Test	X(01)	AN	1	<b>T</b> – Test interchange <b>Blank</b> – Production interchange. Meaningful only on the I Interchange Control record.
60 – 65	Transaction Type	X(06)	AN	6	Identifies the transaction type of data.

Positions	Field Content	COBOL Picture	Format	Length	Description
66 – 71	Functional Group	X(06)	AN	6	Identified functional group of following data.
72 – 73	CNTL Version	X(02)	AN	2	Identifies the version of the Control record. Constant V4.
74 – 97	Gentran:Plus Message	X(24)	AN	24	Text to be moved to \$\$ADD Batch ID field. Valid only at interchange level.
98 – 111	Application Reference	X(14)	AN	14	Text to be moved to Application Reference Data Element on UNB and STX interchanges.
112 – 146	Receiver ID	X(35)	AN	35	EDI-Receiver ID Not used.
147 – 150	Receiver ID Qualifier	X(04)	AN	04	EDI-Receiver ID Qualifier Not used.
151 – 154	Interchange Only Switch	X(04)	AN	04	Used in EBDI011A to control TA1 only acknowledgments
155 – 155	Transaction Release	X(1)	AN	1	Message release for TRADACOMS
156 – 160	Delimiters	X(5)	AN	5	Inbound delimiters used for acknowledgment generation
161 – 162	Rail Constant	X(2)	AN	2	FA or SW used for Rail interchanges
163 – 168	Rail Transaction	X(06)	AN	6	Transaction ID used for Rail interchanges
169 – 180	Control Version	X(12)	AN	12	Optional version to be used on partner profile Lookup of group and transaction records.
181 – 215	UNH Common Access Reference	X(35)	AN	35	Text to be moved to the UNH Common Access Reference (UNH03). Only valid at the transaction level.
216 – 218	Multiple Envelope ID	X(03)	AN	03	Interchange Envelope ID to be used in Header Option field on the control, group, and transaction keys for partner lookup.
219 – 219	Repetition Separator	X(01)	AN	01	Inbound repetition separator used for acknowledgment generation.
220 – ??	Filler	X(01)	AN	01	The length of the Filler field varies, depending on the file the Control record is in.

*User Exit Call Blocks*

The following tables describe User Exit Call Blocks.

**User Exit Call Block Layout (Element Level – version 1)**

<b>Positions</b>	<b>Field Content</b>	<b>COBOL Picture</b>	<b>Format</b>	<b>Length</b>	<b>Description</b>
1 – 1	Processing Option	X(01)	AN	01	This field is set up by the calling program to indicate where in the process the module is called. S – Start of process P – During process at the element level E – End of process
2 – 81	User Defined Area	X(80)	AN	80	This field can be defined for any purpose. Its intent is to provide a method of communicating between different exits and subsequent invocations of the same exit in different documents. This field is not used by the mapping process.
82 – 161	Field To Be Mapped	X(80)	AN	80	On entry to the exit, this field contains the result of the mapping up to this point. The user exit can modify this field causing the data contained in this field to be mapped.
162 – 163	Length To Be Mapped	S9(04) COMP	BI	02	On entry to the exit, this field contains the length of the field to be mapped. If the exit modified the field to be mapped, it also should set this length field to correspond.
164 – 165	Standard Minimum Length	S9(04) COMP	BI	02	This field contains the minimum length of the element to be mapped as defined by the standard. It should not be modified by the user exit.
166 – 167	Standard Maximum Length	S9(04) COMP	BI	02	This field contains the maximum length of the element to be mapped as defined by the standard. It should not be modified by the user exit.
168 – 169	Standard Data Type	X(02)	AN	02	This field contains the data type of the element to be mapped as defined by the standard. It should not be modified by the user exit.
170 – 4569	Constants 01- 55 (55 occurrences)	X(80)	AN	4400	These fields contain the 55 reserved constants that can be referred to on the mapping screens. The user exit can use these fields to pass data within the same document.

Positions	Field Content	COBOL Picture	Format	Length	Description
4570 - 4769	Hash Totals 01-20 (20 occurrences)	S9(12) V9(6) COMP-3	PD	200	These fields contain the 20 reserved hash totals that can be referred to on the mapping screens. The user exit can use these fields to pass data within the same document.
4770 - 4970	Accumulators 01 thru 20 (20 occurrences)	S9(18) COMP-3	PD	200	These fields contain the 20 reserved accumulators that can be referred to on the mapping screens. The user exit can use these fields to pass data within the same document.

## User Exit Call Block Layout (Segment Level – version 1)

Positions	Field Content	COBOL Picture	Format	Length	Description
1 - 256	Segment	X(256)	AN	256	The segment read (inbound), or segment to be written (outbound).
257 - 4656	Constants 01- 55 (55 occurrences)	X(80)	AN	4400	These fields contain the 55 reserved constants that can be referred to on the mapping screens. The user exit can use these fields to pass data within the same document.
4657 - 4856	Hash Totals 01-20 (20 occurrences)	S9(12) V(6) COMP-3	PD	200	These fields contain the 20 reserved hash totals that can be referred to on the mapping screens. The user exit can use these fields to pass data within the same document.
4857 - 5056	Accumulators 01-20 (20 occurrences)	S9(18) COMP-3	PD	200	These fields contain the 20 reserved accumulators that can be referred to on the mapping screens. The user exit can use these fields to pass data within the same document.

## All User Exits (version 2)

Positions	Field Content	COBOL Picture	Format	Length	Description
1 - 80	USER-EXIT-USER- AREA	X(80)	AN	80	Used by the user exit module to communicate information to other user exit modules and subsequent invocations of the same module. It is never initialized or accessed by the mapping program, so any data that is placed in the field by any exit module remains in the field until it is removed or modified by the user.

Positions	Field Content	COBOL Picture	Format	Length	Description
81 – 380	USER-EXIT-FIELD	X(300)	AN	300	Valid only for element level exits. This field contains the data that is mapped to the target field or data element, and it can be modified by the exit module. The value in this field is mapped to the target field. If this field is modified, the next field, USER-EXIT-FIELD-LENGTH, should also be modified to reflect the length of the field to be mapped.
381 – 382	USER-EXIT-FIELD-LENGTH	S9(04)	BI	2	Valid only for element level exits. This field contains the length of the data that is mapped to the target field or data element. It should be modified by the exit module if the previous field, USER-EXIT-FIELD, is modified.
383 – 383	USER-EXIT-FIELD-FLAG	X(01)	AN	01	Must be equal to <b>Y</b> if any changes to have been made to the user exit field.
384 – 619	USER-EXIT-REFERENCE-AREA	Group		236	For reference purposes only. <b>Caution:</b> If you modify the data in these fields, unpredictable results can occur. If you must map the content of any of these fields, its value should be moved to one of the 99 available constants and mapped from there.
384 – 384	PROCOPT	X(01)	AN	01	Indicates where in the process the user exit is invoked. Valid values: <b>S</b> – Starts process exit. This exit is invoked at the beginning of the mapping process. <b>E</b> – Ends process exit. This exit is invoked at the end of the mapping process. <b>P</b> – Element level exit. This exit is invoked just before a target application field (inbound) or target data element (outbound) is loaded. <b>B</b> – Segment level exit. This exit is invoked just after a segment is read (inbound) or just before a segment is written (outbound). <b>R</b> – Application record level exit. This exit is invoked just after an application record is read (outbound) or just before an application record is written (inbound). <b>X</b> – Envelope generation level exit. This exit is invoked only during the outbound process, just before the envelopes are written to the EDI output file.
385 – 386	FIELD-MIN	S9(04)	BI	02	Valid only for element level exits. This field contains the minimum length of the intended target field of data element.



Positions	Field Content	COBOL Picture	Format	Length	Description
387 – 388	FIELD-MAX	S9(04)	BI	02	Valid only for element level exits. This field contains the maximum length of the intended target field of data element.
389 – 390	FIELD-TYPE	X(02)	AN	02	Valid only for element level exits. This field contains the data type of the intended target field of data element.
391 – 429	USER-INFO	Group		39	User information fields. Only after the entire Header portion of the input has been loaded and the trading relationship mode is user/partner, do these fields contain valid user information. The fields are valid during processing of the following: <ul style="list-style-type: none"> <li>• element level exit</li> <li>• outbound segment level exit</li> <li>• inbound application record level</li> <li>• inbound segment level outbound application record</li> </ul> <p><b>Caution:</b> When using these types of exits, carefully review the fields before updating to avoid inadvertent modifications.</p>
391 – 425	USER-ID	X(35)	AN	35	The resolved User ID used to translate the document. Values are valid only when the trading mode on the Configuration file is set to <b>M</b> for mixed. If the trading mode is set to <b>R</b> (Relationship), only the first 15 characters of the USER-ID is used. The USER-ID-QUAL field will be spaces.
426 – 429	USER-ID-QUAL	X(04)	AN	04	The resolved User ID/Qualifier used to translate the document. Values are valid only when the trading mode on the Configuration file is set to <b>M</b> for mixed. If the trading mode is set to <b>R</b> (Relationship), only the first 15 characters of the USER-ID is used. The USER-ID-QUAL field will be spaces.

Positions	Field Content	COBOL Picture	Format	Length	Description
430 – 468	PARTNER-INFO	Group		39	Partner information fields. These fields contain trading partner information only after the entire Header portion of the input has been loaded. The fields are valid only when processing the following: <ul style="list-style-type: none"> <li>• element level exit</li> <li>• outbound segment level exit</li> <li>• inbound application record level</li> <li>• inbound segment level outbound application record</li> </ul> <b>Caution:</b> When using these types of exits, carefully review the fields before updating to avoid inadvertent modifications.
430 – 464	PARTNER-ID	X(35)	AN	35	The resolved Partner ID used to translate the document.
465 – 468	PARTNER-ID-QUAL	X(04)	AN	04	The resolved Partner ID/Qualifier used to translate the document.
469 – 488	INTERCHANGE-REF	X(20)	AN	20	The value of the Interchange Reference (control number) field. This field is valid only during inbound processing. It always contains valid information during element level exit processing and inbound application record level processing, and contains valid information during inbound segment level processing only after the entire Header portion of the input has been loaded. <b>Caution:</b> When using these types of exits, carefully review the fields before updating to avoid inadvertent modifications.
489 – 508	GROUP-REF	X(20)	AN	20	The value of the Functional Group Reference (control number) field. This field is valid only during inbound processing. It always contains valid information during element level exit processing and inbound application record level processing, and contains valid information during inbound segment level processing only after the entire Header portion of the input has been loaded. <b>Caution:</b> When using these types of exits, carefully review the fields before updating to avoid inadvertent modifications.

Positions	Field Content	COBOL Picture	Format	Length	Description
509 – 528	TRANSACTION-REF	X(20)	AN	20	The value of the transaction (message) reference (Control Number) field. This field is valid only during inbound processing. It always contains valid information during element level exit processing and inbound application record level processing, and contains valid information during segment level processing only after the entire Header portion of the input has been loaded. <b>Caution:</b> When using these types of exits, carefully review the fields before updating to avoid inadvertent modifications.
529 – 548	USER-REF	X(20)	AN	20	The value of the User Reference field. This field is valid during inbound and outbound processing. It always contains valid information during element level exit processing, outbound segment level processing, and inbound application record level processing, and contains valid information during inbound segment level processing and outbound application record level processing only after the entire Header portion of the input has been loaded. <b>Caution:</b> When using these types of exits, carefully review the fields before updating to avoid inadvertent modifications.
549 – 572	PLUS-BATCH-ID	X(24)	AN	24	The value of the Gentran:Plus Batch ID field. This field is valid only during outbound processing. It contains valid data only if the batch ID is mapped to the ##ADD card. It always contains valid information during element level exit processing and outbound segment level processing, and contains valid information during outbound application record level processing only after the entire Header portion of the input has been loaded and the interchange envelopes have been generated. <b>Caution:</b> When using these types of exits, carefully review the fields before updating to avoid inadvertent modifications.

Positions	Field Content	COBOL Picture	Format	Length	Description
573 – 586	PLUS-BATCH-NUMBER	X(14)	AN	14	The value of the Gentran:Plus batch number. This field is valid only during inbound processing. It is passed to the Inbound Mapping program by the Inbound Editor for variable-format standards. It contains valid information during inbound segment level processing only after the entire Header portion of the input has been loaded. <b>Caution:</b> When using these types of exits, carefully review the fields before updating to avoid inadvertent modifications.
587 – 592	FUNCTIONAL-GROUP	X(06)	AN	06	The value of the functional group for the document to be translated. This field is valid during inbound and outbound processing. It always contains valid information during element level exit processing, outbound segment level processing, and inbound application record level processing, and contains valid information during inbound segment level processing and outbound application record level processing only after the entire Header portion of the input has been loaded. <b>Caution:</b> When using these types of exits, carefully review the fields before updating to avoid inadvertent modifications.
593 – 598	TRANSACTION-SET	X(06)	AN	06	The value of the transaction set identifier of the document to be translated. This field is valid during inbound and outbound processing. It always contains valid information during element level exit processing, outbound segment level processing, and inbound application record level processing, and contains valid information during inbound segment level processing and outbound application record level processing only after the entire Header portion of the input has been loaded. <b>Caution:</b> When using these types of exits, carefully review the fields before updating to avoid inadvertent modifications.

Positions	Field Content	COBOL Picture	Format	Length	Description
599 – 608	TRANSACTION-MAP	X(10)	AN	10	The name of the transaction map used to translate the document. This field is valid during inbound and outbound processing. It always contains valid information during element level exit processing, outbound segment level processing, and inbound application record level processing, and contains valid information during inbound segment level processing and outbound application record level processing only after the entire Header portion of the input has been loaded. <b>Caution:</b> When using these types of exits, carefully review the fields before updating to avoid inadvertent modifications.
609 – 618	APPLICATION-DEF	X(10)	AN	10	The name of the application definition used to translate the document. This field is valid during inbound and outbound processing. It always contains valid information during element level exit processing, outbound segment level processing, and inbound application record level processing, and contains valid information during inbound segment level processing and outbound application record level processing only after the entire Header portion of the input has been loaded. <b>Caution:</b> When using these types of exits, carefully review the fields before updating to avoid inadvertent modifications.
619 – 619	DIRECTION	X(01)	AN	01	Indicates the Mapper used to invoke the exit. Valid Values: <b>I</b> – Inbound Mapper <b>O</b> – Outbound Mapper
620 – 637	USER-EXIT-SOURCE-INFO	Group		18	Fields that describe the source of the mapping information. The information contained in these fields varies depending on the type of exit and the direction of the translation.

Positions	Field Content	COBOL Picture	Format	Length	Description
620 – 629	USER-EXIT-SRC-REC-ID	X(10)	AN	10	Valid values: Inbound segment level=Contains the Segment ID of the segment where the exit is invoked. Inbound record level=Not valid Outbound segment level=Not valid Outbound record level=Contains the record ID of the record where the exit is invoked. Inbound element level=Contains the segment ID of the segment that contains the element to be mapped from. Outbound element level=Contains the record ID of the record that contains the field to be mapped from.
630 – 633	USER-EXIT-SRC-REC-SEQ	S9(05)	BI	04	Valid values: Inbound segment level=Contains the segment sequence number of the segment where the exit is invoked. Inbound record level=Not valid Outbound segment level=Not valid Outbound record level=Contains the record sequence number of the record where the exit is invoked. Inbound element level=Contains the segment sequence number of the segment that contains the element to be mapped from. Outbound element level=Contains the record sequence of the record that contains the field to be mapped from.
634 – 637	USER-EXIT-SRC-FLD-SEQ	S9(05)	BI	04	Valid values: Inbound segment level=Not valid Inbound record level=Not valid Outbound segment level=Not valid Outbound record level=Not valid Inbound element level=Contains the element sequence number of the element to be mapped from. Outbound element level=Contains the field sequence of the field to be mapped from.
638 – 655	USER-EXIT-TARGET-INFO	Group		18	Fields that describe the target of the mapping information. The information contained in these fields varies depending on the type of exit and the direction of the translation.

Positions	Field Content	COBOL Picture	Format	Length	Description
638 – 647	USER-EXIT-TGT-REC-ID	X(10)	AN	10	Valid values: Outbound segment level=Contains the Segment ID of the segment where the exit is invoked. Outbound record level=Not valid Inbound segment level=Not valid Inbound record level=Contains the Record ID of the record where the exit is invoked. Outbound element level=Contains the Segment ID of the segment that contains the element in which to map. Inbound element level=Contains the Record ID of the record that contains the field in which to map.
648 – 651	USER-EXIT-TGT-REC-SEQ	S9(05)	BI	04	Valid values: Outbound segment level=Contains the segment sequence number of the segment where the exit is invoked. Outbound record level=Not valid Inbound segment level=Not valid Inbound record level=Contains the record sequence number of the record where the exit is invoked. Outbound element level=Contains the segment sequence number of the segment that contains the element in which to map. Inbound element level=Contains the record sequence of the record that contains the field in which to map.
652 – 655	USER-EXIT-TGT-FLD-SEQ	S9(05)	BI	04	Valid values: Outbound segment level=Not valid Outbound record level=Not valid Outbound segment level=Not valid Outbound record level=Not valid Outbound element level=Contains the element sequence number of the element in which to map. Inbound element level=Contains the field sequence of the field in which to map.
656 – 954	USER-EXIT-EXAMINER-AREA	Group		299	Not used.
656 – 656	XMR-PROCESS-TYPE	X(01)	AN	01	Not used.

Positions	Field Content	COBOL Picture	Format	Length	Description
657 – 668	XMR-MAPPING-STATUS-FIELDS	Group		12	Not used.
	XMR-MAPPING-STATUS (six occurrences)	X(02)	AN	02	
669 – 908	XMR-MAPPING-CONSTANT-FIELDS	Group		240	Not used.
	XMR-MAPPING-CONSTANT (12 occurrences)	X(20)	AN	20	
909 – 953	XMR-REFERENCE	X (45)	AN	45	Not used.
954 – 954	XMR-CHANGED-FLAG	X(01)	AN	01	Not used.
955 – 1258	USER-EXIT-ERROR-HANDLING	Group		304	Data that is returned to the mappers from an application user exit.
955 – 955	USER-EXIT-ERR-FLAG	X(01)	AN	01	Field set by the user to the value <b>Y</b> to indicate that an error occurred in the user exit.
956 – 957	USER-EXIT-MAP-STATUS	X(02)	AN	02	Reserved for future use
958 – 959	USER-EXIT-VAL-STATUS	X(02)	AN	02	Status that the Mapper terminates with at end of processing. Valid values: 04 – Warning message 16 – Fatal error message <b>Note:</b> Status 16 will cause the mapping process to terminate.
960 – 965	USER-EXIT-MSG-NUMBER	X(06)	AN	06	Message number that is displayed on the mapping report for the error that occurred in your user exit. This number begins with an alphanumeric character that can be distinguished from other Gentran:Basic errors.
966 – 1085	USER-EXIT-MSG-TXT1	X(120)	AN	120	Message that is displayed in the mapping report for the error that occurred in your user exit.
967 – 1205	USER-EXIT-MSG-TXT2	x(120)	AN	120	Reserved for future use
1206 – 1206	USER-EXIT-ACTION1	X(01)	AN	01	Reserved for future use
1207 – 1207	USER-EXIT-ACTION2	X(01)	AN	01	Reserved for future use



Positions	Field Content	COBOL Picture	Format	Length	Description
1208 – 1208	USER-EXIT-ACTION3	X(01)	AN	01	Reserved for future use
1209 – 1258	USER-EXIT-ERR-FILLER	X(50)	AN	50	Reserved for future use
1259 – 34028	USER-EXIT-RECORD-AREA	Group		32770	Two fields used for the segment level and application record level exits to contain the segment or record that the exit is associated with. Valid values: Inbound segment level=Contains the input segment information just after the segment is read. Inbound record level=Contains the output application record information just before the record is written. Outbound segment level=Contains the output segment information just before the segment is written. Outbound record level=Contains the input application record information just after the record is read.
1259 – 34025	RECORD	X(32767)	AN	32767	Contains the actual record or segment data. This field can be modified by the user. If this field is modified, the UE-RECORD-LENGTH field should be modified and the length of the record changed. The UE-RECORD-CHANGED-FLAG should be set to <b>Y</b> to instruct the mapping program that the record is changed.
34026 – 34027	RECORD-LENGTH	S9(04)	BI	02	The number of characters of valid data that are contained in the UE-RECORD field.
34028 – 34028	RECORD-CHANGED-FLAG	X(01)	AN	01	Set by the user to <b>Y</b> to indicate that a change is made to the UEW-RECORD data.
34029 – 56601	USER-EXIT-ENVELOPE-AREA	Group		22573	Valid data on return from an outbound envelope level exit.
34029 – 36578	VARIABLE-ENVELOPE (10 occurrences)	X(255)	AN	2550	If variable envelopes are generated, ten fields that contain the envelopes to be written. The exit can modify this area. If the number of envelopes to be written is changed, the new count should be set in the UE-ENVELOPE-COUNT variable. If there are any envelope changes the UE-ENVELOPE-CHANGED-FLAG should be set to <b>Y</b> .

Positions	Field Content	COBOL Picture	Format	Length	Description
36579 – 56578	FIXED-ENVELOPE (10 occurrences)	X(2000)	AN	20,000	If fixed-format (user-defined) envelopes are generated, ten fields that contain the envelopes to be written. The exit can modify this area. If the number of envelopes to be written is changed, the new count should be set in the UE-ENVELOPE-COUNT variable. If the length of any of the envelopes to be written is changed, the new length should be set in the appropriate occurrence of the UE-ENVELOPE-LENGTH variable. If there are any envelope changes, the UE-ENVELOPE-CHANGED-FLAG should be set to <b>Y</b> .
56579 – 56598	ENVELOPE-LENGTH (10 occurrences)	S9(04)	BI	20	Occurrence of this variable should be modified to reflect any length changes of fixed-format, user-defined envelopes that are written.
56599 – 56600	ENVELOPE-COUNT	S9(04)	BI	02	Determines how many envelopes are written.
56601 – 56601	ENVELOPE-CHANGED-RECORD	X(01)	AN	01	Set by the user to <b>Y</b> to indicate that a change is made to the UE-VARIABLE-ENVELOPE or UE-FIXED-ENVELOPE data.
56602 – 64521	CONSTANT-AREA	Group		7920	The 99 mapping constants that are available to the mapping system. These values can be tested or modified by the exit module. Modified constants are eligible for subsequent mapping.
56602 – 64521	CONSTANT (99 occurrences)	X(80)	AN	7920	The 99 mapping constants that are available to the mapping system. These values can be tested or modified by the exit module. Modified constants are eligible for subsequent mapping.
64522 – 64721	HASH-TOTAL-AREA	Group		200	The 20 mapping hash totals that are available to the mapping system. These values can be tested or modified by the exit module. Modified hash totals are eligible for subsequent mapping.
64522 – 64721	HASH-TOTAL (20 occurrences)	S9(18)	PD	10	The 20 mapping hash totals that are available to the mapping system. These values can be tested or modified by the exit module. Modified hash totals are eligible for subsequent mapping.
64722 – 64921	ACCUMULATOR-AREA	Group		200	The 20 mapping accumulators that are available to the mapping system. These values can be tested or modified by the exit module. Modified accumulators are eligible for subsequent mapping.

Positions	Field Content	COBOL Picture	Format	Length	Description
64722 – 64921	ACCUMULATOR	S9(18)	PD	10	The 20 mapping accumulators that are available to the mapping system. These values can be tested or modified by the exit module. Modified accumulators are eligible for subsequent mapping.
64922 – 64931	TOTAL-ITEMS	S9(18)	PD	10	The value of the reserved word constant TOTAL-ITEMS, which contains the value of the number of line items read (inbound) or written (outbound) at the current point in the process.
64932 – 64941	TOTAL-SEGMENTS	S9(18)	PD	10	The value of the reserved word constant TOTAL-SEGMENTS, which contains the value of the number of segments read (inbound) or written (outbound) at the current point in the process.

*Outbound Control Totals File (SYS007)*

This file contains total counts for the outbound mapping process in a data format and is in sequential format. The record generated is 300 bytes.

Position	Field Content	COBOL Picture	Format	Length	Description
001 – 008	Summary Start Date	9(08)	YYYYMMDD	8	This field defines the Summary Start Date.
009 – 016	Summary Start Time	9(08)	HHMMSShh	8	This field defines the Summary Start Time.
017 – 024	Summary End Date	9(08)	YYYYMMDD	8	This field defines the Summary End Date.
025 – 032	Summary End Time	9(08)	HHMMSShh	8	This field defines the Summary End Time.
033 – 047	Summary Documents Input	9(15)	N	15	This field defines the number of documents in the input file.
048 – 062	Summary Records Input	9(15)	N	15	This field defines the number of records in the input file.
063 – 077	Summary Characters Input	9(15)	N	15	This field defines the number of characters in the input file.
078 – 092	Summary Databank Documents Output	9(15)	N	15	This field defines the number of documents written to the Databank Directory file.
093 – 107	Summary Databank Records Output	9(15)	N	15	This field defines the number of records written to the Databank Message Store file.
108 – 122	Summary Documents Reprocessed	9(15)	N	15	This field defines the number of documents reprocessed.
123 – 137	Summary Records Reprocessed	9(15)	N	15	This field defines the number of records reprocessed.
138 – 152	Summary Characters Reprocessed	9(15)	N	15	This field defines the number of characters reprocessed.
153 – 167	Summary Documents Suspended	9(15)	N	15	This field defines the number of documents in the suspense file.
168 – 182	Summary Records Suspended	9(15)	N	15	This field defines the number of records in the suspense file.
183 – 197	Summary Characters Suspended	9(15)	N	15	This field defines the number of characters in the suspense file.

Position	Field Content	COBOL Picture	Format	Length	Description
198 – 212	Summary EDI Documents Out	9(15)	N	15	This field defines the number of EDI documents in the output file.
213 – 227	Summary EDI Segments Out	9(15)	N	15	This field defines the number of EDI segments in the output file.
228 – 242	Summary EDI Characters Out	9(15)	N	15	This field defines the number of EDI characters in the output file.
243 – 257	Summary Applications Processed	9(15)	N	15	This field defines the number of application definitions processed.
258 – 272	Summary Maps Processed	9(15)	N	15	This field defines the number of Maps processed.
273 – 287	Summary Partners Processed	9(15)	N	15	This field defines the number of partners processed.
288 – 300	Summary Packages Processed	9(13)	N	13	This field define the number packages processed.

**Note:** A COBOL copybook has been provided in the source utility product library. The member is SUMMPWS.

The Instream Viewpoint User Tracking record should be generated by the user application tracking program as the last record for a document. The Instream Viewpoint User Tracking record assigns a unique Trace ID to every document for which Viewpoint Tracking Management is performed.

**Note:** Records are inserted in the application input file (SYS003).

Positions	Field Content	COBOL Picture	Format	Length	Description
001 – 007	Viewpoint Tracking Literal	X(07)	AN	07	Literal Constant ###VPTM.
008 – 008	Filler	X(01)	AN	01	Not used.
009 – 018	Viewpoint Tracking Trace ID	X(10)	AN	10	A unique ID value that identifies the item in the Viewpoint Tracking Management System.

### Outbound Mapper Audit Trail Report

The Outbound Mapper Audit Trail report is generated when the Outbound Mapper program (EBDI042) is executed. This report indicates errors encountered, processing options, summary control counts, and a performance summary for mapping outgoing data.

Figure 3.4 through Figure 3.6 illustrate the Outbound Mapper Audit Trail report (Errors Encountered Mapping Outgoing Data), the Outbound Mapper Audit Trail report (Processing Options for Mapping Outgoing Data), the Outbound Mapper Audit Trail report (Summary Control Counts for Mapping Outgoing Data).

```

EBDI042  RUN 12/01/2005 TIME 16:07  ERRORS ENCOUNTERED MAPPING OUTGOING DATA  PAGE 1
ERROR  **RECORD**  FIELD SEG ELE  ERROR MESSAGE
NUMBER  NBR ID  SEQ # ID  SEQ INFORMATION  ERROR MESSAGE
RECEIVER: VENDOR-1-TUTORIAL - XYZ COMPUTER COMPANY
RECEIVER: VENDOR-1
QUAL:  APPL REF: PONUM BER-001  DOCUMENT #: 2
689  80  DATABANK KEY: VENDOR-1  /POFILE  /PONUMBER-001
RECEIVER: VENDOR-2-TUTORIAL - BULK PAPER COMPANY
RECEIVER: VENDOR-2
QUAL:  APPL REF: PONUM BER-002  DOCUMENT #: 3
689  80  DATABANK KEY: VENDOR-2  /POFILE  /PONUMBER-002
RECEIVER: VENDOR-3-TUTORIAL - TWO WAY COMMUNICATIONS
RECEIVER: VENDOR-3
QUAL:  APPL REF: PONUM BER-003  DOCUMENT #: 4
689  80  DATABANK KEY: VENDOR-3  /POFILE  /PONUMBER-003

PROCESSING ENDED WITH ERRORS - PROCESSING COUNTS BELOW
APPLICATION RECORDS READ ----- 68
APPLICATION RECORDS SUSPENDED - 0
TOTAL RECORDS WRITTEN ----- 170
RETURN CODE FOR MAPPING ----- 4
    
```

**Figure 3.4 Outbound Mapper Audit Trail Report (Errors Encountered Mapping Outgoing Data)**

```
EBDI042      RUN  12/01/2005 TIME 16:07      PROCESSING OPTIONS FOR MAPPING OUTGOING DATA      PAGE      1

APPLICATION TO PROCESS-----POFILE
USER EXIT VERSION SUPPORTED-----1
DATABANK PROCESSING CONFIGURATION-----DIRECTORY AND MESSAGE STORE
DATABANK PROCESSING LEVEL-----DIRECTORY AND MESSAGE STORE
PARTNER PROCESSING SEQUENCE-----SEARCH PARTNER PROFILE
DIRECTORY POSTING OPTION-----POST RECEIVER ONLY
PARTNER PRINT OPTION-----PRINT PARTNER INFO WHEN LOADED
DEFAULT SENDER PARTNER ID-----
DEFAULT SENDER PARTNER ID QUALIFIER-----0002
USE GROUP PARTNER WITH VERSION-----Y
USE TRANSACTION PARTNER WITH VERSION-----Y
CNTL RECORD GENERATION OPTIONS-----1
ENVELOPE GENERATION OPTION-----MAPPER GENERATES ENVELOPES
```

**Figure 3.5 Outbound Mapper Audit Trail Report  
(Processing Options for Mapping Outgoing Data)**

```

EBDI042  RUN  12/01/2005      TIME 12:00 SUMMARY CONTROL COUNTS MAPPING OUTGOING DATA      PAGE      1
PROCESSING BEGAN ON 12/01/2005 AT 12:00 PM.
SEQUENTIAL INPUT DOCUMENTS READ ----- 4
SEQUENTIAL INPUT RECORDS READ ----- 68
SEQUENTIAL INPUT CHARACTERS READ ----- 17,000
DOCUMENTS STORED ON DATA BANK ----- 4
RECORDS STORED ON DATA BANK ----- 68
DOCUMENTS REPROCESSED ----- 0
RECORDS REPROCESSED ----- 0
CHARACTERS REPROCESSED ----- 0
DOCUMENTS SUSPENDED ----- 0
RECORDS SUSPENDED ----- 0
CHARACTERS SUSPENDED ----- 0
EDI DOCUMENTS GENERATED ----- 4
TOTAL RECORDS WRITTEN ----- 170
NUMBER OF APPLICATIONS PROCESSED ----- 1
NUMBER OF MAP DEFINITIONS PROCESSED ----- 1
NUMBER OF TRADING PARTNERS PROCESSED ----- 4
PROCESSING ENDED ON 12/01/2005 AT 12:00 PM.
    
```

**Figure 3.6 Outbound Mapper Audit Trail Report  
(Summary Control Counts Mapping Outgoing Data)**



## *Field Definitions*

### **Errors Encountered Mapping Outgoing Data**

Error Numbr:

This field displays the error code number associated with the error message.

Record Nbr:

This field displays the application record number.

Record ID:

This field displays the application record identifier.

Field Seq #:

This field displays the field sequence number.

Seg ID:

This field displays the EDI segment identifier.

Ele Seq:

This field displays the EDI element sequence number.

Information:

This field displays descriptive information identifying additional information about the error message.

Error Message:

This field displays descriptive information identifying the error message.

(Message):

This field displays a message regarding the error or the statement “no errors occurred during processing.”

Application Records Read:

This field displays the total number of application records read.

Application Records Suspended:

This field displays the total number of application records suspended.

Total Records Written:

This field displays the total number of records written.

Return Code for Mapping:

This field displays the return code for mapping.

### **Processing Options for Mapping Outgoing Data**

This section of the report displays all the parameter options in effect for mapping outgoing data. Only those parameter options used are displayed.

### **Summary Control Counts Mapping Outgoing Data**

(Message):

This field displays a text message indicating the date (in MM/DD/YYYY format) and the time (in HH:MM format) that processing began.

**Sequential Input Documents Read:**

This field displays the total number of sequential input documents read.

**Sequential Input Records Read:**

This field displays the total number of sequential input records read.

**Documents Stored on Data Bank:**

This field displays the total number of documents stored on the data bank.

**Records Stored on Data Bank:**

This field displays the total number of records stored on the data bank.

**Documents Reprocessed:**

This field displays the total number of documents reprocessed.

**Records Reprocessed:**

This field displays the total number of records reprocessed.

**Characters Reprocessed:**

This field displays the total number of characters reprocessed.

**Documents Suspended:**

This field displays the total number of documents suspended.

**Records Suspended:**

This field displays the total number of records suspended.

**Characters Suspended:**

This field displays the total number of characters suspended.

**EDI Documents Generated:**

This field displays the total number of EDI documents generated.

**Total Records Written:**

This field displays the total number of records generated.

**Number of Applications Processed:**

This field displays the total number of applications processed.

**Number of Map Definitions Processed:**

This field displays the total number of map definitions processed.

**Number of Trading Partners Processed:**

This field displays the total number of trading partners processed.

## Outbound Envelope Generation

## EBDI011M, EBDI011A

The outbound envelope generation programs generate EDI interchange, group, and transaction envelopes for files that contain special format Control (CNTL) records. There are actually two envelope generation programs. They each perform the same function, the only difference being the source of the input data.

**EBDI011M Program–** This program generates envelopes for data that is produced by the Outbound Mapping (EBDI042) program. The generation of envelopes should be performed by the Outbound Mapping program in all but the most exceptional circumstances.

**EBDI011A Program–** This program generates the envelopes for outbound acknowledgment data that is produced by the Inbound Editor (EBDI001) program. The resulting output is further processed by the Outbound Editor (EBDI002) program. Use EBDI011A if the global parameter USE CNTL FOR ACKNOWLEDGEMENTS is used.

### Envelope Generation

Envelope generation is controlled by the placement of special format Control records in the input data. A Control record exists that corresponds to each level of EDI enveloping. The Control record level is determined by a one character code, I (interchange), G (group), T (transaction), and O (object).

The envelopes are generated with the information provided on the interchange control, group, and transaction partner records for the trading partner. Optionally, some of this data can be generated on the Control records and overrides the partner defaults.

Control numbers at all three levels can be provided on the Control records. These numbers are moved directly to the envelopes. In Package structures, the Object Header does not contain a control number. Therefore, a control number on a Control record cannot be moved to a UNO segment.

Control numbers can be automatically determined by the partner records at the appropriate levels. Control numbers are automatically incremented by one at the interchange, group and transaction level.

The separators and terminators from the inbound document are passed on the CNTL record from the Inbound Editor (EBDI001) program to the Acknowledgment Envelope Generation (EBDI011A) program. If the delimiters stored on the partner profile do not match the delimiters received on the inbound document, the delimiters on the CNTL record override the partner profile delimiters. This ensures that the delimiters used in the outgoing acknowledgment are consistent between the envelopes and data segments.

### Selectable Feature – TDDC Numbering

TDCC Standards (i.e., UCS, WINS, AIR, MOTOR and RAIL) allow for special numbering of the transaction segments. These control numbers are a combination of the low-order characters in the group control number and an incrementing number for each transaction within the group.

**Use UCS Numbers**

The USE UCS NUMBERS Runtime parameter generates special numbers for every document with a specified EDI version. The following shows how transaction control numbers are generated for different version types:

- Rail, Motor, Air - A nine-digit number is generated with the first five digits of the number resulting from the last five digits of group control number. The last four digits of the number are incremented by one within the group.
- UCS, Wins - An eight-digit number is generated with the first five digits of the number resulting from the last five digits of group control number. The last three digits of the number are incremented by one within the group.
- All other versions - A nine-digit number is generated with the first six digits of the number resulting from the last six digits of group control number. The last three digits of the number are incremented by one within the group.

Because of the consolidation of the TDCC Standard within ANSI, the following special numbering parameters are provided at the transaction set level for all EDI versions processed or for specific EDI versions. For example, special numbering may be required for an 875 document for ANSI version 003050, but not for an 875 document for ANSI version 003060.

**Use 5/4 Numbers**

The USE 5/4 NUMBERS Runtime parameter generates special numbers for every document that has the specified transaction set identifier for all EDI versions processed or specific EDI versions as follows:

A nine-digit number is generated with the first five digits of the number coming from the last five digits of group control number. The last four digits of the number are incremented by one within the group.

**Use 5/3 Numbers**

The USE 5/3 NUMBERS causes special numbers to be generated for every document that has the specified transaction set identifier for all EDI versions processed or specific EDI versions as follows:

An eight-digit number is generated with the first five digits of the number coming from the last five digits of group control number. The last three digits of the number are incremented by one within the group.

**Use 6/3 Numbers**

The USE 6/3 NUMBERS Runtime parameter generates special numbers for every document that has the specified transaction set identifier for all EDI versions processed or specific EDI versions as follows:

A nine-digit number is generated with the first six digits of the number coming from the last six digits of group control number. The last three digits of the number are incremented by one within the group.

*For record formats of these Runtime parameters, see the Record Layout topic under Envelope Processing Parameters (SYS000).*

### *For “RAIL” Documents Only*

#### **EBDI011A – for Acknowledgments**

The CNTL record passed from the Inbound Editor (EBDI001) program contains a value in the RAIL-AUTH field for Rail ISA interchanges. If a value other than spaces is in this field, EBDI011A overrides the ISA01 and ISA02 from the partner profile when generating the ISA interchange envelope for the acknowledgment. The ISA01 is set to 04 and ISA02 is set to the RAIL-AUTH field from the CNTL record. The RAIL-AUTH field contains FAxxx, where xxx is the transaction set ID of the document being acknowledged (e.g., 810, 850). If the RAIL-AUTH field contains spaces, the acknowledgment ISA envelope is generated using the partner profile.

Rail documents also bypass the envelope consolidation process for acknowledgments. The default processing is to consolidate acknowledgments going to the same trading partner under one ISA interchange. For Rail documents, the consolidation process may cause problems by combining acknowledgments from different transactions under one common ISA interchange. If a value is found in the RAIL-AUTH field on the CNTL record, envelopes for each CNTL record received are generated.

#### **EBDI011M – for Mapping**

This Outbound Envelope Generation program inputs the documents containing CNTL records from the Mapper. A value other than spaces in the RAIL-AUTH field on the CNTL record triggers the ISA01 set to 04 and the ISA02 set to the CNTL RAIL-AUTH field. The RAIL-AUTH field contains SWxxx, where xxx is the Transaction Set ID of the document type being processed (e.g., 426, 810, 850).

#### ***GS Envelopes – Generating the GS05***

It is possible to generate the time element in the GS segment (i.e., GS05) with varying lengths. The default is to generate GS envelopes with a four-digit GS05 time for all versions. Using the parameter GENERATE MAX LENGTH GS TIME in the SYS000 parameter record, the GS envelope record will be generated with the GS05 time field at the maximum length allowed according to the standard. If this parameter is used, X12 versions 003020 through 003030 will have a six-digit GS time field and version after 003030 will have an eight-digit GS time field. Versions prior to 003020 will have a four-digit GS time field.

#### ***UNB Envelopes – Generating the UNB07***

The Application Reference field in the UNB segment (i.e., UNB07) is populated with the value in the UNB07 field from the partner profile or message ID of the transaction if the value in the UNB07 field from the Control Information screen (EDIM019) is spaces. Using the parameter ALLOW SPACES IN UNB07 in the SYS000 parameter record, the UNB envelope record will be generated with spaces in the UNB07 field instead of the message ID if the partner profile contains spaces in the UNB07 field on the Control Information screen (EDIM019).

#### **Envelope Processing**

The envelope generation is controlled through the use of CNTL records. The Envelope Generation program uses the CNTL records included in the input data stream and the Gentran:Basic partner profile to determine what Header and Trailers to build.

The three types of input CNTL records are:

- **I-Type CNTL Record**

The I-Type CNTL record indicates the beginning of an interchange Header. The I-Type record includes information specifying the Partner ID (Partner/Qualifier mode) or the User/Partner ID (Relationship mode), as well as a (optional) test/production indicator, (optional) interchange control number, interchange Header Option, and (optional) group version.

When processing the I-Type record, the system uses the partner information on the Control record to read the Gentran:Basic partner profile. The information is then used on the partner profile Control record to determine what type of interchange to build (i.e., ISA, ICS, BG, GS, SCH, UNA, UNB, or STX). Based on how this partner is defined, the I-Type record supplies the interchange Header Option needed to read the partner profile Control record so that the appropriate fields can be moved from the partner profile to the interchange Header record. If the user has coded a Control number on the I-Type record, it is loaded into the control number field of the interchange record. When not coded, the interchange number is obtained from the Control Number field for the partner on the outbound Control file. Each control number on interchange Control segments (i.e., ISA, ICS, BG, GS, SCH, UNA, UNB, or STX) for this partner is one higher than the previous interchange Control number used for this partner.

In technical terms, Gentran:Basic reads the Control record for this partner, takes the interchange control number, adds one to it, places it in the Control Number field of the interchange control segment, and then updates the outbound Control file record. Finally, when a new I-Type CNTL record is found in the input data or the end of input is reached, the system builds the corresponding interchange Trailer (i.e., IEA, ICE, EG, UNE, EG, or END) for this interchange. When processing the input data, Gentran keeps track of the number of functional groups and fills this information in on the interchange Trailer.

- **G-Type CNTL Record**

The second type of record found in the input data is the G-Type CNTL record. The G-Type record indicates the beginning of a functional group (e.g., Purchase Order, Invoice). The G record contains the Functional Group ID, (optional) group control number, and the (optional) Partner ID (Partner/Qualifier mode) or the User/Partner ID (Relationship mode).

When Gentran finds a G-Type CNTL record, it reads the partner profile using either the Partner ID or the User/Partner ID found on the G record, or when not coded, the Partner ID or the User/Partner ID from the previous I record. The partner profile is read using this Partner ID or the User/Partner ID, specified Functional Group ID, and interchange Header Option. If this functional group record exists, information is moved from the partner profile record to the group Header record. If the specified functional group does not exist, the system reads the partner profile to obtain the !!!DFT group record, and build the group Header from the information located on this record.

The information that the program loads onto the group record depends upon the type of group record (i.e., GS, UNG, or BAT). The group Control Number field is moved based on a hierarchy. The system uses the first control number it finds, first if coded on the G-Type CNTL record, second the control number on the functional group record of the partner profile, or finally, the control number from the !!!DFT group record. By specifying the control number on a group record, the customer can have each functional group segment for a specific Functional ID be in sequence (incremented by one each time this type of functional group is sent). By specifying the control number on !!!DFT group record, the customer can have all functional groups in sequence regardless of type. If no control number is found, the system indicates the error and continues processing the input data. Finally, when a new G-Type CNTL record is found in the input data or the end of input is reached, the system builds the corresponding group Trailer for this functional group. When processing the input data, the system keeps track of the number of transaction sets and fills this information in on the group Trailer.

The BAT group record is only assigned to the !!!DFT Group ID of the partner. For every G-Type control, the BAT control number is incremented.

For UNG or BAT processing, either segment is optional. Control of the creation of the UNG or BAT is performed on the partner Group screen. If the group control number is not coded (contains spaces), no UNG or BAT segment is created. If the group control number is coded, the UNG or BAT segment is created.

- **T-Type CNTL Record**

The T-Type of CNTL record indicates the beginning of a transaction set (e.g., 880, 875). The T record contains the Transaction Set ID, (optional) Functional Group ID, the transaction set control number, and (optional) Partner ID (Partner/Qualifier mode) or the User/Partner ID (Relationship mode). When the system finds a T-Type record, it reads the partner profile using either the Partner ID or the User/Partner ID found on the T record, or when not coded, the Partner ID or the User/Partner ID from the previous I record.

The partner profile is read using this Partner ID or the User/Partner ID, specified Transaction Set, and interchange Header Option. If this transaction set record exists, information is moved from the partner profile record to the transaction Header record. If the specified transaction set does not exist, the system reads the partner profile to obtain the functional group record. If no group record exists, the system reads the !!!DFT transaction set record and builds the transaction set Header from the information located on this record.

The transaction set Control Number field is moved based on a hierarchy. The system uses the first control number it finds, first if coded on the T-Type record. Second, if coded on the control number on the transaction set record. Third, if coded on the transaction control number on the functional group record of the partner profile. Fourth and finally, if coded on the control number from the !!!DFT transaction set record.

Specifying the transaction control number on the transaction record designates that each transaction segment for a specific Transaction ID will be in sequential order (incremented by one each time this type of transaction is sent). Specifying

the transaction control number on a group record designates that each transaction segment within this type of functional group will be in sequential order. Specifying the transaction control number on the default transaction record for this partner designates that all transactions will be numbered in sequence regardless of type. If no control number is found, the system indicates the error and continues processing the input data.

**Note:** MHD transaction sets (ANA TRADACOMS) are always numbered from one by one within an interchange or group.

Finally, when a new T-Type record is found in the input data or the end of input is reached, the system builds the corresponding transaction set Trailer for this transaction set.

- **O-Type CNTL Record**

The O-Type of CNTL record indicates the beginning of an EDIFACT Package Structure (e.g., UNO/UNP). The O record contains a (optional) Partner ID (Partner/Qualifier mode) or the User/Partner ID (Relationship mode), a Functional Group ID of UNO, and a Transaction Set ID of UNO. For O-Type CNTL records, the partner profile is never read.

When the system finds a O-Type record, it removes it from the data. Following the O-Type record is the Object Header (UNO) segment. In the Mapper, the UNO and UNP segments were supplied via the application data. The UNO was defined as the first segment of a transaction set and the UNP was treated as the last segment of a transaction set. The actual object data was treated as the detail or middle segments within a transaction set. Therefore, when envelopes are generated, the CNTL O-Type record is dropped and the UNO segment becomes the Object Header record.



## Files Used

The following files are used by the EBDI011M and EBDI011A programs:

Filename	Description
SYS000	Envelope Parameters File This file contains input parameters to the envelope generation process. <b>Note:</b> See Envelope Processing Parameters – SYS000 on page 3-70 for specific parameter information and layout.
SYS001	Partner Processing Parameters file This file contains input parameters that instruct the program how to access the partner profile. <b>Note:</b> See Partner Profile Access Parameters – SYS001 on page 3-73 for specific parameter information and layout.
ERRCTL	Error Message and Control File This file contains descriptions of every error in the Gentran:Basic system. Error rejection is not supported by the envelope generation function.
SYS003	Input File This file contains the customers data with imbedded CNTL records. The format of the data varies depending on the source of the input. <ul style="list-style-type: none"> <li>• EBDI011M – Mapper data</li> <li>• EBDI011A – Outbound acknowledgment data</li> </ul>
SYS004	Output File This file is the output containing the customer's data and the generated envelopes. The format of the data varies depending on the destination of the output. <ul style="list-style-type: none"> <li>• EBDI011M – Input to OB Editor</li> <li>• EBDI011A – Input to OB Editor</li> </ul>
SYS005	Control Segment Generation Report This is a report generated during envelope processing. It includes the number of interchanges, groups, and transactions processed and error information.
SYS090	Partner Profile The partner profile is used for input during the envelope generation process as specified previously. The partner profile records used include control records, group records, and transaction records.
SYS091	Outbound Control File This file is used to control the generation of the envelope control numbers at the interchange, group, and transaction level.

Filename	Description
SYS095	Partner Cross Reference File This file is used to cross-reference the partner identification given on the Control records to the partner profile. <b>Note:</b> The program can access this file only when processing in the Partner/Qualifier mode.
SYS099	Suspense File This file contains the customer's data for which envelopes could not be generated.
EDICFG	System Configuration File
EDIPREL	Partner Relationship File This file is used when processing in the Relationship mode that cross-references EDI IDs to User/Partner IDs.

### Envelope Processing Parameters – SYS000

#### UCS NUMBER Record

The following table describes the UCS NUMBER Record.

Positions	Field Content	COBOL Picture	Format	Length	Description
001 – 020	UCS Number	X(20)	AN	20	Indicates UCS number structure for transaction control number to be used. Valid value: USE UCS NUMBERS FOR.
021 – 032	Version	X(12)	AN	12	Version of the standards that causes UCS numbering of the transactions.
033 – 033	Filler	X(01)	AN	01	Not used.
034 – 036	Agency	X(03)	AN	03	Indicates the numbering structure for standards agency to be used.
037 – 080	Filler	X(44)	AN	44	Not used.

#### 6/3 NUMBER Record

The following table describes the 6/3 NUMBER Record.

Positions	Field Content	COBOL Picture	Format	Length	Description
001 – 020	Alternate Record Number	X(20)	AN	20	This parameter generates 6/3 alternate numbering.
021 – 026	Transaction Set	X(06)	AN	06	Transaction set identifier of the standards used in alternate numbering sequencing.
027 – 027	Filler	X(01)	AN	01	Not used.

Positions	Field Content	COBOL Picture	Format	Length	Description
028 – 039	Version	X(12)	AN	12	Standards version to be used for alternate numbering.
040 – 040	Filler	X(01)	AN	01	Not used.
041 – 043	Agency	X(03)	AN	03	Standards agency to be used for alternate numbering.
044 – 080	Filler	X(37)	AN	37	Not used.

**5/3 NUMBER Record**

The following table describes the 5/3 NUMBER Record.

Positions	Field Content	COBOL Picture	Format	Length	Description
001 – 020	Alternate Record Number	X(20)	AN	20	This parameter generates 5/3 alternate numbering.
021 – 026	Transaction Set	X(06)	AN	06	Transaction set identifier of the standards used in alternate numbering sequencing.
027 – 027	Filler	X(01)	AN	01	Not used.
028 – 039	Version	X(12)	AN	12	Standards version to be used for alternate numbering.
040 – 040	Filler	X(01)	AN	01	Not used.
041 – 043	Agency	X(03)	AN	03	Standards agency to be used for alternate numbering.
044 – 080	Filler	X(37)	AN	37	Not used.

**5/4 NUMBER Record**

The following table describes the 5/4 NUMBER Record.

Positions	Field Content	COBOL Picture	Format	Length	Description
001 – 020	Alternate Record Number	X(20)	AN	20	This parameter generates 5/4 alternate numbering.
021 – 026	Transaction Set	X(06)	AN	06	Transaction set identifier of the standards used in alternate numbering sequencing.
027 – 027	Filler	X(01)	AN	01	Not used.
028 – 039	Version	X(12)	AN	12	Standards version to be used for alternate numbering.
040 – 040	Filler	X(01)	AN	01	Not used.
041 – 043	Agency	X(03)	AN	03	Standards agency to be used for alternate numbering.
044 – 080	Filler	X(37)	AN	37	Not used.

*PLUS Record*

The following table describes the PLUS Record.

Positions	Field Content	COBOL Picture	Format	Length	Description
001 – 004	Gentran:Plus Parameter	X(04)	AN	04	Instructs the program to generate \$\$ADD cards for Gentran:Plus. Valid value: PLUS.

*TCR Record*

The following table describes the TCR Record.

Positions	Field Content	COBOL Picture	Format	Length	Description
001 – 004	TCR parameter	X(04)	AN	04	Instructs the program to generate \$TCR user-defined information. Valid value: TCR.

*GS Time Record*

The following table describes the GS Time Record.

Positions	Field Content	COBOL Picture	Format	Length	Description
001 – 027	GS Time Parameter	X(27)	AN	27	Instructs the program to generate the GS envelope with the GS05 time field having a length equal to the maximum length allowed according to the standard. Valid value: GENERATE MAX LENGTH GS TIME.

*SNRF Record*

The following table describes the SNRF Record.

Positions	Field Content	COBOL Picture	Format	Length	Description
001 – 004	Six Character Sender's Reference Parameter	X(04)	AN	04	Instructs the program to generate six character SNRF elements for STX interchanges (ANA TRADACOMS) or six character interchange control reference UNB05 for UNB interchange (EDIFACT).

*Partner Profile Access Parameters – SYS001*

The following table describes the partner profile access parameters.

**Note:** Partner profile access parameters are ignored by the program EBDI011M. Partner access is controlled by the Outbound Mapper.

Positions	Field Content	COBOL Picture	Format	Length	Description
001 – 001	Filler	X(01)	AN	01	Not used.
002 – 002	Pass Name Information	X(01)	AN	01	<b>I</b> – Instructs the program to look up partner name information and pass to Outbound Editor.
003 – 003	Pass EDI IDs	X(01)	AN	01	<b>I</b> – Instructs the program to pass EDI IDs, not resolved Partner/User IDs to the Outbound Editor.

*CONTROL Record*

The following table describes the CONTROL Record when processing in the Partner/Qualifier mode.

Positions	Field Content	COBOL Picture	Format	Length	Description
001 – 004	Label	X(04)	AN	04	Constant CNTL.
005 – 039	Partner ID	X(35)	AN	35	Partner ID of partner receiving transactions. This field is required on at least the first CNTL record.
040 – 043	Partner Qualifier	X(04)	AN	04	Partner ID/Qualifier of partner receiving transactions. This field is required on at least the first CNTL record.
044 – 044	Record Type	X(01)	AN	01	Must be one of the following: <b>I</b> – Interchange <b>G</b> – Group <b>T</b> – Transaction <b>O</b> – Object
045 – 058	Number	X(14)	AN	14	Control number to be used on the Header/Trailer segments of the level indicated by the previous field. If this field is blank, the appropriate control numbers must be in the partner profile.
059 – 059	Test	X(01)	AN	01	<b>T</b> – Test interchange <b>Blank</b> – Production interchange. Meaningful only on the I Interchange Control record.
060 – 065	Transaction Type	X(06)	AN	06	Identifies transaction type of data.
066 – 071	Functional Group	X(06)	AN	06	Identified functional group of following data.
072 – 073	CNTL Version	X(02)	AN	02	Identifies the version of Control record. Constant <b>V4</b> .

Positions	Field Content	COBOL Picture	Format	Length	Description
074 – 097	Gentran:Plus Message	X(24)	AN	24	Text to be moved to \$\$ADD Batch ID field. Valid only at interchange level.
098 – 111	Application Reference	X(14)	AN	14	Text to be moved to Application Reference Data Elements on UNB and STX interchanges.
112 – 146	Receiver ID	X(35)	AN	35	EDI – Receiver ID Not used.
147 – 150	Receiver ID Qualifier	X(04)	AN	04	EDI – Receiver ID/ Qualifier Not used.
151 – 154	Interchange Only Switch	X(04)	AN	04	Used in EBDI011A to control TA1 only acknowledgments
155 – 155	Transaction Release	X(1)	AN	1	Message release for TRADACOMS
156 – 160	Delimiters	X(5)	AN	5	Inbound delimiters used for acknowledgment generation
161 – 162	Rail Constant	X(2)	AN	2	FA or SW used for Rail interchanges
163 – 168	Rail Transaction	X(06)	AN	6	Transaction ID used for Rail interchanges
169 – 180	Control Version	X(12)	AN	12	Optional version to be used on partner profile Lookup of Group and Transaction records.
181 – 215	UNH Common Access Reference	X(35)	AN	35	Text to be moved to the UNH Common Access Reference (UNH03). Only valid at the transaction level.
216 – 218	Multiple Envelope ID	X(03)	AN	03	Interchange Envelope ID to be used is Header Option field on Control Group, and Transaction keys for partner lookup.
219 – ??	Filler	X(??)	AN	??	The length of the Filler field varies, depending on the file the Control record is in.

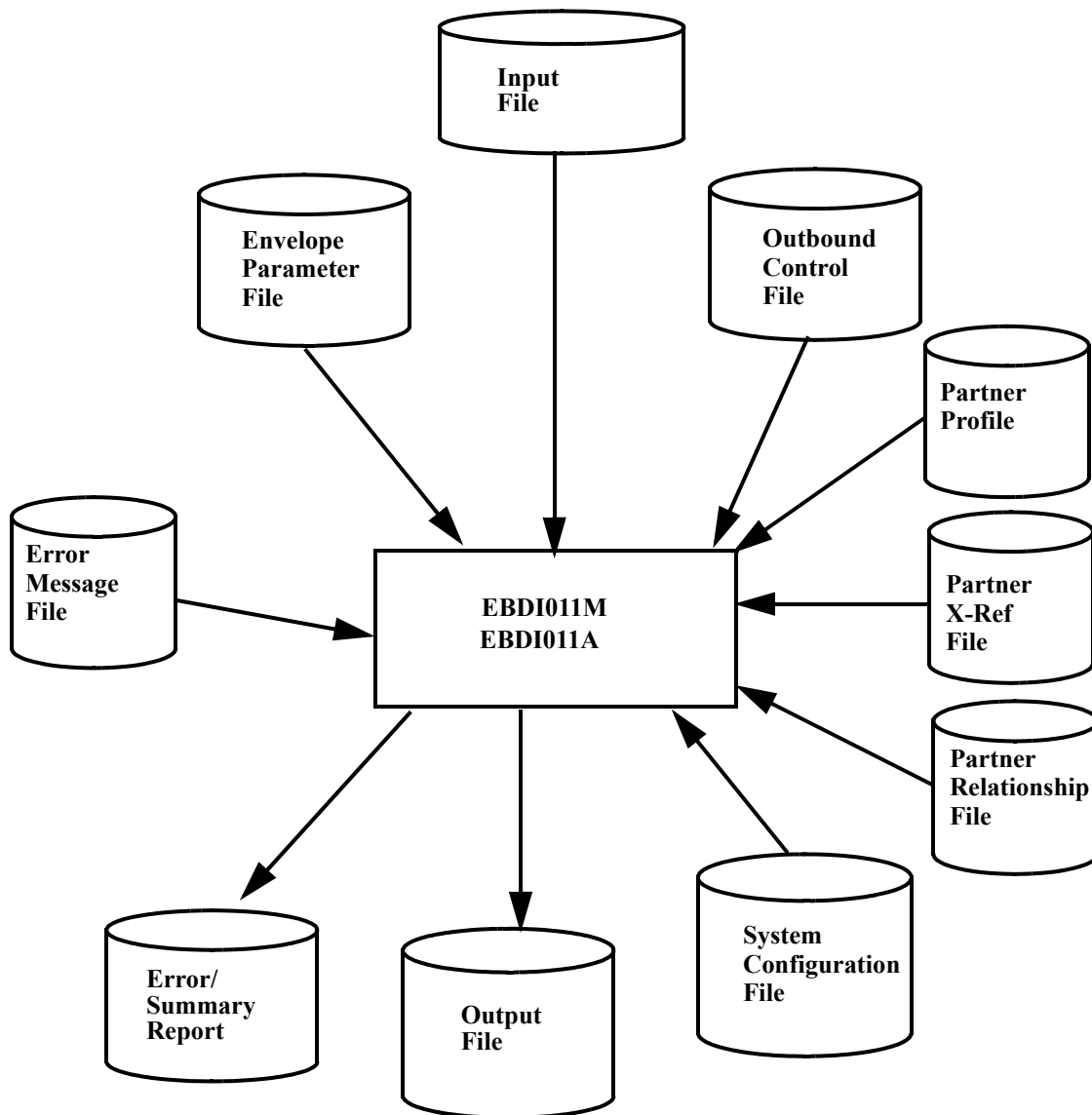
The following table describes the CONTROL record when processing in the Relationship mode.

Positions	Field Content	COBOL Picture	Format	Length	Description
001 – 004	Label	X(04)	AN	04	Constant CNTL.
005 – 019	User ID	X(15)	AN	15	User ID portion of User/Partner ID. This field is required on at least the first CNTL record.
020 – 034	Partner ID	X(15)	AN	15	Partner ID portion of User/Partner ID. This field is required on at least the first CNTL record.
035 – 035	Record Type	X(01)	AN	01	Must be one of the following: <b>I</b> – Interchange <b>G</b> – Group <b>T</b> – Transaction <b>O</b> – Object

Positions	Field Content	COBOL Picture	Format	Length	Description
036 – 049	Number	X(14)	AN	14	Control Number to be used on the Header/Trailer segments of the level indicated by the previous field. If this field is blank, the appropriate control numbers must be in the partner profile.
05 – 050	Test	X(01)	AN	01	<b>T</b> – Test interchange <b>Blank</b> – Production interchange. Meaningful only on the I Interchange Control record.
051 – 056	Transaction Type	X(06)	AN	06	Identifies transaction type of data.
057 – 063	Functional Group	X(06)	AN	06	Identified functional group of following data.
064 – 065	CNTL Version	X(02)	AN	02	Identifies the version of Control record. Constant <b>V4</b> .
066 – 089	Gentran:Plus Message	X(24)	AN	24	Text to be moved to \$\$ADD Batch ID field. Valid only at interchange level.
090 – 103	Application Reference	X(14)	AN	14	Text to be moved to Application Reference Data Elements on UNB and STX interchanges.
104 – 138	Receiver ID	X(35)	AN	35	EDI – Receiver ID Not used.
138 – 141	Receiver ID Qualifier	X(04)	AN	04	EDI – Receiver ID Qualifier Not used.
142 – 145	Interchange Only Switch	X(04)	AN	04	Used in EBDI011A to control TA1 only acknowledgments
146 – 146	Transaction Release	X(1)	AN	1	Message release for TRADACOMS
147 – 151	Delimiters	X(5)	AN	5	Inbound delimiters used for acknowledgment generation
152 – 153	Rail Constant	X(2)	AN	2	FA or SW used for Rail interchanges
154 – 159	Rail Transaction	X(06)	AN	6	Transaction ID used for Rail interchanges
160 – 171	Control Version	X(12)	AN	12	Optional version to be used on partner profile lookup of Group and Transaction records.
172 – 206	UNH Common Access Reference	X(35)	AN	35	Text to be moved to the UNH Common Access Reference (UNH03). Only valid at the Transaction level.
206 – 214	Filler	X(09)	AN	09	Not used.
215 – 217	Multiple Envelope ID	X(03)	AN	03	Interchange Envelope ID to be used is Header Option field on Control Group, and Transaction keys for partner lookup.
218 – ??	Filler	X(??)	AN	??	The length of the Filler field varies depending on the file the Control record is in.

*Outbound Envelope Generation*

Figure 3.7 illustrates the Outbound Envelope Generation process.



**Figure 3.7 Program EBDI011M and EBDI011A**



---

## Outbound Editor

## EBDI002

The Outbound Editor is responsible for validating data formats, compliance checking data to the standards, databank controls, data wrapping, control records for data separation, and error rejection.

The Editor uses the information prepared in other parts of the system. Global parameters and the partner profile give flexibility to data processed by the Editor.

### Global Parameters

Global parameters are used as input in both of the Editor (EBDI001 and EBDI002) programs. Global parameters, enable you to use features within Gentran:Basic and set such options as the standards for compliance checking, level of detail for reports, and level of databank processing to be performed. Global parameters and partner profile are used in conjunction to control the system's operation.

Global parameters are divided into two types: System-level and Runtime. Both types are described below.

### *System-Level Global Parameters*

System-level global parameters define to the Editors which system features and options that you want to use at your facility. For previous Gentran users, the global parameter records used by the Outbound Editor and now identified as system-level are as follows.

#### *General Processing Options*

- ICS Tables
- Code Check
- Error Report

#### *Databank Processing Options*

- Partner Databank
- Outbound EDI

#### *Partner Processing Options*

- Partner Version
- Verify Partner

#### *Partner Look up Options*

- GS Sender Qualifier
- GS Receiver Qualifier

#### *Rejection Processing*

- Partner Error Control

*Data Separation Options*

- Directed Output Files
- Split by Partner

*Outbound Wrapping Options*

- Compress

*EDIFACT/TRADACOMS Options*

- Delete Optional Blank Elements
- Delete All Blank Elements

Once set, these options typically experience little change and represent the system configuration used by the majority of your outbound runs. System-level global parameters reside on the Configuration file and are set up through the online system. **All system-level global parameters can be overridden at runtime.**

See Appendix A, Global Parameters, for runtime override information.

See Chapter 5 in the *Gentran:Basic for zSeries Release 6.4 User's Guide* for additional information on system-level global parameters.

*Runtime Global Parameters*

Runtime global parameters provide instructions to the Editor at execution time and may vary from run to run. For previous Gentran users, global parameter records defined as Runtime are as follows:

- All or Nothing
- Detail Report Always
- Use Subelement Separator For ANSI
- Use Subelement Separator For XXXXXXXXXXXXXXXbXX
- Output Databank Only
- Version Conversion
- Implementation Convention Conversion

These global parameters are set up as instream parameters (SYS030) in the JCL. This group of parameters also include override parameters for system-level global parameters.

*See Appendix A for runtime override information.*

## Sequence of Processing

All system-level and Runtime global parameters are processed by the Outbound Editor during its start-up and initialization process. If all of the values in the system-level global parameter record located on the Configuration file are not set up, the Editor uses its built-in default values for any missing values.

See Chapter 5 in the *Gentran:Basic for zSeries Release 6.4 User's Guide* for additional information on system-level global parameters, or Appendix A of this guide for default value information.

The sequence of parameter processing performed by the Editor is as follows.

1. *Defaults*. The Editor initializes parameters based on its internal default values.
2. *System-level Global Parameters*. The Editor overlays default settings with these values.
3. *Runtime Global Parameters*. The Editor loads internal tables with Runtime global parameters and overlays system-level global parameter values with its corresponding override parameter value, if present.

**Note:** The Runtime global parameters are processed in the same sequence they are read. If you accidentally have more than one Runtime override parameter for the same system-level global parameter, the last one read is used.

## Partner Profile Options

Prior to Gentran:Basic 5.0, both mapping and the Editor in the outbound flow duplicated much of their processing against the partner profile. The mapping process looked up the interchange, group, and transaction records to generate envelopes. The Editor looked up interchange, group, and transaction records on the partner profile during its compliance processing. This duplicate processing has been eliminated. The outbound mapping process looks up the information for envelope generation and retrieves the information needed by the Editor.

## ###MAP Records

The Outbound Mapper program (Envelope Generation if mapping is not used) creates pass-thru ###MAP records for the Outbound Editor. The Map records contain:

- Resolved Sender/Receiver Partner IDs and Qualifiers (Partner/Qualifier mode) or User/Partner ID (Relationship mode)
- Version, Version-To-Use, Verify, Send, and Databank flags
- Other information from the partner profile.

All information needed by the Outbound Editor except data separation and error rejection comes from the Map records. This improves performance by eliminating duplicate partner lookup processing whenever possible.

By having partner information resolved in the mapping process, the number of Global parameters needed by the Outbound Editor is significantly reduced over the Inbound Editor.

When the partner information is actually coming from the Map record, the (###Map record) symbol is used.

## Global Parameters Processing

Processing begins with the Runtime global parameters. These parameters are positional (pre-described format) and are ignored by the Editor if they start (position 1) with a space, 'X' or '\*'. Otherwise, they are validated by the Editor and an error message (Error Number 558) displays on the Error report should a mistake be detected. The global parameters give instructions to the Editor and the error message is only a warning that some of these instructions were not understood.

Valid global parameters are used to update the Editor's internal decision tables. Throughout this section, global parameters are included where they are actually used by the Editor.

With the exception of input records containing '###' in their Segment IDs, all records are validated against the Table file for compliance checking to the standards.

## Compliance Checking to Standards

### *Delimiters*

Delimiters control and separate individual EDI segments and elements. Interchange Control Headers (i.e., ISA, ICS, BG, UNB/UNA, and STX/SCH) set delimiters or default delimiters to pre-determined values. Once the delimiters are set, each functional group, transaction, and data segment within the interchange can be translated. Inbound, the interchange Control Header sets delimiters for each interchange. Outbound, the partner profile sets delimiters for each interchange. Each delimiter has a unique value within the interchange. Gentran:Basic uses the following delimiters:

- **Segment Separator**  
ANA/TRADACOMS only, this is a special delimiter used to delimit the segment identification from the elements. The default value is an '=' (equal) sign.
- **Element Separator**  
All interchanges use element separators to delimit individual elements. Each element can be either a simple element or composite element (containing subelements).
- **Subelement Separator**  
Component Elements or subelements are supported within Gentran:Basic. The subelement separator is an additional delimiter that separates the component pieces of a composite or group element. A group element is collection of elements that are functionally connected. The 'subelement' is the component piece. A composite element defines a relation of individual subelements.
- **Repetition Separator**  
Repeating data elements and repeating data structures are supported within Gentran:Basic. The repetition separator is an additional delimiter that separates each occurrence of a repeating data element (simple element) and/or data structure (composite element).
- **Release Character**  
For EDIFACT and TRADACOMS, String element type (AN) or ID element types can use release character delimiters. Release character delimiters indicate that the character following the release character is part of the string and not a defined

delimiter (element separator, segment separator, subelement separator, or segment terminator). Release characters are not included in maximum field lengths.

See Element Compliance for more details for string and ID element types.

- **Decimal Notation**  
For EDIFACT and TRADACOMS, a decimal point can have a period (.) or comma (,) as a value. For ANSI/X12, the default value a decimal point is a period(.).
- **Segment Terminator**  
All interchanges use segment terminator to signal the end of EDI segments.

### Standard Version Selection

Gentran:Basic Editors use information contained in the envelopes to identify which standard version and version agency to use for compliance editing. This occurs at the interchange, group, and transaction level depending on the type of interchange.

Standard versions and transactions that are used in compliance editing are pre-selected by means of the online Standards Maintenance – Transaction in Use screen. The selected versions and transaction are downloaded through the Selective Download Process (EBDI030) program. The optimized table is the output of this process and is used by the Editors to compliance edit the data.

The Editor uses the information in the envelopes to determine which version/transaction to select from the optimized Table file. A combination of version, agency, and transaction ID is used in this selection. Gentran:Basic builds a 12-byte Standard Version Identifier from the envelopes as follows:

#### Envelopes

ISA12	==>	Interchange Standard Version Identifier
STX0101,STX0102	==>	Standard Version Identifier
ICS03	==>	Interchange Standard Version Identifier
GS08	==>	Standard Version Identifier
UNG0501, UNG0502, UNG0503	==>	Standard Version Identifier
UNH0201, UNH0202, UNH0204	==>	Standard Version Identifier

The Standard and Version Identifier is defined as follows:

Version Number	Version Release	Version Association
Bytes 1 – 3	Bytes 4 – 6	Bytes 7 – 12

Because various standard committees use the same version, an agency code is now required to differentiate the versions. The envelope agency is used with the version to select a standard unless there is a version association present. This version association is checked against the standards Association file, EDISASC. The file is a list of association/agency relationships defined to Gentran:Basic. If a version association is detected by the Editors, the association is used to identify the corresponding agency. This information is provided to the Batch Table Optimization process.

Gentran:Basic uses the Standard Version Agency from the envelopes as follows:

### Envelopes

GS07	==>	Standard Version Agency
UNG04	==>	Standard Version Agency
UNH0203	==>	Standard Version Agency

See Chapter 5 in the *Gentran:Basic for zSeries Release 6.4 User's Guide*, and the topic "Selective Standards Download (EBDI030) program" in Chapter 4 of this guide for additional information.

After identifying and loading either control syntax or user transaction document, the Editor begins compliance checking. Gentran:Basic validates every detected transaction, segment, and element to the supported standard. This is done to ensure absolute adherence to the supported EDI Standards. This is done to ensure every EDI Standards requirement is properly implemented. The Editor validates segment compliance, element compliance, and code list compliance.

### Segment Compliance

Segment compliance validates each non-suspended EDI segment. The Editor scans all input EDI data. Segments are checked for placement within the identified transaction. Gentran:Basic validates segments that start with either two- or three-byte segment identification. Repeated segments within the identified transaction are checked for maximum usage and maximum looping. Gentran:Basic verifies mandatory segments for active loops. Segments are classified as control segments or data segments according to the following criteria.

- Any segment that is defined to the identified transaction and is positioned between the transaction control Header segment and transaction control Trailer segment is considered a Data segment.
- Any segment that is not included within an identified EDI transaction is considered either a Control segment or invalid Data segment.
- Non-EDI data is considered anything prior to the interchange Header and after interchange Trailer. Interconnect information and network specific records are not processed by Gentran:Basic. The Inbound Editor suspends and bypasses this information. Pass-Thru '###' records, \$TCR, and \$\$ADD records are not considered EDI data and are restricted to passing information between processes.
- Any data that is defined to the identified Package Structure and is positioned between the Object Header segment and Object Trailer segment is considered Binary data. The Outbound Editor does not compliance check binary data, it only passes it through.

Each Editor checks Audit ID indicators, conditional codes, and segment activities during segment and element compliance editing. Auditing User fields are extracted and processed after each element.

See the topics "Online Standards Maintenance" and "Segment Element Maintenance" in Chapter 4 for additional information on Audit ID Indicators.

Conditional codes and Segment Activity requirements are processed after each element is validated.

See Chapter 5 in the *Gentran:Basic for zSeries Release 6.4 User's Guide* for information on Online Standards Maintenance, Conditional Code Maintenance, or Segment Activity Maintenance.

After the segment has been found within the identified transaction, segment elements, element dictionaries, and optional code lists are used to validate individual elements. Each element is passed through element compliance.

### *Element Compliance*

The Editor uses the element types defined in each dictionary entry within the batch optimized tables to edit each element. Gentran:Basic supports an 18-digit floating point number for internal accumulators and hash totals. The Editor has element types divided into four major classifications. Each classification is separated into:

#### **Date and Time Types**

DT	- YYYYMMDD	- Standard Date – fixed 8 bytes – 003072 – X12
	- YYMMDD	- Standard Date – fixed 6 bytes – pre 003072 – X12
D8	- YYYYMMDD	- Standard Date – fixed 8 bytes – EDIFACT
CD	- MMDDYYYY	- Alternate Format – fixed 8 bytes – international
TM	- HHMM	- Standard Time – fixed four bytes – X12
T6	- HHMMSS	- Standard Time w/option seconds
T8	- HHMMSSHS	- Standard Time w/option seconds/hundredths

#### **Real Decimal Types**

R0 through R9 and R- Maximum decimal precision is 9 digits to the right of the decimal. Real number usage, decimal point, and sign are consider in length of the element. The element type validates decimal point placement. Hex characters '00' or Low values are unacceptable.

#### **Implied Decimal Types**

N0 through N9 – Maximum decimal point is 9 digits to the right of the implied decimal. The element type determines the decimal pointer placement. Hex characters '00' or Low values are unacceptable.

#### **String Types**

AN – String elements can be supported. Special characters, spaces, period, comma, uppercase/lowercase letters, and numeric characters can be contained in a string type. String types can support a code list up 40 bytes in length. Code verification is active for string types.

## ID Types

ID – Code types are a special string type. ID types contain all numeric characters or all letters. ID types can support code lists up to 40 bytes in length. Code verification is active for ID types.

## Binary Type

B – Code types are supported but cannot be edited because of the character interpretation.

## Code List Compliance

Code checking is optional in Gentran:Basic. Code list compliance can be configured through the Administrative Main Menu, in the Global Parameter Maintenance subsystem.

Online code files are available to the Editors. The Editor determines which code file to use based on the maximum length, as defined by the standards, of the element detected.

File	Minimum Length	Maximum Length
EDISCD1	1	5
EDISCD2	6	10
EDISCD3	11	20
EDISCD4	21	40

The Editors keep an internal Smart Code List of valid codes. This code list is a stack of code values and provides a buffer of code values for the Editors to use during element validation within each transaction. The stack is LIFO (last-in-first-out), which means that when a new valid code is placed on the stack, the last valid code drops off the stack. The code file is checked only when the code value is not on the Smart Code List.

## Simple Codes

Simple Codes are the most common type of codes. With Simple Codes, strings of characters are used and validated against one of four online standards code files. Simple codes are validated individually.

## Composite Codes

Composite Codes have multiple parts to the detected code value. The Composite Code definition is used to substring each code value. Each part is validated against one of the four code files. If any part of the code is invalid, the complete composite code is invalid.

## Document Compliance

### Error Handling

When compliance checking a document, three results can occur from the Editor:

- Document is clean – no errors.
- Document is suspended.
- Document has errors.



### Document Is Clean

A clean document is a document with no errors.

### Document Is Suspended

Suspension automatically occurs if:

- The standards version selected is not on your Table file.
- The interchange envelope is missing key information such as Sender ID, a separator, or a terminator.
- The segment contains data that the Editor cannot interpret.

Suspension is not always automatic; you can control suspension by using global parameters. If the VERIFY global parameters are active and the partner profile is not found, the Editor suspends. If the VERIFY global parameters are active and the partner profile is found, but the receive flag on the group or transaction record is not set to Y, the Editor suspends. If the global parameters PARTNER VERSION ALWAYS INTERCHANGE or PARTNER VERSION ALWAYS GROUP are active and the partner version on the group/transaction record is either spaces or not on the Table file, the Editor suspends.

### Document Has Errors

Gentran:Basic considers all errors as warnings unless you indicate differently. Two types of errors can occur in the Gentran:Basic system, compliance and Gentran:Basic (i.e., noncompliance related). Currently the error handling process treats both types of errors equally.

### Error Message and Control File

In Gentran:Basic, all errors have a numeric value assigned to them. These error numbers with a description are maintained on the Error Message and Control file. Along with the description, each error message record has two flags for error rejection processing. There is one flag for inbound and one flag for outbound. As shipped, these flags are set to spaces. You update the flags on the error message and Control file through the error rejection screen on the online partner system and the online error message system.

See the topic “Error Rejection Screen (EDIM055)” in Chapter 2 of the *Gentran:Basic for zSeries Release 6.4 User’s Guide* for additional information.

### Error Rejection Processing

When using the Gentran:Basic Error Rejection feature, the Outbound Editor uses information from the global parameters file, the partner profile, and the Error Messages and Control file to determine what action to take when compliance errors occur.

Each error message has an associated return code. Error message text and additional user information are associated with errors generated by the Editors. The Inbound and Outbound Editors have separate return codes, which provide control over error processing.

You can use the Print flag control in Error Message Maintenance to suppress error messages or modify return code values.

See the topic “Administrative Main Menu, Error Message maintenance” in the *Gentran:Basic for zSeries Release 6.4 User’s Guide* for additional information.

### *Global Parameters File*

In the system-level global parameters file, the PARTNER ERROR CONTROL flag must be set to **Y**. This flag simply tells the system that Error Rejection is being used. The system, in turn, uses the Error Message and Control file and the partner profile to determine what actions to perform.

### *Partner Profile*

In the partner profile, error rejection records must be coded. To reject outbound data, either:

- create error rejection records for the Receiver ID (Partner/Qualifier mode) or the Partner ID (Relationship mode) of the interchange or group being sent (i.e., BG, ISA, GS, and ICS).
- OR**
- create error rejection records for the Global Partner (!!!GENTRAN-RESERVED-PARTNER- ID-1 for the Partner/Qualifier Mode or !!!GENTRAN-RU1/ !!!GENTRAN-RP1 for the Relationship Mode).

The information in the partner error rejection record indicates the level (partner, group, or transaction), the specific error number, and whether this error is to be rejected inbound, outbound, or both.

### **Levels of Error Rejection**

Gentran:Basic enables you to control Error Rejection at three levels for inbound and outbound processing.

**Note:** If error rejection occurs at any level in the Outbound process, the entire interchange is rejected.

#### **Partner Level**

This level specifically refers to errors at the interchange Header/Trailer level for inbound and outbound levels. It enables you to control errors that occur on the interchange Header/Trailer segments.

#### **Group Level**

The Group ID is the key to the partner profile for this level, specifically referring to errors at the group Header/Trailer level for inbound and outbound levels. It enables you to control errors that occur on the group Header or Trailer segments. When using the group level you have the option of coding this record for a specific group (e.g., purchase orders, invoices) or for all groups. To code the partner error record for a specific group, enter the specific Functional Group ID (e.g., PO, IV) into the Level ID field. To code the partner error record for all groups, code the Level ID field with ALL. The effect of coding ALL causes the system to perform the indicated action (reject inbound or outbound) no matter what type of group Header Functional ID is in the outbound data. The coding of specific Functional Group IDs enables you to control error rejection on a group-by-group basis.

#### **Transaction Level**

The Transaction Set ID is the key to the partner profile for this level, specifically referring to errors at the transaction set level for inbound and outbound levels. It enables you to control errors that occur on any segments within the transaction Header and Trailer segments. When using the

transaction level, you have the option of coding this record for a specific transaction (e.g., 875, 880), or for all transactions. To code the partner error record for a specific transaction, enter the Transaction ID into the Level ID field. To code the partner error record for all transactions, code the Level ID field with ALL. The effect of coding ALL causes the system to perform the indicated action (reject outbound) no matter what type of Transaction Set ID is in the outbound data. The coding of specific Transaction Set IDs allows the customer to control error rejection on a transaction by transaction basis.

**Note:** When processing Package Structure documents and a specific transaction is desired, the Transaction Set ID must be set to **UNO**.

### **Error Numbers in Error Rejection.**

The types of errors which can be rejected are determined by their identifying number. Every error is described within Error Messages. When the system is processing inbound or outbound data and a compliance error occurs, the action taken when using Error Rejection is determined by whether that error number has a corresponding partner error record. If a partner error record exists for this error number, the action specified in the error record is taken (reject outbound). Each error you wish to control must have a corresponding partner error record. By default, the Gentran:Basic system treats compliance errors as Warnings.

Only applicable when used with system-level global parameter, PARTNER ERROR CONTROL.

### **Direction Control in Error Rejection.**

The system provides two fields, Inbound Rejection and/or Outbound Rejection, when determining the action for each specific error.

Gentran:Basic enables you to control each error number and its corresponding action separately for inbound and outbound processing. It also enables you to set up an error number and its corresponding action for both inbound and outbound processing on the same error record.

The Inbound Rejection field tells Gentran:Basic whether you are rejecting this error on incoming data. The possible values include **Y** (Yes) or **N** (No).

The Outbound Rejection field tells Gentran:Basic whether you are rejecting this error on outgoing data. The possible values include **Y** (Yes) or **N** (No).

### ***Error Message and Control File***

The Error Message file contains all compliance error messages issued during the Inbound and Outbound Editor processing. In addition to the error messages, the file contains each error number and two special fields, Rejection Decision Inbound and Rejection Decision Outbound. The combination of messages, numbers, and fields made up of three levels used by the Error Rejection feature.

The following provides information about the error message fields and values.

#### **Rejection Decision Inbound**

The Rejection Decision Inbound field on the error message record is used by the Inbound Editor (EBDI001) when this error occurs during inbound processing.

**Rejection Decision Outbound**

The Rejection Decision Outbound field on the error message record is used by the Outbound Editor (EBDI002) when this error occurs during outbound processing.

**Level Values for Rejection Decision Inbound and Outbound**

Two values impact how the Editors proceed with this error. Those values are the same for each direction.

**Inbound and Outbound Values**

- A** – For inbound or outbound processing, the value A causes the system to reject this interchange, group, or transaction regardless of what is coded on the partner profile. Enter A in these fields in the Message Maintenance subsystem.
- P** – For inbound or outbound processing, the value P tells the system that this error is coded by specific partner. For inbound processing, the system reads the partner profile error records based on the incoming Sender ID (Partner/Qualifier mode) or the User ID (Relationship mode) to determine the action. For outbound processing, the system reads the partner profile error records based on the outgoing Receiver ID (Partner/Qualifier mode) or the Partner ID (Relationship mode) to determine the action. The P value is specified in these fields when Error Rejection is set up for a specific Partner ID at the partner level and rejection Inbound or Outbound is coded Y.
- D** - The D value tells Gentran that this error is coded for the global partner (!!!GENTRAN-RESERVED-PARTNER). Gentran reads the partner profile for the global partner to determine the action. The D value is assigned when Error Rejection is set up at the global partner level and the Rejection Inbound or Outbound is coded Y.

**Error Rejection with the Databank****All or Nothing**

When activated, this Runtime global parameter instructs Gentran to provide only valid, error-free updated data to the databank. The parameter monitors all interchanges and prevents and stores data with errors from passing through to the trading partner. When an error occurs, all databank directory and message store information for this run is written “interrupted.” After correcting the errors, either by the inbound trading partner or through the outbound application, Gentran releases the data for translation.

Errors produced by the Editor and described in the System Message Guide activate the interrupt action, while continuing the current run. After the current run is complete, two events occur:

1. The current updates that have been made to the databank, both directory and message store, are indicated as errored databank information.
2. The external user return code equals 0016.

By running the databank Housekeeping functions, removal of this interrupted databank information can be performed.

See the topics “Outbound EDI Databank Maintenance (EDID201)” and “Inbound EDI Databank Maintenance (EDID301)” for more information.

**Note:** This Runtime global is used for EDI data that has known syntax and compliance problems either through application or inbound partner. Use of the global should be limited to this situation.

## Error Rejection Numbers

This section will assist you in setting up Error Rejection. The section lists both inbound and outbound processing error numbers. It is up to YOU to choose which errors are important to your processing.

See the *Gentran:Basic for zSeries Release 6.4 System Message Guide* for more details on error messages.

### *Error Rejection Numbers – Outbound*

#### Error Numbers at the Partner Level

The following are error numbers at the Partner level:

ICS	ISA	BG	GS	EDIFACT	TRADACOMS
	21	21	21	21	21
21	22	22	22	22	22
22	78	142	207	142	147
23	142	207	218	207	207
24	207	218	532	534	
25		219	533	535	
26		220	534	538	
29		380	535		
40			536		
41			538		
42			539		
43			568		
44					
78					
142					
207					
221					

#### Error Numbers for all Interchanges

The following are error numbers for all interchanges:

20,45,46,67,183,354,355,356

#### Date and Time Error Numbers

The following are Date and Time error numbers:

32,34,35,36,37,38,39,83

**Error Numbers at the Group Level**

The following are error numbers at the Group level:

47,48,49,55,67,141,195, 562

**Error Numbers at the Transaction Level**

The following are error numbers at the Transaction level:

50,51,52,53,54,57,60,61,62,63,64,65,67,79,80,81,82,  
84,85,86,87,88,89,90,91,92,93,131,140,145,146,147,326,  
327,385,386,387

**Databank Error Numbers**

The following are Databank error numbers:

689

A standard contains different record types: Version record, Transaction record, Segment record, Element record, Element Activity record, Element Dictionary records, and Code records.

See the topic “Gentran:Basic Tables File” for further information on standard records.

**Edit Criteria**

Each element of the interchange is validated against the standard previously selected. The primary edit considerations for an interchange include:

1. Is the element mandatory or optional? If mandatory, is it present? If not, data is suspended.
2. Is the element within the minimum/maximum field size established by the standard? If not, data is suspended.
3. Is the element in proper format (AN, Nx, Rx, ID)? If not, data is suspended.
4. If applicable, does it contain a proper value as found in the Code record?
5. What global parameters are in effect?

**Processing Methods**

Outbound documents can be processed using either standard enveloping or deferred enveloping. The following sections describe these methods of process.

***Standard Enveloping***

Outbound documents contain interchange envelopes (i.e., BG, ISA, ICS, GS, UNA, UNB, SCH, STX), group envelopes (i.e., GS, UNG, BAT), transaction envelopes (i.e., ST, UNH, MHD), and Object Header envelope (i.e., UNO). After validation by the Editor, these documents are ready to transmit.

### *Deferred Enveloping*

Outbound documents contain interchange, group, and transaction control records. These control records have a Segment ID of CNTL and contain a record type of I (interchange), G (group), T (transaction), or O (object). To create a deferred envelope, set the Envelope Suppression Switch on the outbound Mapper (EBDI042) parameter to **Y**.

See the topic “The Outbound Mapper” in this chapter for further details.

### **Deferred Enveloping Restrictions**

The following Gentran:Basic features are not available when you implement deferred enveloping:

- Deferring TRADACOMS data
- WINS 980 and X12 831 transactions
- Deferred processing of outbound acknowledgments
- CNTL Pass-Thru records in EBDI042
- Mixed processing of deferred and standard (nondeferred) data per run

### **Interchange Envelope Processing (Standard Enveloping)**

Interchange envelopes must have a Segment ID of BG, ISA, ICS, GS, UNA, SCH, UNB, or STX. Anything else causes the input records to be written to the suspense file until a valid interchange is found. Also, an error report is generated.

Compliance checking begins with selecting the version of the standards to be used from the Table file.

### *Domestic*

#### **ISA Interchange**

Segment element ISA12 contains the version number to use to validate the group envelope. However, this version can be overridden through system-level global parameters. PARTNER VERSION ALWAYS INTERCHANGE uses the version from the partner profile and ignores the version on the ISA interchange. PARTNER VERSION INTERCHANGE instructs the Editor to look at the value found (A, I, or D value), the Editor responds accordingly. The Editor uses either the version from the ISA12 element or the partner control record version. If the version is not found, the Editor checks the VERSION CONVERSION records from the Runtime global parameters. If a match is found, the alternate version from the version conversion record is substituted. If this search fails to find the version on the Table file, the document is suspended.

#### **ICS Interchange**

Segment element ICS04 contains the version number used to validate the group envelope. However, this version can be overridden through system-level global parameters. PARTNER VERSION ALWAYS INTERCHANGE uses the version on the ICS interchange. PARTNER VERSION INTERCHANGE instructs the Editor to look at the partner control record for the version to use flag. Based on the value found (A, I, or D value), the Editor responds accordingly. The Editor uses either the version from the ICS04 element or the partner control record version. If the version is not found, the Editor checks the VERSION CONVERSION records from the Runtime global parameters. If a match is found, the alternate version from this version conversion record is substituted. If this search fails to find the version on the Table file, the document is suspended.

**GS Interchange**

Normally considered a group envelope, the GS segment can also be an interchange in some TDCC Standards. Segment element GS08 contains the version number to be used to validate the transaction. However, this version can be overridden through system-level global parameters. PARTNER VERSION ALWAYS GROUP uses the version from the partner profile (Map record) and ignores the version on the GS08 group. PARTNER VERSION GROUP instructs the Editor to look at the partner group record (Map record) for the version to use flag. Based on the value found (A, I, or D value), the Editor responds accordingly. The Editor uses either the version from the GS08 element or the partner control record version. If the version is not found, the Editor checks the VERSION CONVERSION records from the Runtime global parameters. If a match is found, the alternate version from the version conversion record is substituted. If this search fails to find the version on the Table file, the Editor suspends the document.

*International***UNA Interchange**

Although considered an interchange, this segment is not validated against the Table file. It is an option in EDIFACT to assign: Component and Data Element Separators and Repetition Separators, Segment Termination character, Decimal Character, and Release Character for processing the EDIFACT records.

**UNB Interchange**

This EDIFACT interchange uses the DEFAULT version from the Table file for validation processing. The actual version number to be used for validating the remaining segments is located in the Group Envelope (UNG) or the Transaction Envelope (UNH).

**UNB\*UNOA:1**

Where UNOA indicates readable characters  
(such as: +?')

Prior to Syntax version 4

**UNB\*UNOB:1**

Where UNOB indicates nonreadable characters  
(Hex 1F, Hex 1D, and Hex 1C)

Interchange processing has two delimiter syntax levels:

<b>SYNTAX A</b>	<b>Character</b>	<b>Hex Value</b>
Component Element Separator	!	7A
Data Element Separator	+	4E
Repetition Separator	*	
Decimal Notation	. or ,	
Release Character	?	6F
Segment Terminator	'	7D

Prior to Syntax version 4

<b>SYNTAX B</b>	<b>Character</b>	<b>Hex Value</b>
Component Element Separator		1F
Data Element Separator		1D
Decimal Notation	. or '	
Release Character		
Segment Terminator		1C



Although not recommended, Syntax B Delimiters are Teletype characters. They could be meaningful to many network or communication subsystems. If determining use, first discuss with your trading partner utilizing Syntax B delimiters.

### SCH Interchange

Similar to utilization as UNA, it is not validated but is not used to assign a segment separator component element separator, element separator, release character, and segment terminator for the interchange.

### STX Interchange

This TRADACOMS interchange, as with UNB, the Gentran:Basic Default Version used validates this segment. The version number in STX01 loads at transaction validation (i.e., message identification) processing. The default delimiter includes the following:

DEFAULT DELIMITER	Character	Hex Value
Segment Separator	=	7E
Component Element Separator	:	7A
Element Separator	+	4E
Segment Terminator	'	7D
Release Character	?	6F

## Global Parameter Overrides

### Partner Version Always Interchange

Instructs the Editor to look at the partner profile for the interchange version. If the Editor does not find a partner record, it generates an error message. The Editor next checks the VERSION CONVERSION Runtime global parameter to see if an alternate version has been selected. If Editor does not find a match, it suspends the document.

### Partner Version Interchange

Instructs the Editor to look at the partner profile (Map record) or the interchange Version-To-Use flag. The proper values for the Version-To-Use flag are:

- A** – *Always*. Instructs the Editor to use the partner interchange version.
- I** - *Input and Space*. Instructs the Editor to use the input version from the interchange record.
- D** - *Default*. Instructs the Editor to first use the input interchange version, with the version from the partner interchange record as a default version to try.

If the Version-To-Use flag contains A, the Editor does not perform a second attempt to find a version. If the Editor does not find the version from the Interchange record, it generates an error message and suspends the document. If the Version-In-Use flag contains I, space, or D, the Editor performs a second attempt to find the version by referencing the global version conversion records before it generates an error message and suspends the document.

### ICS Tables On

ICS interchanges have two slightly different definitions. The “OLD” ICS format defines the control standards identifier as four positions and the “NEW” ICS interchange defines the control standard identifier as one position.

This system-level global parameter instructs the Editor to accept without error only the new format. If you select this option, the Editor generates an error message every time it encounters the old format. The Editor does not suspend for this error.

**Note:** To prevent the document from entering your application system, select error rejection.

The default parameter value is ICS Tables Off. For this value, the Editor accepts either definition of the ICS interchange. The Editor generates an error message for errors found during compliance checking of the ICS against the standards.

### **Verify Partner Interchange**

The Editor verifies the partner on the partner profile (Map record). The Editor suspends data not passing verification.

The element separators determines the start and end of a segment element. For ISA, ICS, GS, and BG interchanges, the Editor uses the first character found after the Segment ID as the element separator. EDIFACT is handled differently. The UNA interchange is optional in North America and contains the values to be used for the element separator, component separator, repetition separator, and segment terminator. Otherwise, the UNB interchange chooses the separator and segment terminator values by referencing standard default values.

## **Group Envelope Processing (Standard Enveloping)**

Group envelopes must have a Segment ID of GS, UNG, or BAT. If missing the Segment ID, the Editor suspends the input records and generates an error report until either it encounters an end of input or it finds a valid interchange.

Compliance checking begins with version selection from the Table file. Elements GS08, UNH02 or UNH07, and STX01 contain the version to load from the Table file. If the Editor cannot find this version, it checks the VERSION CONVERSION record from the Runtime global parameters to see if it should use another version number. If so, the Editor makes a second attempt at loading the version from the Table file. The Editor attempts only two reads. If, on the second attempt, the Editor still cannot find the version, it suspends the data and generates an error report.

### **Partner Version Always Group**

This system-level global parameter instructs the Editor to look at the partner profile for the group version. If the Editor does not find a group partner record, it generates an error message. The Editor next checks the VERSION CONVERSION global parameter to see if an alternate version has been selected. If no match is found, the Editor suspends the document.

### **Partner Version Group**

Instructs the Editor to look at the partner profile for the Group Version-To-Use flag. The proper values for the Version-To-Use flag are:

- A** – *Always*. Instructs the Editor to use the partner group version.
- I** - *Input and Space*. Instructs the Editor to use the version from the input group envelope.
- D** - *Default*. Instructs the Editor to first use the input group envelope version, with the version from the partner group record as a default version to try.

The Editor does not perform a second attempt to find the version if the Version-To-Use flag contains A. If the Editor does not find the version from the group record, it generates an error message and suspends the document. If the Version-In-Use flag contains I, space, or D, the Editor performs a second attempt by referencing the global version conversion records before it generates an error message and suspends the document.

#### **Verify Partner Group**

This Runtime global parameter instructs Gentran:Basic to read the partner profile (Map record) using the Receiver ID (Partner/Qualifier mode) or the User/Partner ID (Relationship mode) and Functional ID from the group envelope and verify that the group record exists. If the Editor does not find the group record, it generates an error report and suspends the data until either it reaches the end of file or finds a valid interchange.

### **Transaction Envelope Processing (Standard Enveloping)**

Transaction set envelopes must have a Segment ID of ST, MHD, or UNH. If missing a valid Segment ID, Gentran suspends the input and generates an error report until either encountering an end of input or being updated with a valid interchange.

The transaction set envelope only contains two elements: Transaction Set ID and a control number. The Editor checks the Transaction Set ID against the transaction set numbers from the version selected on the group envelope. If not found, the Editor suspends the data.

Gentran:Basic uses the control number element in conjunction with the control number in the Trailer (SE, UNT) segment to verify segment integrity and prevent duplicate processing of data should the Databank Directory (Audit) parameters be in effect. Data suspension occurs for a missing control number. However, if the Trailer control number is different, the Editor generates only an error message unless Partner Error Control is on.

#### **Verify Partner Transaction**

This global parameter instructs Gentran:Basic to verify that a transaction record exists for this Transaction Set ID on the partner profile (Map record). If the Editor does not find a Transaction record, it suspends the data.

### **Object Envelope Processing (Standard Enveloping)**

Object Header envelopes must have a Segment ID of UNO. If missing the Segment ID, UNO, the Editor suspends the input and generates an error message until it either encounters the end of input or finds a valid interchange.

Gentran:Basic uses the Package Reference Number and the Length of Object Elements in conjunction with the Package Reference Number and Length of Object in the Trailer (UNP) segment to verify segment integrity. Data suspension occurs if either one of these two components are missing from the Object Header envelope.

#### **Verify Partner Transaction**

This global parameter instructs the Editor to verify that a transaction record exists for Transaction Set ID of UNO on the partner profile. If the Editor does not find the Transaction record, data suspension occurs.

## Segment Processing (Standard Enveloping)

After completing processing of the interchange, group and transaction Header envelopes, data suspension cannot no longer occur. From this point, validation errors result in either warnings or are rejected based on the active system-level PARTNER ERROR CONTROL global parameter. If the global parameter is not active, all errors are treated as warnings only. Otherwise, error processing follows the steps noted earlier in this section.

Gentran:Basic validates the sequence of each segment and its respective elements against the standards selected on the group envelope. Error/warnings is generated for the following conditions:

- Mandatory segment missing
- Mandatory element missing
- Conditional elements not meeting the proper relationship
- Wrong type of data for element
- Incorrect element size (min/max)
- Segments out of sequence
- Too many loops or occurrences of a segment
- Too many elements for a segment
- Invalid code value in element

For the CTT segment, Gentran:Basic validates the count and, if appropriate, the hash totals. Should an error be detected, an error/warning message would be generated on the error report. Also, ###REJ records can be generated, if set up on partner profile and partner error control is active.

## Trailer Envelope Processing (Standard Enveloping)

Trailer envelopes must have a Segment ID of SE, GE, IEA, ICE, EG, UNT, UNE, MTR, EOB, END, UNZ, or UNP. If missing a valid Segment ID, the Editor generates an error/warning report. However, data suspension does not occur for Trailer segment errors. If you want to perform error rejection (data suspension), you must specify this in the partner profile, along with activating partner error control.

**Note:** Previous ###REJs causes the count to be different.

## Control Record Processing (Deferred Enveloping)

For documents being processed in Deferred Envelope mode, global parameters that reference the partner are not used.

### *Interchange CNTL*

The interchange Control record, CNTL I, contains the Partner ID and Qualifier (Partner/Qualifier mode) or the User/Partner ID (Relationship mode), Header Option, and interchange version needed for building the interchange envelope.

### *Group CNTL*

The Group Control record, CNTL G, contains the Partner ID and Qualifier (Partner/Qualifier mode) or the User/Partner ID (Relationship mode), and Functional Group ID needed for building the group envelope.

### *Transaction CNTL*

The Transaction Control record, CNTL T, contains the Partner ID and Qualifier (Partner/Qualifier mode) or the User/Partner ID (Relationship mode), and Transaction Set ID needed to build the transaction envelope.

### *Object CNTL*

The Object Control record, CNTL O, is a place marker that identifies a Package Structure. The actual Object Header envelope follows the CNTL O control record and is treated as the first segment of a UNO transaction set.

## **Segment Processing (Deferred Enveloping)**

The version to use for compliance checking the transaction segments is supplied to the Outbound Editor through a ###MP2 control record created by the Outbound Mapper. Two control records (###MAP and ###MP2) are created by the Outbound Mapper and follow immediately after each Transaction (CNTL) Control record.

During compliance checking of the segments, validation errors either result in warnings or are rejected based on the active system-level PARTNER ERROR CONTROL global parameter. If this global parameter is not active, all errors are treated as warnings only. Otherwise, error processing follows the steps noted earlier in this section. In deferred envelope processing, error rejection impacts only the transaction, not the entire document.

The Editor validates the sequence of each segment and its respective elements against the standards indicated on the ###MP2 control record created by the Outbound Mapper. Errors/warnings are generated for the following conditions:

- Mandatory segment missing
- Mandatory element missing
- Conditional elements not meeting the proper relationship
- Wrong type of data for element
- Incorrect element size (min/max)
- Segments out of sequence
- Too many loops or occurrences of a segment
- Too many elements for a segment
- Invalid code value in element

For CTT segments, the Editor validates the count and, if appropriate, the hash totals. For any errors detected by the Editor, it generates an error/warning message report.

## Databank Processing

The databank consists of two parts: Directory (formerly called Audit) and Message Store.

### *Directory (Audit)*

The level of auditing performed by the Editor is determined by the following global parameters and by the partner profile. Each of these parameters are explained in more detail in Appendix A of this guide.

OUTBOUND EDI INTERCHANGE DIRECTORY  
OUTBOUND EDI INTERCHANGE NONE

OUTBOUND EDI GROUP DIRECTORY  
OUTBOUND EDI GROUP NONE

OUTBOUND EDI TRANSACTION DIRECTORY  
OUTBOUND EDI TRANSACTION NONE

**Note:** For each level of auditing desired, a global parameter is required. Activating the transaction level directory does not automatically activate the group level directory.

In conjunction with the global directory parameters, audit generation can be controlled through the partner profile. The partner profile level can be activated through the PARTNER DATABANK LEVEL global parameter. For those trading partners that you want to perform auditing, you must update the interchange/group/transaction partners through online partner maintenance, by specifying D in the EDI Databank Outbound field. If both the partner level and the global level directories are active, the Editor always looks at the lowest level of detail.

### *Message Store*

To activate the outbound message store feature of the Editor, the following system-level global parameter must be active.

#### **OUTBOUND EDI INTERCHANGE FULL**

For the active system-level OUTBOUND EDI INTERCHANGE FULL global parameter, the Editor copies the wrapped output documents to a message store file for later extraction. To extract a document, please refer to the outbound EDI Databank Extract (EDID205) program definition.

### *Deferred Enveloping*

Deferred enveloping requires activation of the following parameters:

- OUTBOUND EDI INTERCHANGE FULL
- OUTBOUND EDI GROUP DIRECTORY
- OUTBOUND EDI TRANSACTION DIRECTORY
- OUTPUT DATABANK ONLY

**Note:** These parameters are automatically activated by the Outbound Editor when deferred enveloping is detected by the Editor.

### *Databank Trading Partner Modes*

The partner profile and databank have two modes of key structure, Partner/Qualifier and Relationship. The trading mode selection is specified online, through the Configuration file. After selection, the mode applies to all partners and should not be mixed on the databank.

**Note:** The trading mode has no runtime overrides.

#### **Partner/Qualifier Mode**

When creating the directories for the outbound EDI databank, the trading mode to be used is extracted from Record Type 0 in the Configuration file. If the Partner/Qualifier mode is indicated, the sender Partner ID and Qualifier is the key. This is a resolved Partner ID and Qualifier which is the actual partner used by mapping or envelope generation to build the envelopes.

#### **Relationship Mode**

When creating the directories for the outbound EDI databank, the trading mode to be used is extracted from Record Type 0 in the Configuration file. If the Relationship mode is indicated, the partner Relationship file is used to obtain the User/Partner (relationship) partner profile key.

Refer to the earlier section on Partner Relationship Usage which explains the process of determining the partner profile User/Partner ID. The databank uses this partner profile User/Partner ID.

### **Viewpoint Processing**

If Gentran:Viewpoint is installed, information to perform Exception Management and Tracking Management for all Gentran:Basic modules are automatically passed to Viewpoint. No additional global parameters are required to request Viewpoint processing; all the required configuration information is included on the Viewpoint system files and the Gentran:Basic partner profile.

However, if a user wishes to track their data after it passes through the Editor (e.g., into their communications system) a user-written communications tracking program must be created. This user communications tracking program must call the appropriate Viewpoint User Gateway identifying the data file/communications batch with the Viewpoint Trace ID that is specified on the Viewpoint User Tracking record.

See the *Gentran:Viewpoint Tracking Management Guide* for detailed information on Viewpoint User Gateways.

### **Output Processing**

In deferred envelope processing, no output file is created by the Outbound Editor. Only the outbound EDI databank is updated.

#### *Directed Output Files*

For outbound processing, data separation can only occur at the partner level. Actual splitting of data takes place in the Split (EBDI016) program, but the ###DIR control record is created by the Editor. To initiate the Direction Split feature, the global parameter, Directed Output Files must be active in the global parameter file.

Example of a Direction record is as follows:

```
###000000101ISA ###DIRreceiver IDC 0001
###000004001ISA ###DIRreceiver IDC M00020016
```

### Direction Record

The following table describes the Direction record for both the Partner/Qualifier and Relationship modes.

Position	Value	Source of Value
001	###	Editor generated pass through ID
004	0000001	Record number (generated by Editor)
011	01	Sequence number (generated by Editor)
013	ISA	Segment ID from interchange record
016	###DIR	Outbound split constant
<b>Partner/Qualifier Mode</b>		
022	Receiver ID	From interchange record — BG use element BG04 ISA use element ISA08 ICS use element ICS08 GS use element GS03 UNB use element UNB07 STX use element STX05
057	(ID Qualifier)	BG use Space ISA use element ISA07 ICS use element UNB08 STX use element STX05
061	C	Constant value
062	b	Two Values possible <b>b</b> – 1 output file for split <b>M</b> – multiple files for split
063	0001	Up to 4 occurrences starting in position 63 of four characters each. Denotes the last two digits from the Split (EBDI016) program DD name (SYSxxx).



Position	Value	Source of Value
<b>Relationship Mode</b>		
022	User ID	From interchange record — BG use element BG04 ISA use element ISA08 ICS use element ICS08 GS use element GS03 UNB use element UNB07 STX use element STX05
037	Partner ID	BG use Space ISA use element ISA07 ICS use element UNB08 STX use element STX05
061	C	Constant value
062	b	Two Values possible b – 1 output file for split M – multiple files for split
063	0001	Up to 4 occurrences starting in position 63 of four characters each. Denotes the last two digits from the Split (EBDI016) program DD name (SYSxxx).

### *File Selection*

Determining whether a multiple or a single split occurs depends on the values found in the partner profile Data Separation record for the Receiver ID (Partner/Qualifier mode) indicated in position 22 or the User/Partner ID (Relationship mode) indicated in positions 22 and 37 of this record. If more than four splits are selected, the Editor generates multiple Direction records.

If no Data Separation record is found for this Partner ID the Editor checks the partner control for the Partner/Qualifier mode record (!!!GENTRAN-RESERVED-PARTNER-ID) or the Relationship mode record (!!!GENTRAN-RU1 for the User ID or !!!GENTRAN-RP1 for the Partner ID) to see if a default separation record has been set up. If not found, no Direction record is generated and this interchange progresses to the Slide file in the Split (EBDI016) program.

### *Data Compression of Output*

The Outbound EDI Databank Message Store file is the same as the documents written to the Output file. In deferred enveloping, there is no Output file. However, data compression is still needed for the Outbound EDI Databank Message Store.

Data compression in the Outbound Editor accomplishes two tasks. The first is the removal of all leading zeros and trailing spaces from elements in each segment. Elements are not compressed below their minimum length established by the standards used. For R-Type elements, the zeroes before and after a decimal point are considered significant and are not compressed. Second, the Editor wraps the segments to eliminate unnecessary spaces between records to again further reduce the size of your transmission.

Wrapping of segments is controlled through the global parameters. If no global parameter is active, the Editor defaults to COMPRESS TRANSMISSION.

**Compress Transmission**

Each interchange (i.e., ISA, ICS, BG, GS, UNA, UNB, SCH, STX) envelope begins on a new record in position one. Each consecutive segment within that interchange begins immediately following the terminator of the previous segment.

**Compress Group**

Each interchange (i.e., ISA, ICS, BG, GS, UNA, UNB, SCH, STX) envelope begins on a new record in position one.

Each group (i.e., GS, UNG, BAT) begins on a new record in position one.

Each consecutive segment within that group begins immediately following the terminator of the previous segment.

**Compress Transaction**

Each interchange (i.e., ISA, ICS, BG, GS, UNA, UNB, SCH, STX) envelope begins on a new record in position one.

Each group (i.e., GS, UNG, BAT) begins on a new record in position one.

Each transaction (i.e., ST, UNH, MHD) begins on a new record in position one.

Each consecutive segment within that transaction begins immediately following the terminator of the previous segment.

**Compress Segment**

Effect of this selection is no wrapping. Each segment begins on a new record in position one.

***Error Report Processing***

ERROR REPORT ALWAYS — Unless this system-level global parameter is active, Gentran:Basic creates an error report only if an error occurs. In any case, Gentran will always generate the totals.

***Detail Report Processing***

DETAIL REPORT ALWAYS — This Runtime global parameter produces an expanded error report containing detailed information, which is useful for problem resolution. Typically, production of the report is limited to problem resolution situations because of the large length of the report.

***Viewpoint Tracking of Output***

For the Gentran:Viewpoint Tracking Management product, you may wish to monitor your EDI data file(s) as the data passes through your communications system. To track the interchanges, functional groups, and transactions contained within the output file(s), the Editor creates Viewpoint User Tracking records for each output file produced. The global parameter, TRACK OUTPUT FILES, must be specified and Viewpoint Tracking Management must be installed.

Following is an example of a Viewpoint User Tracking record:

```
###VPTM BA00000001
```

Position	Value	Source of Value
001	###VPTM	Literal identifier generated by Editor.
009	BA00000001	Viewpoint Tracking Trace ID assigned by the Viewpoint Sterling Batch Gateway (EDIN100) program.

## Network Tracking

Gentran:Basic network tracking uses the partner profile and the outbound EDI databank. The Databank Interface (EDID202) program for the Outbound Editor creates a reconciliation file to be used in the Network Reconciliation (EDID850) process. After receiving your machine readable status information from the networks, you need to run the Network Response Processor (EDID800 for Commerce:Network, EDID805 for GEIS, and EDID810 for IBM) to create a common status format. Once in a common format, you need to reconcile your outbound EDI databank with the network status information received. Running the Network Reconciliation (EDID850) program using the reformatted status information does this. The following sections explain how to set up this feature.

### Partner Profile

The partner control record contains the flag for activating the Network Tracking feature. To activate the Network Tracking feature, enter **Y** (Yes) in the Network Tracking field on the partner Control Information (EDIM015) screen.

**Note:** Enter **N** (No), if you do not wish to activate the Network Tracking feature.

After updating the partner control record, the only other requirement for network tracking is to set up the interchange level outbound EDI directory. This can be accomplished either by using global parameters or through the partner profile.

### Global Parameters

#### Partner Databank Level

This system-level global parameter indicates to the Editor that the partner profile (Map record) is to be referenced by the Editor to determine the level of databank to create. To complete the requirements for network tracking, enter **D** in the EDI Databank Outbound field on the Control Information (EDIM015) screen.

#### Outbound EDI Interchange Full or Outbound EDI Interchange Directory

Gentran enables you to decide whether to use the partner profile or global parameters to establish your databank level. The Outbound EDI Interchange Full or Outbound EDI Interchange Directory system-level global parameters provide yet another alternative to using the partner profile. These global parameters apply to all of your trading partners and ensure a consistent directory level for your outbound processing.

## Processing

Once network tracking is set up, the Outbound Editor (interface) updates the network reconciliation file. After documents have been processed by the network, run the appropriate version of the Network Response Processor (EDID800 for Commerce:Network, EDID805 for GEIS, and EDID810 for IBM) to reformat the machine readable network status reports. Having reformatted the network status report file, you must run the Network Reconciliation (EDID850) program to update the databank. After the databank is updated, you can view the status of your documents online, through the Databank Maintenance system.

See the Network Response Processor (EDID800/EDID805/EDID810) and the Network Reconciliation (EDID850) program descriptions for additional information.

## Databank Network Extract

This option is an alternative to creating an output file with ####DIR records and using the data separation (Split – EBDI016) program. This option requires the message store feature of databanking and the partner profile.

## Global Parameters

### Directed Output File

Indicates to the Editor that the data separation feature of the Editor is to be used. The Editor goes to the Partner Data Separation record using the receiver Partner (interchange) ID and Qualifier to obtain the split files for the ####DIR record and the Network ID.

### Outbound EDI Interchange Full

Instructs the Editor to initialize the databank for both directory and message store processing.

### Output Databank Only

Indicates to the Editor that no output file is created in this run. Only the message store is updated. Editor verifies that the message store feature has been selected, otherwise this request is ignored.

## Partner Profile

When setting up the data separation record, you must enter the Network ID for this trading partner. If this field is not entered, the extract by network feature is not possible.

## Processing

With this optional set up, you can extract from the databank all partners going to Commerce:Network, IBM, or GEIS. Each network can be extracted separately for processing. This feature is an alternative to running the outbound data (Split – EBDI016) program. The databank Extract feature has several options available.

See the description for the Outbound EDI Databank Extract (EDID205) program.

## Files Used

### Input

The following files are used as input:

Filename	Description
SYS001	Tables File This file contains the details for each version of the ASC X12, TDCC, TRADACOMS, and EDIFACT.
ERRCTL	Error Message and Control File This file contains descriptions and control flags for every error number in the system.
EDISCD1	This file contains code values with lengths of 1 to 5 bytes. Standards Maintenance is used to update the file. This file is optional. See the CODE CHECK record for system-levels.
EDISCD2	This file contains code values with lengths of 6 to 10 bytes. Standards Maintenance is used to update the file. This file is optional. See the CODE CHECK record for global parameters.
EDISCD3	This file contains code value with lengths of 11 to 20 bytes. Standards Maintenance is used to update the file. This file is optional. See the CODE CHECK record for global parameters.
EDISCD4	This file contains code values with lengths of 21 to 40 bytes. Standards Maintenance is used to update the file. This file is optional. See the CODE CHECK record for global parameters.
SYS030	Global Parameter File This file contains all parameter records that are used by the Editors. These records apply to all partners.
SYS040	Input File (30000-character variable input) This file contains the data to be sent to your trading partners.
SYS090	Partner Profile This file is used to load specified information concerning the sender and receiver. The information used is at the interchange, group, and transaction level.

*Output*

The following files are used as output:

<b>Filename</b>	<b>Description</b>
SYS006	Summary Report This report contains summary information for the Editor. Information includes processing parameters and control counts.
SYS007	Summary Report in Data Format This file contains the summary report control counts in data format.
SYS010	Error Report This file contains information from the Editor for each error detected in processing.
SYS050	Output File (80-character Output) This file contains all non-suspended data that is processed by the Editor. If selected, this file also contains embedded Direction and Error Rejection control records.
SYS060	Suspense File (30000-character Variable Output) This file contains all data that did not have valid control records.

*Databank*

The following are databank files:

<b>Filename</b>	<b>Description</b>
EDINRC	Network Reconciliation File (Output) This file contains the data for reconciling reports returned from networks (e.g., Commerce:Network, IBM, GEIS).
EDIOES	Outbound EDI Databank Message Store This file contains EDI data in the same format and is sent to your trading partners. The EDI data on this file can be viewed and optionally edited using CICS Databank Maintenance.
EDIOAL	Outbound Application Databank Link File This file contains control information used to connect each application document on the Outbound Application Databank to the same transaction on the Outbound EDI Databank.
EDIOEA	Outbound EDI Databank Directory (Input and Output) This file contains the audit information for interchange, group, and transaction. If message store is selected, this file also contains the pointers for the message store data in EDIOES.
EDICFG	The Configuration file contains records to control system-level features and options.

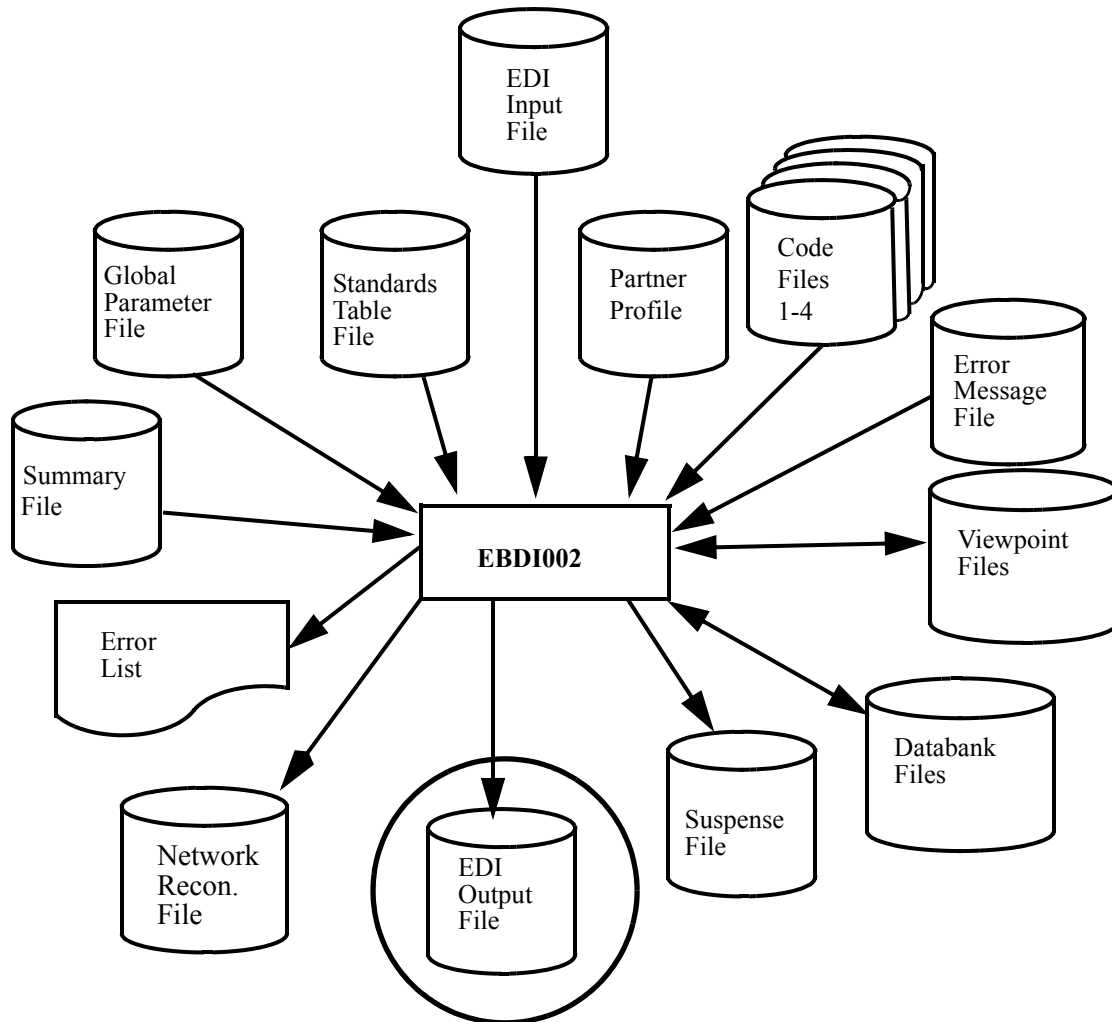
*Gentran:Viewpoint*

The following are Gentran:Viewpoint files:

<b>Filename</b>	<b>Description</b>
EDIVP	The partner profile contains the rules that are used to control the Gentran:Viewpoint system as defined in Extended System Management and Exception Management.
EDIVSF	The Staging file is used to pass data from the Viewpoint enabled Outbound Editor to Gentran:Viewpoint. Staging files contain error messages issued by the Outbound Editor as well as control/status information.
EDIVT	The Trace ID file is used by Gentran:Viewpoint Tracking Management to store unique identifiers for tracked items.

*Outbound Editor*

Figure 3.8 illustrates the Outbound Editor.



**Figure 3.8 EBDI002 — Outbound Editor**

**Note:** The Viewpoint files are used only if Gentran:Viewpoint is installed.



*Outbound Summary Report Data File*

The following table describes the Outbound Summary Report Data file.

<b>Field Content</b>	<b>COBOL Picture</b>	<b>Format</b>	<b>Length</b>	<b>Description</b>
Summary Start Date	9(08)	ZD	08	Start of Editor run. Year, Month, Day.
Summary Start Time	9(08)	ZD	08	Editor Start time. Hours, Minutes, Seconds.
Summary End Date	9(08)	ZD	08	End of Editor run. Year, Month, Day.
Summary End Time	9(08)	ZD	08	Editor end time. Hours, Minutes, Seconds.
Interchange In	9(15)	ZD	15	Total number of interchanges read.
Groups In	9(15)	ZD	15	Total number of groups read.
Transaction In	9(15)	ZD	15	Total number of transactions read.
Segments In	9(15)	ZD	15	Total number of segments read.
Records In	9(15)	ZD	15	Total number of records read.
Characters In	9(15)	ZD	15	Total number of characters read.
Characters Out	9(15)	ZD	15	Total number of characters written.
Characters Suspended	9(15)	ZD	15	Total number of characters suspended.
Interchanges Out	9(15)	ZD	15	Total number of interchanges written.
Interchanges Rejected	9(15)	ZD	15	Total number of interchanges rejected.
Interchanges Suspended	9(15)	ZD	15	Total number of interchanges suspended.
Groups Out	9(15)	ZD	15	Total number of groups written.
Groups Rejected	9(15)	ZD	15	Total number of groups rejected.
Groups Suspended	9(15)	ZD	15	Total number of groups rejected.
Transactions Out	9(15)	ZD	15	Total number of transactions written.
Transactions Rejected	9(15)	ZD	15	Total number of transactions rejected.
Transactions Suspended	9(15)	ZD	15	Total number of transactions suspended.
Segments Out	9(15)	ZD	15	Total number of segments written.
Segments Rejected	9(15)	ZD	15	Total number of segments rejected,
Segments Suspended	9(15)	ZD	15	Total number of segments suspended.
Databank Interchg Out	9(15)	ZD	15	Number of interchanges stored on databank.
Databank Interchg Rej	9(15)	ZD	15	Rejected interchanges stored on databank.
Databank Groups Out	9(15)	ZD	15	Number of groups stored on databank.

Field Content	COBOL Picture	Format	Length	Description
Databank Groups Rej	9(15)	ZD	15	Rejected groups stored on databank.
Databank Trans Out	9(15)	ZD	15	Number of transactions stored on databank.
Databank Trans Rej	9(15)	ZD	15	Rejected transactions stored on databank.
Databank Segments Out	9(15)	ZD	15	Number of segments stored on databank.
Databank Segments Rej	9(15)	ZD	15	Rejected segments stored on databank.
Databank Chars Out	9(15)	ZD	15	Number of characters stored on databank.
Databank Chars Rej	9(15)	ZD	15	Rejected characters stored on databank.
Databank Records Out	9(15)	ZD	15	Number of records stored on databank.
Databank Records Rej	9(15)	ZD	15	Rejected records stored on databank.
Filler	X(30)	AN	30	Not used.
Reject Pass-thru Records	9(15)	ZD	15	Error rejection pass-thru records written
Direction Records Written	9(15)	ZD	15	Direction pass-thru records written.
Pass-thru Records Written	9(15)	ZD	15	Total pass-thru records written.
Records Written	9(15)	ZD	15	Total records written.
Filler	X(30)	AN	30	Not used.
Packages Written	9(15)	ZD	15	Total number of package written
Filler	X(53)	AN	15	Not used.

**Note:** A COBOL copybook has been provided in the source utility product library. The member is SUMTWS. This copybook can be used by either Editor. Fields in the copybook can be shared or restricted to either Editor.

## Outbound Editor Reports

This section provides an explanation, as well as field descriptions, of the two reports produced by the Outbound Editor (EBDI002) program. The Error (SYS010) and Summary (SYS006) reports contain the following:

- *Error Report.* This report displays errors detected while processing outgoing data through the Outbound Editor. Interchange, group, and transaction separate the errors.
- *Summary Report.* This report provides a record of the processing options in effect for editing outgoing data through the Outbound Editor and the control counts. You will find these sections on the Summary Report:
  1. Runtime Global Parameter Overrides
  2. System Configuration Options
  3. Global Parameter List
  4. Global Parameter List – Version Conversion (optional)
  5. Global Parameter List – Implementation Convention (IC) Conversion (optional)
  6. Summary Control Counts Processing Outgoing Data

Figure 3.9 through Figure 3.11 provide samples of the Error and Summary reports.

Error Report

```

EBDI002  RUN 12/01/2005  TIME 9:52  ERRORS ENCOUNTERED PROCESSING OUTGOING DATA  PAGE 1
ERROR RECORD SEG ELT/COMP
NUMBER NUMBER ID NBR INFORMATION ERROR MESSAGE
INTERCHANGE = VENDOR-1 - TYPE = ISA ENV = 000000036 VER = 00300 AGY = X
TUTORIAL - XYZ COMPUTER COMPAN
EDI SENDER STERLING
EDI RECEIVER XYZ
GROUP = VENDOR-1 - TYPE = PO ENV = 000000034 VER = 003030 AGY = X
TRANSACTION = PONUMBER-001 TEST TYPE = 850 ENV = 000000135 VER = 003030 AGY = X
689 47 SE VENDOR-1 DUPLICATE DOCUMENT ADDED TO DATA BANK.
*** DOCUMENT AUDIT ID AUDIT ID = PONUMBER-001
689 48 GE VENDOR-1 DUPLICATE DOCUMENT ADDED TO DATA BANK.
689 49 IEA VENDOR-1 DUPLICATE DOCUMENT ADDED TO DATA BANK.
INTERCHANGE = VENDOR-2 - TYPE = ICS ENV = 000000016 VER = 00300 AGY = X
TUTORIAL - BULK PAPER COMPANY
EDI SENDER GSSENDER
EDI RECEIVER GSRECVER
GROUP = VENDOR-2 - TYPE = PO ENV = 000000015 VER = 003030 AGY = X
TRANSACTION = PONUMBER-002 TEST TYPE = 850 ENV = 000000023 VER = 003030 AGY = X
689 86 SE VENDOR-2 DUPLICATE DOCUMENT ADDED TO DATA BANK.
*** DOCUMENT AUDIT ID AUDIT ID = PONUMBER-002
689 87 GE VENDOR-2 DUPLICATE DOCUMENT ADDED TO DATA BANK.
689 88 ICE VENDOR-2 DUPLICATE DOCUMENT ADDED TO DATA BANK.
INTERCHANGE = TUTORIAL - TWO WAY COMMUNICATI - TYPE = GS ENV = 003030 AGY = X
EDI SENDER STERLING
    
```

Figure 3.9 Outbound Editor Error Report  
(Errors Encountered Processing Outgoing Data)

Summary Report

EBDI002	RUN 12/01/2005	TIME 11:08	RUNTIME GLOBAL PARAMETER OVERRIDES	PAGE 1
DETAIL REPORT ALWAYS USE SUBELEMENT SEPARATOR FOR ANSI VERSION 00300 X 00200 X				
EBDI002	RUN 12/01/2005	TIME 11:08	SYSTEM CONFIGURATION OPTIONS	PAGE 2
INTCHG VERSION = N GROUP VERSION = Y TRANSACTION VERSION = Y TRADING PROFILE MODE PARTNER QUALIFIER MULTIPLE ENVELOPE ENABLED CONCURRENCY DISABLED				
EBDI002	RUN 12/01/2005	TIME 11:08	GLOBAL PARAMETER LIST	PAGE 3
VERIFY PARTNER INTERCHANGE OFF VERIFY PARTNER GROUP OFF VERIFY PARTNER TRANSACTION OFF ERROR REPORT ALWAYS DETAIL REPORT ALWAYS ICS TABLES OFF CODE CHECK ON PARTNER DATABANK OFF COMPRESS TRANSMISSION GS SENDER/RECEIVER QUALIFIER INTERCHANGE USE SUBELEMENT SEPARATOR FOR ANSI OUTBOUND EDI INTERCHANGE FULL OUTBOUND EDI GROUP DIRECTORY OUTBOUND EDI TRANSACTION DIRECTORY				
EBDI002	RUN 12/01/2005	TIME 11:08	GLOBAL PARAMETER LIST - VERSION CONVERSION	PAGE 4
VERSION 00300 X 00200 X				

Figure 3.10 Outbound Editor Summary Report

```

EBDI002    RUN 12/01/2005 TIME 11:08    GLOBAL PARAMETER LIST - IC CONVERSION    PAGE 5
IC    INPUT IDENTIFIER    00200    X
SUMMARY CONTROL COUNTS PROCESSING OUTGOING DATA    PAGE 6
EBDI002    RUN 12/01/2005 TIME 11:08
PROCESSING BEGAN ON 12/01/2005 AT 11:08 AM.
OVERALL INPUTS AND OUTPUTS SUMMARY
INTERCHANGES READ ----- 4
GROUPS READ ----- 4
TRANSACTIONS READ ----- 4
SEGMENTS READ ----- 160
RECORDS READ ----- 172
CHARACTERS READ ----- 6,483
INTERCHANGES WRITTEN ----- 4
INTERCHANGES REJECTED ----- 0
INTERCHANGES SUSPENDED ----- 0
GROUPS WRITTEN ----- 4
GROUPS SUSPENDED ----- 0
TRANSACTIONS WRITTEN ----- 4
TRANSACTIONS REJECTED ----- 0
TRANSACTIONS SUSPENDED ----- 0
SEGMENTS WRITTEN ----- 160
CHARACTERS WRITTEN (WRAPPED OUTPUT) ----- 6,252
SEGMENTS SUSPENDED ----- 0
MISCELLANEOUS OUTPUT SUMMARY
TOTAL PASS-THRU WRITTEN ----- 0
DIRECTION PASS-THRU WRITTEN ----- 0
ERROR REJECTION PASS-THRU WRITTEN ----- 0
TOTAL RECORDS WRITTEN ----- 80
TOTAL PACKAGES WRITTEN ----- 0
PROCESSING ENDED ON 12/01/2005 AT 11:08 AM.

-----
INTERCHANGES STORED ON DATA BANK ----- 4
REJECTED INTERCHANGES ON DATA BANK ----- 0
GROUPS STORED ON DATA BANK ----- 4
REJECTED GROUPS ON DATA BANK ----- 0
TRANSACTIONS STORED ON DATA BANK ----- 4
REJECTED TRANSACTIONS ON DATA BANK ----- 0
SEGMENTS STORED ON DATA BANK ----- 160
CHARACTERS STORED ON DATA BANK ----- 6,400
RECORDS STORED ON DATA BANK ----- 80
-----

-----
INTERCHANGES STORED ON DATA BANK ----- 4
REJECTED INTERCHANGES ON DATA BANK ----- 0
GROUPS STORED ON DATA BANK ----- 4
REJECTED GROUPS ON DATA BANK ----- 0
TRANSACTIONS STORED ON DATA BANK ----- 4
REJECTED TRANSACTIONS ON DATA BANK ----- 0
SEGMENTS STORED ON DATA BANK ----- 160
CHARACTERS STORED ON DATA BANK ----- 6,400
RECORDS STORED ON DATA BANK ----- 80
-----

-----
INTERCHANGES STORED ON DATA BANK ----- 4
REJECTED INTERCHANGES ON DATA BANK ----- 0
GROUPS STORED ON DATA BANK ----- 4
REJECTED GROUPS ON DATA BANK ----- 0
TRANSACTIONS STORED ON DATA BANK ----- 4
REJECTED TRANSACTIONS ON DATA BANK ----- 0
SEGMENTS STORED ON DATA BANK ----- 160
CHARACTERS STORED ON DATA BANK ----- 6,400
RECORDS STORED ON DATA BANK ----- 80
-----

```

Figure 3.11 Continue – Outbound Editor Summary Report

## *Field Definitions*

### **Errors Encountered Processing Outgoing Data**

Error Numbr:

This field displays the error code number associated with the error message.

Record Number:

This field displays the application record number.

Seg ID:

This field displays the EDI segment identifier.

Elt/Comp Nbr:

This field displays the EDI element/component number.

Information:

This field displays descriptive information identifying additional information about the error message.

Error Message:

This field displays descriptive information identifying sequence information.

### **Runtime Global Parameter Overrides**

This section of the report displays Runtime global parameter overrides for this execution of the Outbound Editor, specified via Global Parameter file (SYS030) in the JCL. If you did not specify Runtime global parameters, this section will be blank.

### **System Configuration Options**

This section of the report displays the settings from the Configuration (EDICFG) file used by the Outbound Editor.

### **Global Parameter List**

This section of the report displays all the parameter options in effect for editing outgoing data. Only those parameter options used are displayed.

### **Global Parameter List – Version Conversion (optional)**

This section of the report displays specified version conversion parameters for this execution of the Outbound Editor. If you did not specify version conversion parameters, the Editor will not produce this section.

### **Global Parameter List – Implementation Convention (IC) Conversion (optional)**

This section of the report displays specified IC conversion parameters for this execution of the Outbound Editor. If you did not specify IC conversion parameters, the Editor will not produce this section.

**Summary Control Counts Processing Outgoing Data**

(Message):

This field displays a text message indicating the date (in MM/DD/YYYY format) and the time (in HH:MM format) that processing began.

**Overall Inputs and Outputs Summary**

Interchanges Read:

This field displays the total number of interchanges read. Interchange control segments (i.e., ISA, BG, ICS, GS, UNA/UNB, SCH/STX) increment this total.

Groups Read:

This field displays the total number of groups read. Group control segments (i.e., GS, UNG, and BAT) increment this total.

Transactions Read:

This field displays the total number of transactions read. Transaction control segments (i.e., ST, UNH, MHD) increment this total.

Segments Read:

This field displays the total number of segments read. This field is based on number of segment terminators within the EDI data for each segment including interchange Headers.

Records Read:

This field displays the total number of records read. This field is a 255-byte outbound data segment physical record count. Each data segment is terminated.

Characters Read:

This field displays the total number of characters read. Providing that all interchange Headers have terminators, this count details all characters between first segment character and terminator.

Interchanges Written:

This field displays the total number of interchanges written. Interchange control Trailer segments (i.e., IEA, EG, ICE, GE, UNZ, END) increment this total, which includes sequential and databank output.

Interchanges Rejected:

This field displays the total number of interchanges rejected. Interchange control Trailer segments (i.e., IEA, EG, ICE, GE, UNZ, END) increment this total based on partner rejection.

Interchanges Suspended:

This field displays the total number of interchanges suspended. This field represents the occurrence of each interchange suspension as detected by the Outbound Editor.

Groups Written:

This field displays the total number of groups written. Group control Trailer segments (i.e., GE, UNE, EOB) increment this total, which includes both sequential and databank output.



**Groups Rejected:**

This field displays the total number of groups rejected. Group control Trailer segments (i.e., GE, UNE, EOB) increment this total based on group rejection.

**Groups Suspended:**

This field displays the total number of groups suspended. This field represents the occurrence of each group suspension as detected by the Outbound Editor.

**Transactions Written:**

This field displays the total number of transactions written. Transaction control Trailer segments (i.e., SE, UNT, MTR) increment this total, which includes both sequential and databank output.

**Transactions Rejected:**

This field displays the total number of rejected transactions. Transaction control Trailer segments (i.e., SE, UNT, MTR) increment this total based on transaction rejection.

**Transactions Suspended:**

This field displays the total number of transactions suspended. This field represents the occurrence of each transaction suspension as detected by the Outbound Editor.

**Segments Written:**

This field displays the total number of segments written on the expanded sequential output. This field is based on the number of segment terminators within the EDI data for each segment including interchange Header.

**Characters Written (Wrapped Output):**

This field displays actual characters written to wrapped output file. Direction markers, error rejection markers, and pass-thru records are not included in this field. Actual characters, numbers, and delimiters are included.

**Segments Suspended:**

This field displays EDI data segment suspended with segment terminators.

**Note:** Original envelopes are copied to suspense at the group and transaction level.

**Characters Suspended:**

This field displays characters written to suspension file. This does not include spaces between EDI data segments.

**Miscellaneous Output Summary****Total Pass-Thru Written:**

This field displays the count of any records received with the pass-thru indicator '###' starting at column 1 of input record.

**Direction Pass-Thru Written:**

This field displays the number of direction record markers within the wrapped output file. Global parameter DIRECTED OUTPUT FILES controls the use of direction record markers within the wrapped output file.

**Error Rejection Pass-Thru Written:**

This field displays the number of error rejection markers within the wrapped output file. Global parameter PARTNER ERROR CONTROL controls the use of error rejection markers within the wrapped output file.

**Total Records Written:**

This field displays the total records within wrapped output file. Wrapped segments, error records, direction markers, pass-thru records, and error rejection markers are included.

**Total Packages Written:**

This field displays the total number of EDIFACT packages within the wrapped output files.

**Databank Output Summary****Interchanges Stored on Data Bank:**

This field displays the total number of interchanges stored on the databank. Interchange control Trailer segments (i.e., IEA, EG, ICE, GE, UNZ, END) increment this total. This field represents the actual number of interchange directory records written to the outbound directory file.

**Rejected Interchanges on Data Bank:**

This field displays the total number of rejected interchanges on the data bank. Interchange control Trailer segments (i.e., IEA, EG, ICE, GE, UNZ, END) increment this total based on partner error rejection.

**Groups Stored on Data Bank:**

This field displays the total number of groups on the databank. Group control Trailer segments (i.e., GE, UNE, EOB) increment this total. This field represents the actual number of group directory records written to the outbound directory file.

**Rejected Groups on Data Bank:**

This field displays the total number of rejected groups on the databank. Group control Trailer segments (i.e., GE, UNE, EOB) increment this total based on group rejection.

**Transactions Stored on Data Bank:**

This field displays the total number of transactions stored on the data bank. Transaction control Trailer segments (i.e., SE, UNT, MTR) increment this total. This field represents the actual number of transaction directory records written to the outbound directory file.

**Rejected Transactions on Data Bank:**

This field displays the total number of rejected transactions on the databank. Transaction control Trailer segments (i.e., SE, UNT, MTR) increment this total based on transaction rejection.

**Segments Stored on Data Bank:**

This field displays the total number of segments stored on the databank.

**Characters Stored on Data Bank:**

This field displays the number of compressed characters written. Actual number is Records Stored on Databank multiplied by 80.

**Records Stored on DataBank:**

This field displays the total number of records stored on the databank. The records are 80-byte wrapped from the Editor and logically placed within the message store file.

**(Message):**

This field displays a text message indicating the date (in MM/DD/YYYY format) and the time (in HH:MM format) that processing ended.

---

## Outbound Split

**EBDI016**

The Outbound Split program in Gentran:Basic is used to separate data into distinct disk files.

Documents can be separated for transmitting to third party networks — like Commerce:Network — or to individual trading partners. You can split up to 899 separate files (Partners or Networks).

The Outbound Split program does not generate Direction (####DIR) or Rejection (####REJ) records. These records are generated by the Outbound Editors using both the global parameters and partner profiles.

See the Outbound Editor (EBDI002) program topic on page 3-77 for more details on creating Rejection and Direction records.

Along with the Direction and Rejection Control records, the Write Direction record is one parameter available for controlling the Split program. This parameter is only effective if the input has Direction records. Otherwise, it is ignored by the Split program.

### Outbound Split Processing

Processing begins with the reading of the control card file that contains overrides to be used during processing.

#### Write Direction Records

This parameter instructs the Split program to include the Direction record (####DIR) with the respective transactions. The normal process would delete the Direction records after the transaction records have been written.

OPEN FILE SYSxxx

Where xxx stands for files 001-099 and 200-999.  
Also, SYS100 for Gentran:Plus users.

This parameter is used by the Split program to identify which of the 899 different split files you are assigning. This prevents the time needed to open, close, and define all 899 files. If ####DIR record indicates a file not set up with an open parameter, the Split program opens and closes all split files indicated by Direction records.

After the parameters (if any) are stored, the transaction processing begins. The Split program looks for the Direction (####DIR) or Rejection (####REJ) records to guide its processing.

#### Direction Records (####DIR)

The Direction control record tells the Split program to copy the transactions to specific output files. Each Direction record contains up to four numbers that correspond to the last three digits of the output split files. The split files have a DDNAME of SYSxxx where the xxx can be a value of 001 to 099 and 200 to 999. These numbers instruct the Split program to write the transactions to their proper output files. A copy of the transactions is written to each file indicated on the Direction record. Should any other value be found, the Split program generates an error report and writes the transactions to the default Slide file.

**Rejection Records (###REJ)**

The Rejection control record tells the Split program that all transactions from the start of Rejection (###REJS) until it finds an end of reject marker (###REJE) are to be written to the reject file.

If no control records are found, the Split program writes the transactions to the default slide file.

**Viewpoint User Tracking Records (###VPTM)**

If Gentran:Viewpoint Tracking Management is installed, the outbound Split program writes a Viewpoint User Tracking record to each output file if the user communications tracking has been specified in the preceding Outbound Editor job. One Viewpoint User Tracking record is written as the first record in each split file created.

**Files Used**

The following table describes the files used by the EBDI016 program.

Filename	Description
SYSSIT	Sorted Input Transactions File This file contains sorted input transactions from file SYS160.
SYS000	Slide File (80-character output) This is the default file for the Split program. It is used when no Direction records are found.
SYS001 through SYS099 and SYS200 through SYS999.	Output Direction Files (80-character output) These files are used by the Split program when Direction records are present. The three digit codes on the Direction records correspond to the last three digits of the SYSxxx DDNAME.
SYS160	Input File (80-character input) This file contains the edited data from your trading partner with embedded Direction records and reject records, if requested.
SYS162	Work File This file is used to sort the input transactions from SYS160.
SYS163	Parameter File (80-character input) This file contains input parameters to the Split process. The valid parameters are: WRITE DIRECTION RECORDS Indicates that all Direction records are to be copied with their data to the split files. OPEN FILE SYSxxx This parameter specifies that the indicated split file is opened every time the Split program is run.
ERRCTL	Error Message and Control File The Error Message file contains the descriptions for error numbers generated during processing.

---

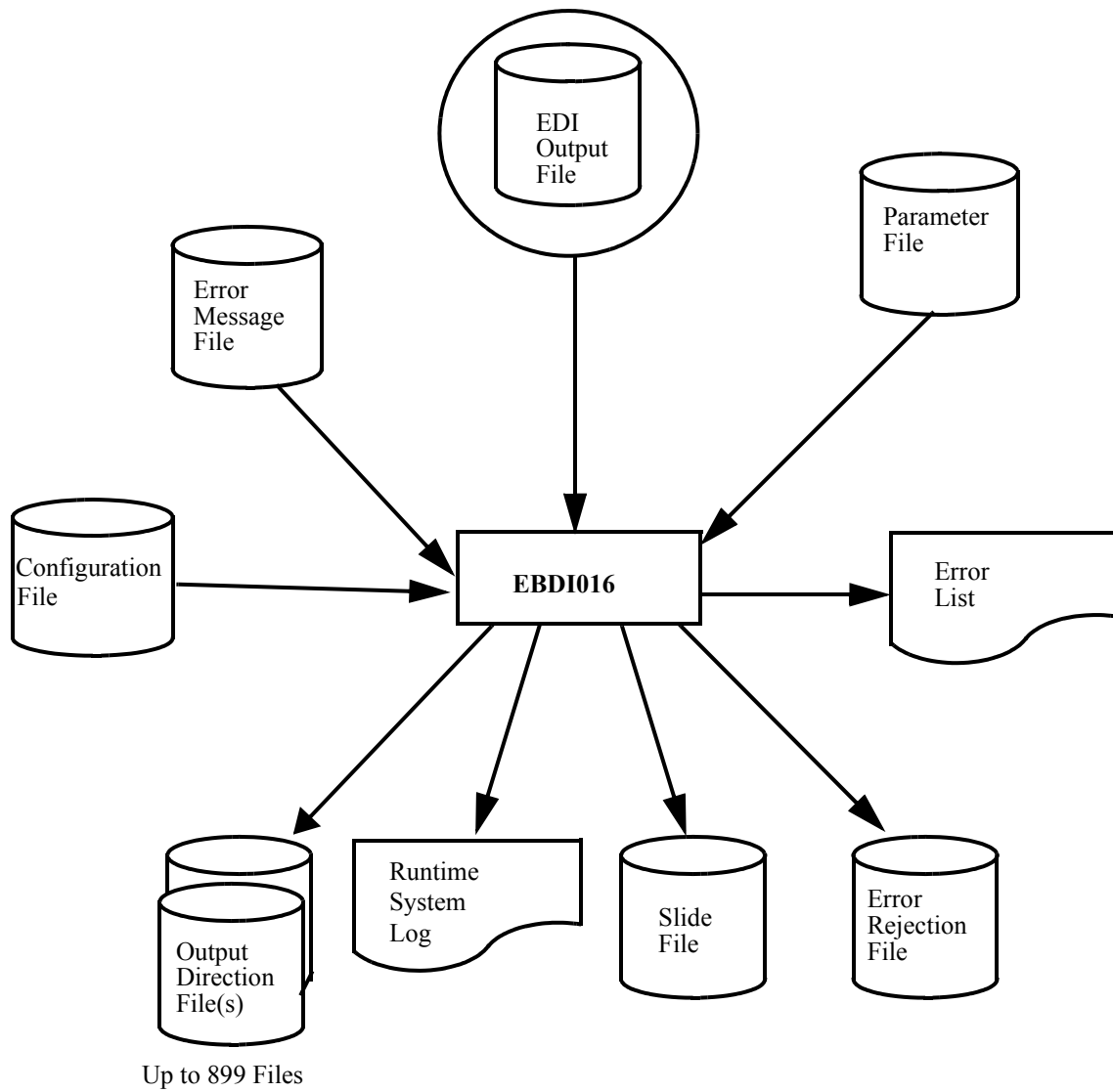
Filename	Description
SYS166	Error Report This report contains any errors detected during processing.
SYS167	Reject File (80-character output) This file contains all data marked with reject records from the Editors.
EDICFG	Configuration File The Configuration file contains system definitions.
SYS100	Output File Gentran:Plus Only (80-character output) This file contains all transmission data for trading partners designated as Gentran:Plus partners.

### Error Report

An error report is generated when the Split program detects a problem with the control records. It contains the sequence number of the record in error along with an error number and description. If an error is detected, the program generates an error message and terminates.

**Outbound Split Process**

Figure 3.12 illustrates the Outbound Split process.



**Figure 3.12 EBDI016 — Outbound Split Process**

## Outbound Split Processing Parameters

### *WRITE DIRECTION RECORDS Parameter*

The following table describes the WRITE DIRECTION RECORDS parameter.

Positions	Field Content	COBOL Picture	Format	Length	Description
1 – 23	Write Direction Records	X(23)	AN	23	Indicates Direction marker records for input file are to be written to output file. Otherwise it is dropped. Valid Value: WRITE DIRECTION RECORDS.
24 – 80	Filler	X(57)	AN	57	Not used.

### *OPEN FILE SYSxxx Parameter*

The following table describes the OPEN FILE SYSxxx parameter.

Positions	Field Content	COBOL Picture	Format	Length	Description
1 – 16	Open File SYSxxx	X(16)	AN	16	Indicates that the split file is opened every time the Split program is run. 'xxx' stands for split file 001-099 and 200-999.
17 – 80	Filler	X(63)	AN	63	Not used.

### *Gentran:Realtime® RESEQUENCE DIR Parameter*

The following table describes the Gentran:Realtime RESEQUENCE DIR parameter.

Positions	Field Content	COBOL Picture	Format	Length	Description
1 – 14	Resequence Directory records	X(14)	AN	14	Used to resequence direction records generated by multiple runs of Gentran:Realtime. Valid value: RESEQUENCE DIR.

## Runtime System Log List

Figure 3.13 is an example of the Runtime System Log List.

```
*****
PROGRAM EBDI016 COMPILED 01/27/0119.27.26
VERSION 6.4 GENTRAN: BASIC 12/01/2005
CURRENT DATE IS 12/01/2005
TIME STARTED IS 13:41:06
*****
PARMLIST:
SORT FIELDS=(0001,0010,CH,A)
RECORD TYPE=F,LENGTH=(000142,,)
*****
RECORDS READ = 00000529
DIRECTION RECORDS READ = 00000015
RECORDS WRITTEN = 00000529
REJECT RECORDS WRITTEN = 00000000
NON-SPLIT RECORDS WRITTEN = 00000000
RECORDS WRITTEN TO SPLIT FILE 0001 = 00000058
RECORDS WRITTEN TO SPLIT FILE 0002 = 00000341
RECORDS WRITTEN TO SPLIT FILE 0003 = 00000058
RECORDS WRITTEN TO SPLIT FILE 0007 = 00000026
RECORDS WRITTEN TO SPLIT FILE 0014 = 00000148
RECORDS WRITTEN TO SPLIT FILE 0025 = 00000014
RECORDS WRITTEN TO SPLIT FILE 0090 = 00000283
*****NORMAL END OF JOB*****
```

**Figure 3.13 Runtime System Log List**

## Miscellaneous Output Summary

### Records Read

This field displays the total number of input records read.

### Direction Records Read

This field displays the total number of direction markers (###DIR) read. Direction markers determine split file processing.

**Note:** Direction Records Read count is included in Records Read.

### Records Written

This field displays the number of records written at least once to the split and non-split output files. If the parameter WRITE DIRECTION RECORDS is active, this field includes counts for direction markers.

### Reject Records Written

Total number of rejected records. This count includes ###REJ start and end control records.

### Non-Split Records Written

Total number of records written with no splitting (no direction markers from input).

### Records Written to Split File xxxx

Total number of records written to a specific split file indicated by 'xxxx' value.



# Utility Programs

## Overview

This chapter contains the following topics related to the Gentran:Basic utility programs:

Topic	Page
Application Change Audit Maintenance Program .....	EBDI066 ..... 4-3
Application Copy from One Set of VSAM Files to Another .....	EBDI060 ..... 4-7
Application Definition Maintenance .....	EDIG100 ..... 4-12
Application Sequence Renumbering .....	EBDI090 ..... 4-18
Batch Separator.....	EBDI930 ..... 4-27
COBOL Copybook Generator.....	EDIG110 ..... 4-30
Code Table Change Audit Maintenance.....	EBDI068 ..... 4-33
Code Table Copy From File to File.....	EBDI076 ..... 4-36
Configuration Change Audit Maintenance Program.....	EBDI102 ..... 4-45
Data Router.....	EBDI095 ..... 4-49
Data Router Parameters .....	EBDI095 ..... 4-52
Data Routing .....	EBDI095 ..... 4-58
Defining the Rules for the Split .....	4-60
Error Message Change Audit Maintenance Program.....	EBDI105 ..... 4-64
Error Message Maintenance .....	EBDIT13..... 4-68
Global Parameter Change Audit Maintenance Program .....	EBDI103 ..... 4-71
JCL File Maintenance.....	EDIRJCLX ..... 4-75
Mapping Version Comparison and Conversion Report.....	EBDI088 ..... 4-80
Object Insert Program.....	EBDI065 ..... 4-97
Online Log File Utility .....	EDIELMT..... 4-102
Partner Change Audit Maintenance Program.....	EBDI093 ..... 4-105
Partners Copy From One File to Another.....	EBDI087 ..... 4-110
Security Change Audit Maintenance Program .....	EBDI100 ..... 4-115
Separator Change Audit Maintenance Program .....	EBDI104 ..... 4-118
Sort Generated Acknowledgments Program .....	EBDI110 ..... 4-122
Standard Change Audit Maintenance Program .....	EBDI033 ..... 4-123
Standards Copy Utility .....	EBDI032 ..... 4-126
Standards Selective Download .....	EBDI030 ..... 4-131

Standards Update .....	EBDI038, EBDI039...	4-135
Transaction Change Audit Maintenance Program.....	EBDI067 .....	4-142
Transaction Copy From One Set of VSAM Files to Another .....	EBDI080 .....	4-146
Transaction Mapping Definition Re-synch .....	EBDI089 .....	4-151

---

## Application Change Audit Maintenance Program

**EBDI066**

The Application Change Audit Maintenance program (EBDI066) maintains the Application Change Audit file. The program functions include deleting, archiving, and restoring change audit information.

The Application change audit file contains records that track changes made to the Application files. Add, delete, and update functions via the online Application subsystem result in records being written to the Application Change Audit file. The Batch copy utility (EBDI060), Version Comparison and Conversion (EBDI088), and Batch Application Re-sequence (EBDI090) also write records to the Application change audit file indicating add, delete, and replace functions. The online Upload process (EDIX235), using its component EDIR060, will also write a change audit record to indicate an add or replace event.

### Operation

The Application Change Audit Maintenance program operates in the following general modes:

#### Periodic Housekeeping

On a periodic basis (anywhere from daily to monthly depending on volume and retention requirements) the Housekeeping operation deletes and optionally archives data based on specified retention periods.

#### Ad Hoc Maintenance (Purge and Restore)

In addition to the regularly scheduled housekeeping runs, the Application Change Audit Maintenance's Purge and Restore functions deletes and optionally archives selected data and restores data from archives.

### Processing

The Application Change Audit Maintenance program processes documents using the following three operations:

- Housekeeping
- Purge
- Restore

#### *Housekeeping Operation*

If the Housekeeping operation is requested, Application Change Audit Maintenance deletes and optionally archives the change audit records that exceed the retention periods specified in the control cards. An option of the Housekeeping operation is to archive all selected change audit records to a sequential output file.

#### *Purge Operation*

If the Purge operation is requested, change records are removed if they match *all* criteria specified. An option of the Purge operation is to archive selected change audit records to a sequential output file.

### Restore Operation

If the Restore operation is requested, change audit records are restored from the Archive file to the Change Audit file if the records match *all* selection criteria specified.

### Files Used

#### Input

Control cards and the Application Change Audit files are the principal input to Application Change Audit Maintenance.

The following files are used as input:

DDname	Filename
APPARCH	Application Change Audit Archive File
EDIACHA	6.4 VSAM Application Change Audit File
EDICNTL	Control Card File
ERRCTL	Error Message and Control File
EDICFG	System Configuration File

#### Output

The following files are used as output:

DDname	Filename
APPARCH	Application Change Audit Archive File
EDIACHA	6.4 VSAM Application Change Audit File

### Reports

Program EBDI066 produces the following reports:

DDname	Filename
APPAUDT	Application Change Audit Maintenance Report
EDILOG	Application Change Audit Processing Log
EDISUM	Application Change Audit Summary Report

### Control Card Options

#### Operations

HOUSEKEEPING (Default Operation)  
PURGE  
RESTORE

**Selection Criteria****Defaults**

For Housekeeping operation:

RETENTION-DAYS

For Purge and Restore operations:

FROM-UPDATE-DATE	01/01/1900
FROM-UPDATE--TIME	00:00
TO-UPDATE-APPLIED-DATE	12/31/2099
TO-UPDATE-APPLIED-TIME	23:59
APPLICATION-ID	ALL
SEND-RECEIVE	ALL

For Purge and Housekeeping operations:

ARCHIVE	NO
---------	----

**Valid Criteria Values****Application ID**

A 10-character identification used to restrict maintenance by Application ID.

ALL  
XXXXXXXXXX

**Archive**

This controls whether an archive is to be created for deleted change records.

NO  
YES

**From-Update--Date**

Specifies the date that the update was applied to the Application record. Starts the range.  
Used with To-Update-Date.

MM/DD/YYYY

**From-Update-Time**

Specifies the time that the update was applied to the Application record. Starts the range.  
Used with To-Update-Time.

HH:MM

**Retention-Days**

Specifies the number of days that have elapsed since the update was applied.

000 - 999

**Send-Receive**

Specifies the direction of the application. Used with Application-Id to restrict the records available for maintenance.

S	Send
R	Receive
ALL	Send and Receive

**To-Update--Date**

Specifies the date that the update was applied to the Code Table. Ends the range. Used with From-Update-Date.

MM/DD/YYYY

**To-Update-Time**

Specifies the time that the update was applied to the Code Table. Ends the range. Used with From-Update-Time.

HH:MM

## Application Copy from One Set of VSAM Files to Another EBDI060

### Purpose

The Application Copy (EBDI060) program is one of the five Test/Production Integration Copy programs. It is used to copy an application definition from a test environment to the production environment and to delete application definitions from the target files.

The application definition may be directly copied (i.e., VSAM to VSAM) or to a sequential unload file for later upload.

The application definition must exist on the Source Header file (unless the DELETE function is specified).

If you have Change Audit for Application enabled on the "1" record of the EDICFG file, a new record will be written to the Application Change Audit file (EDIACHA) for each application that is added, deleted, or replaced.

### Files Used

#### Input

The following files are used as input:

Filename	Description
SYS001	Parameter file This file contains records specifying the applications to be copied from the source files, and the application to be created on the target files. Each record contains 'To' and 'From' fields to allow the application to be renamed on the new file. For the Delete function, the record identifies the application record to be deleted. The Unload function can not be intermixed with the Copy and Delete functions
SYS020	Source Application Header file The Header records from which the application is copied.
SYS021	Source Application Record file The Application records from which the application is copied.
SYS211	Source Application Record Path file This file is a secondary allocation for the Source Application Record file.
SYS022	Source Application Field file The application fields from which the application is copied.
SYS221	Source Application Field Path file This file is a secondary allocation for the Source Application Field file.
EDICFG	System Configuration file <b>Note:</b> If using the Change Audit feature, make sure the DD statement in your EXEC060 JCL for this file contains the dataset name as it appears in your target CICS region.

*Output*

The following files are used as output:

Filename	Description
SYS030	Target Application Header file The header records to which the application is copied, or from which it is deleted.
SYS031	Target Application Record file The application records to which the application is copied, or from which it is deleted.
SYS311	Target Application Record Path file This file is a secondary allocation for the Target Application Record file.
SYS032	Target Application Field file The application fields to which the application is copied, or from which it is deleted.
SYS321	Target Application Field Path file This file is a secondary allocation for the Target Application Field file.
SYS060	Sequential Unload file This file is used during the Unload processing. It contains the records that will be used by the CICS Upload process to update VSAM application files.
EDIACHA	Application Change Audit file <b>Note:</b> If using the Change Audit feature, make sure the DD statement in your EXEC060 JCL for this file contains the dataset name as it appears in your target CICS region.

*Parameter File Layout (Copy Function)*

Position	Field Content	COBOL Picture	Format	Length	Description
001 – 005	FROMb	X(05)	AN	05	Constant value of FROMb.
006 – 015	Copy from Application ID	X(10)	AN	10	Copy from Application ID of application to be copied.
016 – 016	Filler	X(01)	AN	01	Not used
017 – 017	Copy from Send/Receive	X(01)		01	Copy from Send/Receive of application to be copied. Valid values: S – Send R – Receive.
018 – 024	bbbbTOb	X(07)	AN	07	Constant value of bbbbTOb.



Position	Field Content	COBOL Picture	Format	Length	Description
025 – 034	Copy to Application ID	X(10)	AN	10	Copy to Application ID of application to be copied.
035 – 035	Filler	X(01)	AN	01	Not used
036 – 036	Copy to Send/Receive	X(01)	AN	01	Copy to Send/Receive of application to be copied. Valid values: <b>S</b> – Send <b>R</b> – Receive.
037 – 039	Blank	X(03)	AN	03	
040 – 040	Replace Switch (Y/N)	X(01)	AN	01	<b>Y</b> – Deletes the existing application from the Copy To files <b>N</b> – Process data, but system indicates the applications that need to be deleted. <b>Note:</b> The system does not copy if application exists on Copy To file and the replace switch is <b>N</b> .
041 – 080	Blank	X(40)	AN	40	Not used

*Parameter File Layout (Delete Function)*

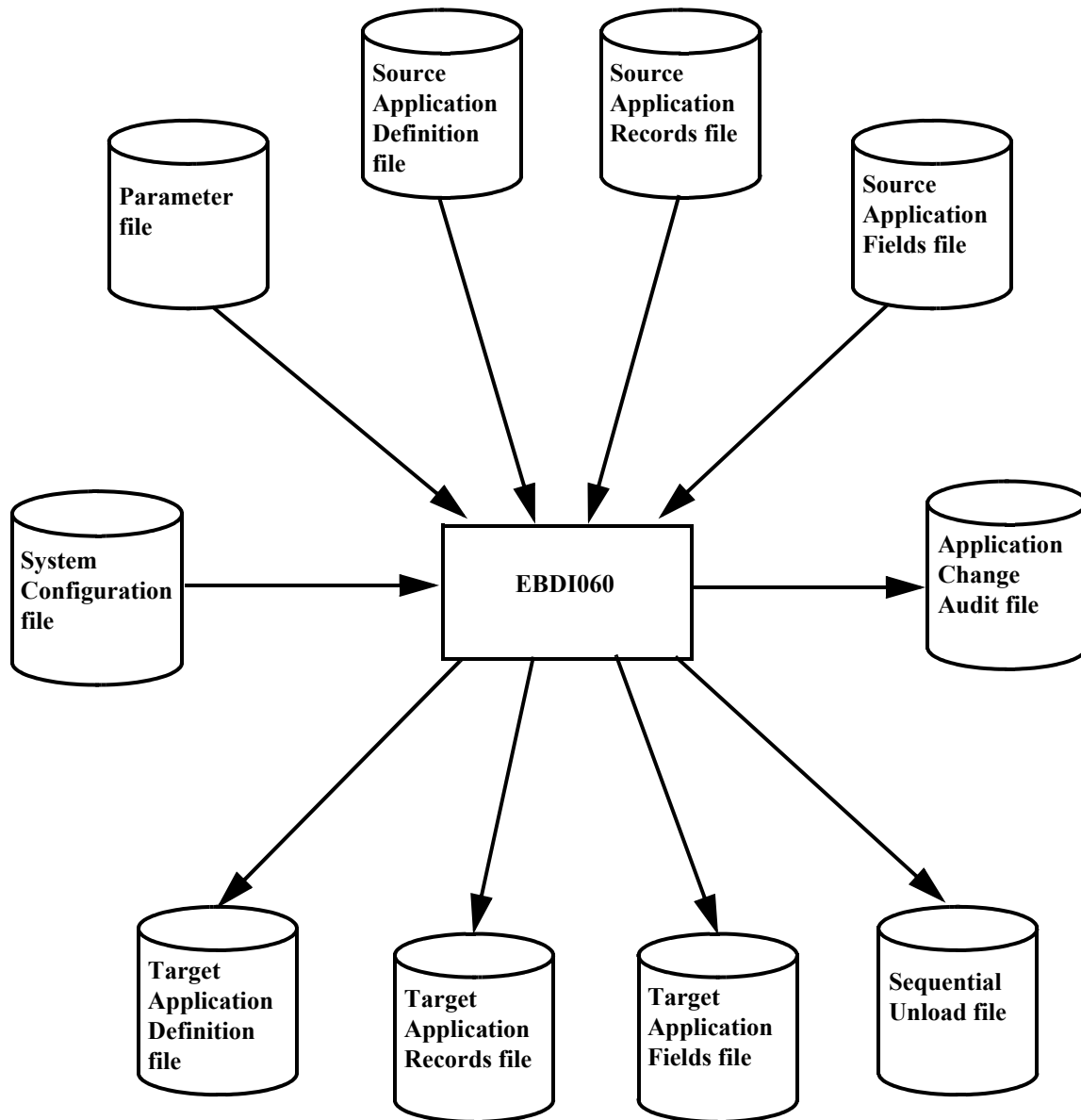
Position	Field Content	COBOL Picture	Format	Length	Description
001 – 005	DEL	X(05)	AN	05	Constant value of DEL.
006 – 015	APPLbID	X(10)	AN	10	Constant value of APPLbID
016 – 024	Filler	X(09)	AN	09	Not used
025 – 034	Application ID to be deleted	X(10)	AN	10	Application ID to be deleted.
035 – 035	Filler	X(01)	AN	01	Not used
036 – 036	Send/Receive of deleted Application ID	X(01)	AN	01	Valid values: <b>S</b> – Send <b>R</b> – Receive
037 – 042	Filler	X(06)	AN	06	Not used
043 – 043	Delete Switch (Y/N)	X(01)	AN	01	<b>Y</b> – Deletes the existing application from the Copy To files.
044 – 080	Blank	X(37)	AN	37	

*Parameter File Layout (Unload Function)*

Position	Field Content	COBOL Picture	Format	Length	Description
001 – 005	UNLD <b>b</b>	X(05)	AN	05	Constant value of UNLD <b>b</b> .
006 – 015	Copy from Application ID	X(10)	AN	10	Copy from Application ID of application to be copied.
016 – 016	Filler	X(01)	AN	01	Not used
017 – 017	Copy from Send/Receive	X(01)	AN	01	Copy from Send/Receive of application to be copied. Valid values: <b>S</b> – Send <b>R</b> – Receive.
018 – 024	bbbbTO <b>b</b>	X(07)	AN	07	Constant value of bbbbTO <b>b</b> .
025 – 034	Copy to Application ID	X(10)	AN	10	Copy to Applicant ID of application to be copied.
035 – 035	Filler	X(01)	AN	01	Not used
036 – 036	Copy to Send/Receive	X(01)	AN	01	Copy to Send/Receive of application to be copied. Valid values: <b>S</b> – Send <b>R</b> – Receive.
037-080	Filler	X(44)	AN	44	Not used

*Application Copy*

Figure 4.1 illustrates how to copy an application definition from one file to another.



**Figure 4.1 Program EBDI060 – Application Copy**

## Application Definition Maintenance

**EDIG100**

### Purpose

The Application Definition Maintenance (EDIG100) program can be used to load or reload application definitions from a sequential Application Definition file.

You can use this program to take advantage of existing file structure information, such as copybooks, when you create application definitions. Using your system Editor, you can modify your existing file structure information to reflect the sequential Application Definition file layout. You can also use this existing file structure information as input to the Application Definition Maintenance program to create application definitions.

### Operation

The Application Definition Maintenance program runs as a batch job step.

### Files Used

#### Input

The following files are used as input:

DDname	Filename
EDIAPD	Application Definition file
ERRCTL	Error Message and Control file
EDICFG	System Configuration file
EDICNTL	Control Card file

#### Output

The following files are used as output:

DDname	Filename
APPFLD	Application Field file
APPHDR	Application Header file
APPRCT	Application Record file

## Reports

The EDIG100 program produces the following reports:

DDname	Report Name	Report ID
EDILOG	Processing Log	G100LOG
EDISUM	Summary Report	G100SUM
EDIRPT	Status Report	G100RPT

## Control Parameters

The Application Definition Maintenance program allows the following operations:

### Operations

LOAD (Default operation)  
 or  
 RELOAD Reload all application definitions.

### Application Definition File Contents

The Application Definition file (EDIAPD) is a sequential file with 255-byte records that contain all the information required to create one or more application definitions.

Structure the records in this file to reflect the Mapping Integration screens on which you can view and modify the application definition. Individual fields on each record correspond to fields on these screens.

See Chapter 6 in the *Gentran:Basic for zSeries Release 6.4 User's Guide* for detailed screen and field descriptions.

The Application Definition file consists of record types that are identified by a 2-byte record type in positions 1-2 of each record. These records are:

- Application Header Definition Records (record types H1, H2, H3, and H4)**
  - HDR1-RECORD (record type H1)  
Contains information that corresponds to the Application Data ID screen (EDIM552).
  - HDR2-RECORD (record type H2)  
Contains information that corresponds to the Application Partner Reference screen (EDIM555).
  - HDR3-RECORD (record type H3)  
Contains information that corresponds to the Outbound Envelope field definition for all standards portion of the Application Envelope Definition record (EDIM558).

HDR4-RECORD (record type H4)

Contains information that corresponds to the Outbound Envelope field definition for individual interchanges portion of the Application Envelope Definition record (EDIM558).

- **Application Record Definition (record type R)**  
Contains information that corresponds to the Application Records screen (EDIM553).
- **Application Field Definition (record type F)**  
Contains information that corresponds to the Application Fields screen (EDIM554).

Each application definition on the file must start with header information consisting of Application Header records H1 and H2, and optionally H3- and/or H4-type header records. Following the header records, one R-type record for each record in the application must be present. Each R-type field is followed by F-type records for each field in the record. This file structure is summarized below:

H1	(required)
H2	(required)
H3	(optional)
H4	(optional)
R	(one per record)
F	(one field per record)

You can repeat the above record sequence for multiple application definitions.

See Chapter 6 in the *Gentran:Basic for zSeries Release 6.4 User's Guide* for application field descriptions.

## File Record Layouts

The following tables describe the Application Header Definition record layouts.

### *Application Header Definition – HDR1-Record*

Position	Length	Format	Field Name	Value
01-02	02	AN	APPL RECORD TYPE	H1
03-12	10	AN	APPLICATION DATA ID	
13-13	01	AN	SEND/RECEIVE	
14-43	30	AN	DESCRIPTION	
44-49	06	AN	FUNCTIONAL GROUP – 1	
50-55	06	AN	FUNCTIONAL GROUP – 2	
56-61	06	AN	FUNCTIONAL GROUP – 3	
62-67	06	AN	FUNCTIONAL GROUP – 4	

Position	Length	Format	Field Name	Value
68-73	06	AN	FUNCTIONAL GROUP – 5	
74-78	05	N	RECORD TYPE START POS	
79-80	02	N	LENGTH	
81-85	05	N	RECORD LENGTH	
86-86	01	AN	FIXED/VARIABLE	
87-89	03	AN	DIVISION CODE	
90-90	01	AN	UPDATE ALLOWED	
91-255	165	AN	UNUSED	

*Application Header Definition – HDR2-Record*

Position	Length	Format	Field Name	Value
01-02	02	AN	APPL RECORD TYPE	H2
03-12	10	AN	APPLICATION DATA ID	
13-13	01	AN	SEND/RECEIVE	
14-28	15	AN	PARTNER FIELD NAME	
29-43	15	AN	PARTNER QUAL FIELD NAME	
44-58	15	AN	USER FIELD NAME	
59-73	15	AN	USER QUAL FIELD NAME	
74-88	15	AN	FIELD NAME – 1	
89-103	15	AN	FIELD NAME – 2	
104-118	15	AN	FIELD NAME – 3	
119-133	15	AN	INTERCHANGE VERSION	
134-148	15	AN	GROUP/TRANSACTION VERSION	
149-163	15	AN	MULTIPLE ENVELOPE ID	
164-255	112	AN	UNUSED	

*Application Header Definition – HDR3-Record*

Position	Length	Format	Field Name	Value
01-02	02	AN	APPL RECORD TYPE	H3
03-12	10	AN	APPLICATION DATA ID	
13-13	01	AN	SEND/RECEIVE	
14-28	15	AN	INTERCHANGE SENDER ID	
29-43	15	AN	INTERCHANGE SENDER ID QUAL	
44-58	15	AN	GROUP SENDER ID	
59-73	15	AN	GROUP SENDER ID QUAL	
74-88	15	AN	INTERCHANGE RECEIVER ID	
89-103	15	AN	INTERCHANGE RECEIVER ID QUAL	
104-118	15	AN	GROUP RECEIVER ID	
119-133	15	AN	GROUP RECEIVER ID QUAL	
134-148	15	AN	INTERCHANGE CONTROL NUMBER	
149-163	15	AN	\$\$ADD BATCH ID	
164-178	15	AN	GROUP CONTROL NUMBER	
179-193	15	AN	TRANSACTION CONTROL NUMBER	
194-255	62	AN	UNUSED	

*Application Header Definition – HDR4-Record*

Position	Length	Format	Field Name	Value
01-02	02	AN	APPL RECORD TYPE	H4
03-12	10	AN	APPLICATION DATA ID	
13-13	01	AN	SEND/RECEIVE	
14-28	15	AN	BG COMM ID	
29-43	15	AN	BG COMM PASSWORD	
44-58	15	AN	ISA/UNB TEST INDICATOR	
59-73	15	AN	UNB/STX APPL REFERENCE	
74-88	15	AN	UNB/STX PRIORITY CODE	
89-103	15	AN	STX RECIPIENT TRANS REFERENCE	
104-255	152	AN	UNUSED	



*Application Record Definition*

Position	Length	Format	Field Name	Value
01-02	02	AN	APPL RECORD TYPE	R
03-05	03	N	SEQ NO	
06-15	10	AN	RECORD TYPE	
16-16	01	AN	AREA	
17-46	30	AN	DESCRIPTION	
47-52	06	AN	LOOP ID	
53-57	05	N	MAX USE	
58-62	05	N	RECORD LENGTH	
63-63	01	AN	MANDATORY CODE	
64-64	01	AN	WRITE FLAG	
65-255	191	AN	UNUSED	

*Application Field Definition*

Position	Length	Format	Field Name	Value
01-02	02	AN	APPL RECORD TYPE	F
03-05	03	N	RECORD SEQ NO	
06-08	03	N	FIELD NO	
09-23	15	AN	FIELD NAME	
24-25	02	AN	DATA TYPE	
26-27	02	N	FIELD LENGTH	
28-32	05	N	FIELD START	
33-52	20	AN	FIELD VALUE	
53-82	30	AN	FIELD DESCRIPTION	
83-83	01	AN	MANDATORY CODE	
84-255	172	AN	UNUSED	

## Application Sequence Renumbering

**EBDI090**

### Purpose

The Application Sequence Renumbering (EBDI090) program detects Application ID errors, renumbers the respective application records and fields, and then displays them on a report. This report also includes a printout of the maps that the program has determined are linked to the application definition, by reading the Application Link file. The maps must then be run through the re-synch process, using the EBDI089 program to bring both application and translation definitions back into synch with each other.

A parameter record must contain this information: Application ID, Send/Receive, Starting Record Sequence Number, Record Sequence Increment, Starting Field Sequence Number, Field Sequence Increment, and Restart Switch. If a starting and increment record or field sequence numbers are not entered, the default number is by tens (10s). When the field is left blank or invalid data is entered in the Restart Switch field, the default value is N.

**The following condition(s) are checked and an appropriate message is written on the output report for each condition:**

If no fields are defined for an Application Record, the following message is printed on the report along with the record:

**\*\*\* W A R N I N G: THERE ARE NO FIELDS DEFINED FOR THE FOLLOWING RECORD \*\*\***

Below the message the following information is printed:

- Sequence number of the Application Record
- Record ID of the Application Record
- Type of Application Record
- Description of Application Record

**Processing terminates if any of the following conditions are true:**

- The Application ID or Send/Receive field on the parameter record is left blank
- The Application ID on the parameter record does not exist on the Application (APPHDR) file
- The starting sequence and increment multiplied by the number of records or fields exceed the maximum number of lines possible
- No Application records exist for the application
- Invalid data is entered on the parameter record
- All files were not successfully opened
- An invalid return code is received on a READ or WRITE statement
- More than 300 transaction maps are linked to the application definition

If you have Change Audit for Application enabled on the "1" record of the EDICFG file, a new record will be written to the Application Change Audit file (EDIACHA) for each application that is renumbered.

### Application Link File Processing

The Application Link file is read, and a list of the transaction maps associated with this Application ID is listed on the report with a message indicating that these maps need to be re-synched. If no transaction maps are linked to this application definition, the word NONE is printed under the message.

### Restart Process

If a fatal error occurs in the renumbering process that causes the program to abend, the restart process can be invoked. This can only be done if the program had completed building the temporary files and was in the process of deleting the original application records and fields records. To perform the restart process successfully, the following conditions must be true:

- The value **Y** in position 30 of the parameter record must be present.
- The temporary files created on the initial run are still available and intact.
- The original application records and fields records have been deleted from the online file.

**Note:** All original application records and fields records must be deleted from the online prior to running the restart process or you end up with duplicate records that have different sequence numbers.

### Files Used

#### Input

The following files are used as input:

Filename	Description
SYS001	Parameter file This file contains the application definition and the necessary information to renumber it.
SYS040	Application Header file This file contains all the header records for applications that have been identified to the system.
SYS050	Application/Translation Link file This file contains a cross-reference of all application and transaction definitions.
EDICFG	System Configuration file

*Input/Output*

The following files are used as input and output:

<b>Filename</b>	<b>Description</b>
SYS041	Application Records file This file contains all the records for applications that have been identified to the system.
SYS042	Application Fields file This file contains all the field records for applications that have been identified to the system.
SYS046	Temporary Application Records file This file contains the records to be used in the restart process. If all processing is successful, this file may be deleted after the program is finished. If errors occur, this file can be used in the restart process to finish building the renumbered Application records.
SYS047	Temporary Application Fields file This file contains the records to be used in the restart process. If all processing is successful, this file may be deleted after the program is finished. If errors occur, this file can be used in the restart process to finish building the renumbered Application fields.

*Output*

The following file is used as output:

<b>Filename</b>	<b>Description</b>
SYS002	Application Sequence Renumbering Report This file contains the report of the application renumbering.
EDIACHA	Application Change Audit File

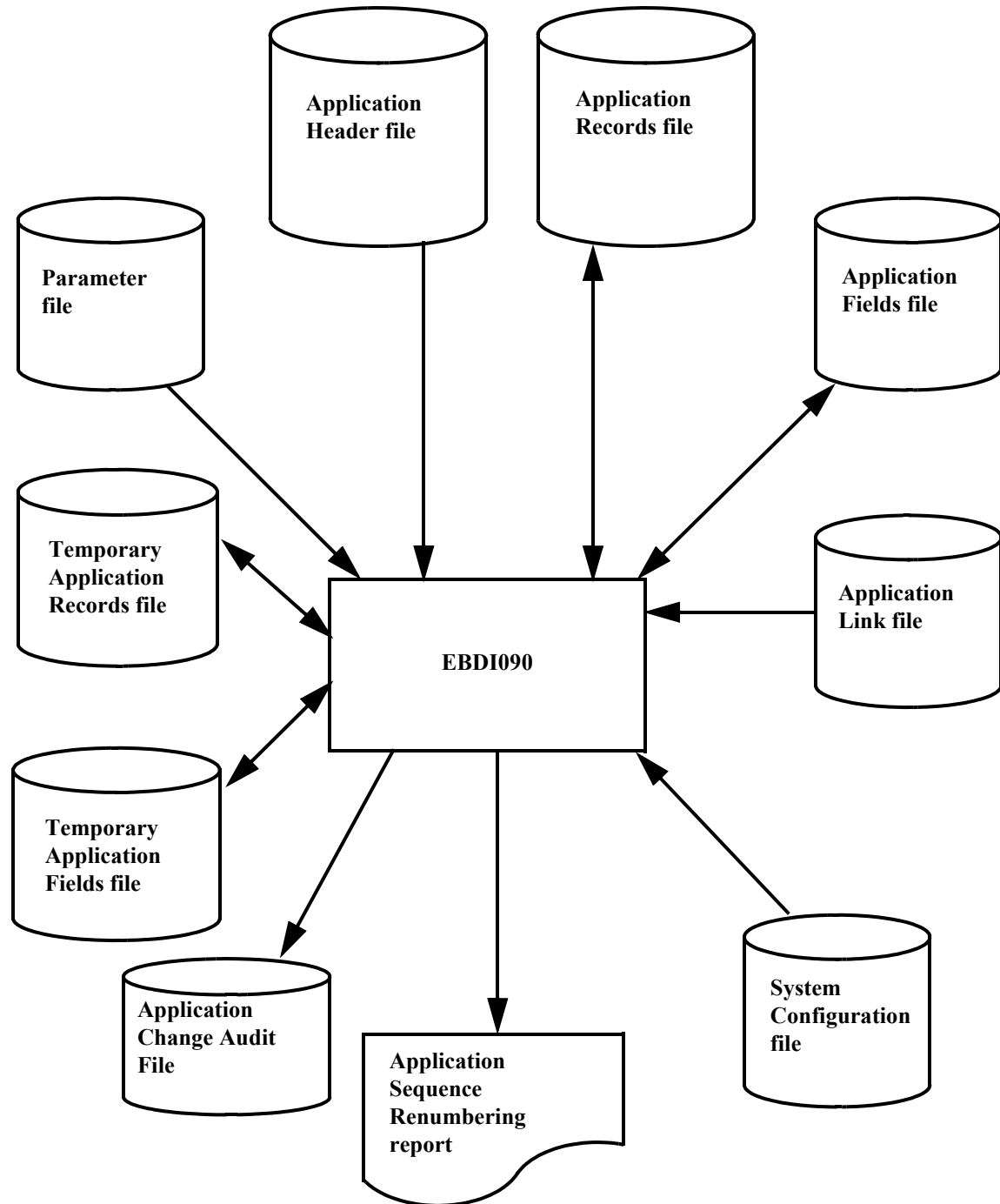
*Parameter File Layout*

The following table describes the parameter file layout:

Position	Field Content	COBOL Picture	Format	Length	Description
001 – 010	Application ID	X(10)	AN	10	The name assigned to the application definition being renumbered.
011 – 011	Application Send/Receive	X(01)	AN	01	Valid values: <b>S</b> – Send <b>R</b> – Receive.
012 – 014	Filler	X(03)	AN	03	Not used
015 – 017	Starting Record Sequence #	9(03)	N	03	The number at which the record renumbering begins.
018 – 020	Record Increment	9(03)	N	03	The number used to increment the record sequence.
021 – 022	Filler	X(02)	AN	02	Not used
023 – 025	Starting Field Sequence #	9(03)	N	03	The number at which the field record renumbering begins.
026 – 028	Field Increment	9(03)	N	03	The number used to increment the field record sequence.
029 – 029	Filler	X(01)	AN	01	Not used
030 – 030	Restart Switch	X(01)	AN	01	Code: <b>Y</b> – if performing the restart process <b>N</b> – if not performing a restart.

*Application Sequence Renumbering Report (EBDI090 Program)*

Figure 4.2 illustrates the Application Sequence Renumbering report.



**Figure 4.2 Program EBDI090 – Application Sequence Renumbering Report**

## Sample Reports

Figure 4.3 and Figure 4.4 illustrate the Application Sequence Renumbering report (Successful Completion) and the Application Sequence Renumbering report (Error Detail).

```

EBDI090      DATE: 12/01/2005      APPLICATION DEFINITION RENUMBER      PAGE 1
GENTRAN MAPPING INTEGRATION      TIME: 12:00

PARAMETER RECORD:
  APPLICATION ID: BS811      S
  STARTING REC SEQ: 010
  INCREMENT: 005
  STARTING FLD SEQ: 010
  INCREMENT: 002
  RESTART: N

THE FOLLOWING TRANSACTIONS MUST BE RUN THROUGH THE RE-SYNCH PROGRAM --> EBDI089:

BS811      S

*** SUCCESSFUL COMPLETION OF APPLICATION RENUMBERING ***

FINAL TOTALS:
  APPLICATION RECORDS READ: 64
  APPLICATION FIELDS READ: 1202
  TEMPORARY RECORDS WRITTEN: 64
  TEMPORARY FIELDS WRITTEN: 1202
  APPLICATION RECORDS DELETED: 64
  APPLICATION FIELDS DELETED: 1202
  NEW APPLICATION RECORDS WRITTEN: 64
  NEW APPLICATION FIELDS WRITTEN: 1202
  CHANGE AUDIT RECORDS WRITTEN: 1

```

**Figure 4.3 Application Sequence Renumbering Report (Successful Completion)**

```

EBDI090      DATE: 12/01/2005      APPLICATION DEFINITION RENUMBER
GENTRAN MAPPING INTEGRATION      PAGE 1
                                TIME: 12:00

PARAMETER RECORD:
  APPLICATION ID: BS811      S
  STARTING REC SEQ: 010
  INCREMENT: 010
  STARTING FLD SEQ: 010
  INCREMENT: 005
  RESTART: N

***** PROCESSING ENDED ABNORMALLY *****
* STARTING POSITION AND INCREMENT USED *
* ARE GREATER THAN THE MAXIMUM NUMBER OF *
* ALLOWABLE ENTRIES ==> 999. *
* -- USE SMALLER INCREMENT -- *
***** JOB CANCELLED *****

*** PROCESSING ENDED ABNORMALLY -- FINAL COUNT BELOW ***

      FINAL TOTALS:
      APPLICATION RECORDS READ: 64
      APPLICATION FIELDS READ: 220
      TEMPORARY RECORDS WRITTEN: 15
      TEMPORARY FIELDS WRITTEN: 182
      APPLICATION RECORDS DELETED: 0
      APPLICATION FIELDS DELETED: 0
      NEW APPLICATION RECORDS WRITTEN: 0
      NEW APPLICATION FIELDS WRITTEN: 0

```

**Figure 4.4 Application Sequence Renumbering Report  
(Error Detail)**



## Field Definitions

### Parameter Record

**Application ID:**

This field displays the application definition being renumbered.

**Starting Rec Seq:**

This field identifies the sequence number of the record (the order of the record within the document).

**Increment:**

This field identifies the number that the program used to increment the record sequence number.

**Starting Fld Seq:**

This field identifies the sequence number of the field (the order of the field within the record).

**Increment:**

This field identifies the number that the program used to increment the field sequence number.

**Restart:**

This field displays the switch entered in the parameter record indicating whether to restart the application when processing is complete (**Y** = Yes, **N** = No). The default value is **N**.

See the following Restart Process section for more details.

**The Following Transactions Must Be Run Through the Re-Synch program —> EBDI089:**

This field identifies the Transaction ID(s) of the transactions that must be processed through the Batch Transaction Mapping Definition Re-Synch program (EBDI089).

**(Message Field):**

This field displays a message indicating any errors encountered during program execution, or the successful completion of sequence renumbering.

### Final Totals

**Application Records Read:**

This field displays the total number of application records read during execution of the program.

**Application Fields Read:**

This field displays the total number of application fields read during execution of the program.

**Temporary Records Written:**

This field displays the total number of records written to the temporary file during execution of the program.

**Temporary Fields Written:**

This field displays the total number of fields written to the temporary file during execution of the program.

**Application Records Deleted:**

This field displays the total number of application records deleted during execution of the program.

**Application Fields Deleted:**

This field displays the total number of application fields deleted during execution of the program.

**New Application Records Written:**

This field displays the total number of new application records written during execution of the program.

**New Application Fields Written:**

This field displays the total number of new application fields written during execution of the program.

**Change Audit Records Written:**

This field displays the total number of Application Change Audit Records written during the execution of this program. This will only be displayed if you have Change Audit for Application enabled on the EDICFG file.

***Restart Process***

If a fatal error occurred in the renumbering process that caused the program to abend, the restart process can be invoked. This can only be done if the program has reached the point of having the temporary files built and was in the process of deleting the original application records and fields records. To perform the restart process, the following conditions must be true:

- The value **Y** in position 30 of the parameter record must be present.
- The temporary files created on the run are still available.
- The original application records and fields records have been deleted from online.

---

## Batch Separator

## EBDI930

### Purpose

The Batch Separator program (EBDI930) is the main control program used to separate inbound EDI batch data. The driver separates the interchanges into a temporary file, identifies the type of interchange (X12, EDIFACT, TRADACOMS, or XML data) and then passes it for processing by the Interchange Priority Lookup and Router programs. The program verifies that valid headers and trailers are present in the data and passes invalid data to a special error file.

Use this program to separate data that comes in as a large batch file. The separated data will run through the inbound process more efficiently.

### Operation

The Batch Separator is executed as a batch program (by the JCL: EXEC930). The driver reads the inbound data from a sequential file and identifies the start of an interchange and then looks for the end of the interchange. All the data in the interchange is written to a temporary sequential file.

If either the header or trailer is invalid (or not found), then the data is passed to an error file. If the data has a valid header and trailer then the file is passed to a Priority Lookup program. The Lookup program determines the destination file for the data by evaluating the contents of the interchange, group, and/or transaction headers. After the destination is determined, the Separator I/O program writes the data to the destination file. If multiple transaction IDs are present in the interchange, the one with the highest priority is used to set the destination file.

**Note:** The interchange remains as a complete unit of work.

### Key Points

- Four online screens are used to enter the separator information (stored in a VSAM file). These screens are accessible from the Administration Maintenance Menu (see chapter 5 of the *Gentran:Basic User's Guide*).
- All input data must be in 80 byte records.
- Up to 100 separator files can be used.
- There are six batch programs.
- XML data must begin with a '<?XML' identifier. The end of XML data is determined by the presence of a '</doc id>' that corresponds to the '<doc id>' identifier at the beginning. It is separated into a specific XML file.
- Incorrectly formed interchanges are separated into a error file. Interchanges for which a file is specified but which is not present in the JCL go to a slide file. Interchanges with multiple transaction sets are sent to the separator file that is identified with the highest priority (specified on the screens).

### Subprograms

The following subprograms are used in the Batch Separator operation.

- X12 Interchange Priority Lookup                      EBDI931
- EDIFACT Interchange Priority Lookup                EBDI932

- TRADACOMS Interchange Priority Lookup EBDI933
- XML Interchange Priority Lookup EBDI93X
- Separator I/O program EBDI940
- Batch Router EBDI945

### Lookup Programs Operation

The information used by the Lookup programs from the envelopes is:

- Interchange Sender ID and Qualifier
- Interchange Receiver ID and Qualifier
- Interchange Test/Production Indicator
- Group Functional ID
- Group Version
- Group Sender ID
- Group Receiver ID
- Transaction ID

The Lookup programs use the information entered in the Separator Control file to determine the destination file. This information is entered on the Separator screens accessed through the Administrative Main Menu. (See the *Gentran:Basic User's Guide* for details on the Separator screens.)

DDName	Usage	Description
EDICNTL	Input	Instream parameter control file
EDISINP	Input	Inbound data – EDI and XML
EDISTEMP	I/O	Temporary hold file for the interchange data
EDISCNTL	I/O	Temporary hold file for the control information for the interchange data in EDISTEMP
EDISTHDR	I/O	Temporary hold file for the headers (interchange, group and transactions) for use by the Lookup programs
EDIRSEP	Input	Separation records VSAM file
SYSERR	Output	Error file – holds all interchanges that have errors and any unidentified data in the inbound file
SYSSLD	Output	Slide file – holds interchanges which have a destination file assigned but the file is not present in the JCL
SYSXML	Output	XML file – all valid XML data is passed here. Invalid XML data is sent to the SYSERR file.
SYS001 – SYS100	Output	Separator Destination files
EDISUM	Output	Summary Report

*Instream Parameters*

Trace parameter – all subprograms will write information about processing.

Position	Field Content	COBOL Picture	Format	Length	Description
1 – 8	TRACE ON	X(8)	AN	8	Instructs the program to write trace information to the SYSOUT file. Helpful in determining the processing of the lookup programs.

Detail Report parameter – provides additional information on the EDISUM report to show the interchanges, groups and transactions that are in each interchange. See sample report below.

Position	Field Content	COBOL Picture	Format	Length	Description
1 – 16	REPORT DETAIL ON	X(16)	AN	16	Instructs the program to write auditing information about each interchange encountered.

Open file parameters – causes an explicit open of the file specified. This is needed to clear out any previous data from the file for cases where not all destination files are used during each execution of the Separator.

Position	Field Content	COBOL Picture	Format	Length	Description
1 – 13	OPEN FILE SYS	X(13)	AN	13	Identifies that a open is to be performed on the file (SYS###).
14 – 16	###	9(3)	N0	3	Number that identifies the file.

## COBOL Copybook Generator

EDIG110

### Purpose

The COBOL Copybook Generator (EDIG110) program creates COBOL copybooks from application definitions for use in application programs that interface with Gentran:Basic. This program also produces a formatted report of the application definition.

**Note:** The COBOL Copybook Generator program does not generate copybook fields correctly if there are Applications field redefined over other application fields.

### Operation

The COBOL Copybook Generator program runs as a batch job step.

### Files Used

#### Input

The following files are used as input:

DDname	Filename
APPHDR	Application Header file
APPRCT	Application Record file
APPRCT1	Application Record Path
APPFLD	Application Field file
APPFLD1	Application Field Path
ERRCTL	Error Message and Control file
EDICFG	System Configuration file
EDICNTL	Control Card file

#### Output

The following file is used as output:

DDname	Filename
EDICCB	COBOL Copybook file (sequential 80 bytes)

#### Reports

The EDIG110 program produces the following reports:

DDname	Report Name	Report ID
EDILOG	Processing Log	G110LOG
EDISUM	Summary Report	G110SUM

DDname	Report Name	Report ID
EDIRPT	Application Definition Report	G1100RPT

## Control Parameters

The COBOL Copybook Generator program allows the following operation and selection criteria to be specified through control parameters.

The control parameter is specified as follows:

### Operations

SELECT (Default operation)

### Criteria

#### Default

APPLICATION-DEFINITION	ALL
SEND-RECEIVE	ALL
REDEFINE	NO
VALUE-CLAUSE	ALL

### Valid Criteria Values

#### Application-Definition

One to ten characters that identify the application definition used to create the COBOL copybook.

aaaaaaaaaa  
ALL

#### Send-Receive

One to three characters that indicate the direction of the application definition.

S	Send/Outbound application definition.
R	Receive/Inbound application definition.
ALL	Both Inbound and Outbound application definitions are selected.

#### Redefine

One to three characters that indicate whether or not the generated copybook redefines a single area in working storage for all record types.

YES  
NO

**Note:** If Redefine = Yes and the application is defined as variable length, the copybook starts with a 2000-byte record definition.

**Value-Clause**

Characters that indicate how COBOL VALUE statements in the copybook are generated.

RECORD-TYPE	VALUE statements are generated for all fields, if any, that are defined with the starting position and length of the application definition record type.
FIELD-VALUE	VALUE statements are generated for all fields that have a field value specified.
NONE	No VALUE statement is generated.
ALL	VALUE statements are generated for record types and field values.



---

## Code Table Change Audit Maintenance

**EBDI068**

### Purpose

The Code Table Change Audit Maintenance program (EBDI068) maintains the Code Table Change Audit file. The program functions include deleting, archiving, and restoring change audit information.

The Code Table change audit file contains records that track changes made to the Mapping Code Table files. Add, delete, and update functions via the online Mapping Code Table subsystem result in records being written to the Code Table Change Audit file. The Batch copy utility (EBDI076) will also write records to the Code Table change audit file indicating add, delete, and replace functions. The online Upload process (EDIX235), using its component EDIR076, will also write a change audit record to indicate an add or replace event.

### Operation

The Code Table Change Audit Maintenance program operates in the following general modes:

#### **Periodic Housekeeping**

On a periodic basis (anywhere from daily to monthly depending on volume and retention requirements) the Housekeeping operation deletes and optionally archives data based on specified retention periods.

#### **Ad Hoc Maintenance (Purge and Restore)**

In addition to the regularly scheduled housekeeping runs, the Code Table Change Audit Maintenance's Purge and Restore functions deletes and optionally archives selected data and restores data from archives.

### Processing

The Code Table Change Audit Maintenance program processes documents using the following three operations:

- Housekeeping
- Purge
- Restore

#### *Housekeeping*

If the Housekeeping operation is requested, Code Table Change Audit Maintenance Deletes and optionally archives the change audit records that exceed the retention periods specified in the control cards. An option of the Housekeeping operation is to archive all selected change audit records to a sequential output file.

#### *Purge*

If the Purge operation is requested, change records are removed if they match ALL criteria specified. An option of the Purge operation is to archive selected change audit records to a sequential output file.

### Restore

If the Restore operation is requested, change audit records are restored from the Archive file to the Change Audit if the records match ALL selection criteria specified.

### Files Used

#### Input

The following files are used as input:

DDname	Filename
CDEARCH	Code Table Change Audit Archive File
EDICCHA	6.4 VSAM Code Table Change Audit File
ERRCTL	Error Message and Control file
EDICFG	System Configuration file
EDICNTL	Control Card file

#### Output

The following files are used as output:

DDname	Filename
CDEARCH	Code Table Change Audit Archive File
EDICCHA	6.4 VSAM Code Table Change Audit File

#### Reports

The EBDI068 program produces the following reports:

DDname	Report Name	Report ID
CDEAUDIT	Code Table Change Audit Maintenance Report	
EDILOG	Code Table Change Audit Processing Log	G110LOG
EDISUM	Code Table Change Audit Summary Report	G110SUM

### Control Card Options

#### Operations

- Housekeeping (Default Operation)
- Purge
- Restore

*Selection Criteria***Defaults**

For Housekeeping operation:

RETENTION-DAYS

For Purge and Restore operations:

FROM-UPDATE-DATE 01/01/1900

FROM-UPDATE-TIME 00:00

TO-UPDATE-DATE 12/31/2099

TO-UPDATE-TIME 23:59

TABLE-ID ALL

For Purge and Housekeeping operations:

ARCHIVE NO

*Valid Criteria Values***Archive**

This controls whether an archive is to be created for deleted change records.

NO

YES

**From-Update--Date**

Specifies the date that the update was applied to the Code Table record. Starts the range.

Used with To-Update-Date.

MM/DD/YYYY

**From-Update-Time**

Specifies the time that the update was applied to the Code Table record. Starts the range.

Used with To-Update-Time.

HH:MM

**Retention-Days**

Specifies the number of days that have elapsed since the update was applied.

000 - 999

**To-Update--Date**

Specifies the date that the update was applied to the Code Table. Ends the range. Used with From-Update-Date.

MM/DD/YYYY

**To-Update-Time**

Specifies the time that the update was applied to the Code Table. Ends the range. Used with From-Update-Time.

HH:MM

**Table-Id**

A 10-character identification used to restrict maintenance by Code Table ID.

ALL

XXXXXXXXXX

## Code Table Copy From File to File

EBDI076

### Purpose

The Code Table Copy (EBDI076) program is one of the five Test/Production Integration Copy programs. The Code Table Copy is used to copy a table definition from one set of files to another set of files. It can also be used to delete translation tables from the target files.

Tables can be directly copied (i.e., VSAM to VSAM) or to a sequential unload file for later upload.

**Caution:** Table files are sensitive to the partner profile trading mode specified on the Configuration file.

The table to be copied is specified in the first parameter record. The table to be created is specified in the second parameter record. Also, the table to be copied must exist on the source definition file to be copied (unless the DELETE function is specified).

If you have Change Audit for Code Tables enabled on the “1” record of the EDICFG file, a new record will be written to the Code Table Change Audit file (EDICCHA) for each code table that has been added, deleted, or replaced.

### Files Used

#### Input

The following files are used as input:

Filename	Description
SYS001	Parameter file Two records. The first contains the name of the table to be copied from the source. The second record contains the name of the table to be created on the target file. Multiple pairs of records may be entered. The Unload function can not be intermixed with the Copy and Delete functions.
SYS020	Source Table Definition file The definitions from which the table is copied.
SYS021	Source Code Translation Table file The code translation records from which the table is copied.
SYS022	Source Data Translation Table file The data translation records from which the table is copied.
SYS023	Source Code Validation Table file The code validation records from which the table is copied.
EDICFG	Gentran System Configuration file <b>Note:</b> If using the Change Audit feature, make sure the DD statement in your EXEC076 JCL for this file contains the dataset name as it appears in your target CICS region.

**Output**

The following files are used as output:

Filename	Description
SYS030	Target Table Definition Table file The definitions to which the table is copied, or from which it is deleted.
SYS031	Target Code Translation Table file The code Translation records to which the table is copied, or from which it is deleted.
SYS032	Target Data Translation Table the data translation records to which the table is copied, or from which it is deleted.
SYS033	Target Code Validation Table file the code validation records to which the table is copied, or from which it is deleted.
SYS060	Sequential Unload file
EDICCHA	Code Table Change Audit file <b>Note:</b> If using the Change Audit feature, make sure the DD statement in your EXEC076 JCL for this file contains the dataset name as it appears in your target CICS region.

**Parameter File Layout (Copy Function) – Partner/Qualifier Mode**

The following tables describe the Copy function parameter file layout.

**First Parameter Record**

Position	Field Content	COBOL Picture	Format	Length	Description
001 – 005	FROMb	X(05)	AN	05	Constant value of FROMb.
006 – 015	Copy from Table ID	X(10)	AN	10	Copy from Table ID to be copied.
016 – 016	Space	X(01)		01	
017 – 051	Copy from Partner ID	X(35)	AN	35	Copy from Partner ID of table to be copied.
052 – 052	Space	X(01)		01	
053 – 056	Copy from Partner Qualifier	X(04)	AN	04	Partner ID Qualifier of table to be copied.
057 – 080	Filler	X(24)		24	Not used

## Second Parameter Record

Position	Field Content	COBOL Picture	Format	Length	Description
001 – 003	TOb	X(03)	AN	03	Constant value of TOb.
004 – 013	Copy to Table ID	X(10)	AN	10	Copy to Table ID of table created.
014 – 014	Space	X(01)		01	
015 – 049	Copy to Partner ID	X(35)	AN	35	Copy to Partner ID of table created.
050 – 050	Spaces	X(01)		01	
051 – 054	Copy to Partner Qualifier	X(04)	AN	04	Copy to Partner ID Qualifier of table created.
055 – 060	Spaces	X(05)		05	
061 – 061	Replace Switch (Y/N)	X(01)	AN	01	<p><b>Y</b> – Deletes the existing transaction table from the Copy To files.</p> <p><b>N</b> – Process data, but system indicates the transaction table that need to be deleted.</p> <p><b>Note:</b> The system does not copy if Table exists on Copy To file and the Replace Switch is <b>N</b>.</p>
062 – 080	Spaces	X(18)	AN	18	

*Parameter File Layout (Unload Function) – Partner/Qualifier Mode***First Parameter Record**

Position	Field Content	COBOL Picture	Format	Length	Description
001 – 005	UNLD <b>b</b>	X(05)	AN	05	Constant value of UNLD <b>b</b> .
006 – 015	Copy from Table ID	X(10)	AN	10	Copy from Table to be copied.
016 – 016	Space	X(01)		01	
017 – 051	Copy from Partner ID	X(35)	AN	35	Copy from Partner ID of table to be copied.
052 – 052	Space	X(01)		01	
053 – 056	Copy from Partner/Qualifier	X(04)	AN	04	Partner ID Qualifier of table to be copied.
057 – 080	Filler	X(24)		24	

**Second Parameter Record**

Position	Field Content	COBOL Picture	Format	Length	Description
001 – 003	TO <b>b</b>	X(03)	AN	03	Constant value of TO <b>b</b> .
014 – 013	Copy to Table ID	X(10)	AN	10	Copy to Table ID of table created.
014 – 014	Space	X(01)		01	
015 – 049	Copy to Partner ID	X(35)	AN	35	Partner ID of table being created.
050 – 050	Space	X(01)		01	
051 – 054	Copy to Partner/Qualifier	X(04)	AN	04	Copy to Partner ID Qualifier table created.
055 – 063	Spaces	X(08)		08	
064 – 064	Delete Switch (Y/N)	X(01)	AN	01	<b>Y</b> – Deletes the existing transaction table from the Copy To files.
065 – 080	Spaces	X(16)	AN	16	

**Parameter File Layout (Delete Function) – Partner/Qualifier Mode**

The following tables describe the Delete function parameter file layouts.

**First Parameter Record**

Position	Field Content	COBOL Picture	Format	Length	Description
001 – 005	DEL	X(05)	AN	05	Constant value of DELETE
006 – 015	CODE TBL	X(10)	AN	10	Constant value of CODE TBL
016 – 080	Filler	X(65)		65	Not used

**Second Parameter Record**

Position	Field Content	COBOL Picture	Format	Length	Description
001 – 003	Filler	X(03)	AN	03	Not used
004 – 013	Table ID being deleted	X(10)	AN	10	Delete Table ID
014 – 014	Space	X(01)		01	
015 – 049	Partner ID	X(35)	AN	35	Partner ID of table being deleted.
050 – 050	Spaces	X(01)		01	
051 – 054	Partner Qualifier	X(04)	AN	04	Partner ID Qualifier being deleted.
055 – 063	Spaces	X(08)		08	
064 – 064	Delete Switch (Y/N)	X(01)	AN	01	Y – Deletes the existing transaction table from the Copy To files.
065 – 080	Spaces	X(16)	AN	16	



*Parameter File Layout (Copy Function) – Relationship Mode***First Parameter Record**

Position	Field Content	COBOL Picture	Format	Length	Description
001 – 005	FROMb	X(05)	AN	05	Constant value of FROMb.
006 – 015	Copy from Table ID	X(10)	AN	10	Copy from Table to be copied.
016 – 016	Space	X(01)		01	
017 – 031	Copy from User ID	X(15)	AN	15	Copy from User ID of table to be copied.
032 – 032	Space	X(01)		01	
033 – 047	Copy from Partner ID	X(15)	AN	15	Copy from Partner ID of table to be copied.
048 – 080	Filler	X(33)		33	

**Second Parameter Record**

Position	Field Content	COBOL Picture	Format	Length	Description
001 – 003	TOb	X(03)	AN	03	Constant value of TOb.
014 – 013	Copy to Table ID	X(10)	AN	10	Copy to Table ID of table created.
014 – 014	Space	X(01)		01	
015 – 029	Copy to User ID	X(15)	AN	15	Copy to User ID of table being created.
030 – 030	Space	X(01)		01	
031 – 045	Copy to Partner ID	X(15)	AN	15	Copy to Partner ID table created.
046 – 060	Spaces	X(15)		15	
061 – 061	Delete Switch (Y/N)	X(01)	AN	01	Y – Deletes the existing transaction table from the Copy To files.
062 – 080	Spaces	X(18)	AN	18	

*Parameter File Layout (Delete Function) – Relationship Mode***First Parameter Record**

Position	Field Content	COBOL Picture	Format	Length	Description
001 – 005	DELETE	X(05)	AN	05	Constant value of DELETE.
006 – 015	CODE TBLD	X(10)	AN	10	Constant value of CODE TBLD.
016 – 080	Spaces	X(65)		65	

**Second Parameter Record**

Position	Field Content	COBOL Picture	Format	Length	Description
001 – 003	Spaces	X(03)		03	
014 – 013	Table ID being deleted	X(10)	AN	10	Delete Table ID.
014 – 014	Space	X(01)		01	
015 – 029	User ID of table being deleted	X(15)	AN	15	User ID of table being deleted.
030 – 030	Space	X(01)		01	
031 – 045	Partner ID of table being deleted	X(15)	AN	15	Partner ID of table being deleted.
046 – 063	Spaces	X(18)		18	
061 – 064	Delete Switch (Y/N)	X(01)	AN	01	Y – Deletes the existing transaction table from the Copy To files.
065 – 080	Spaces	X(16)	AN	16	

*Parameter File Layout (Unload Function) – Relationship Mode***First Parameter Record**

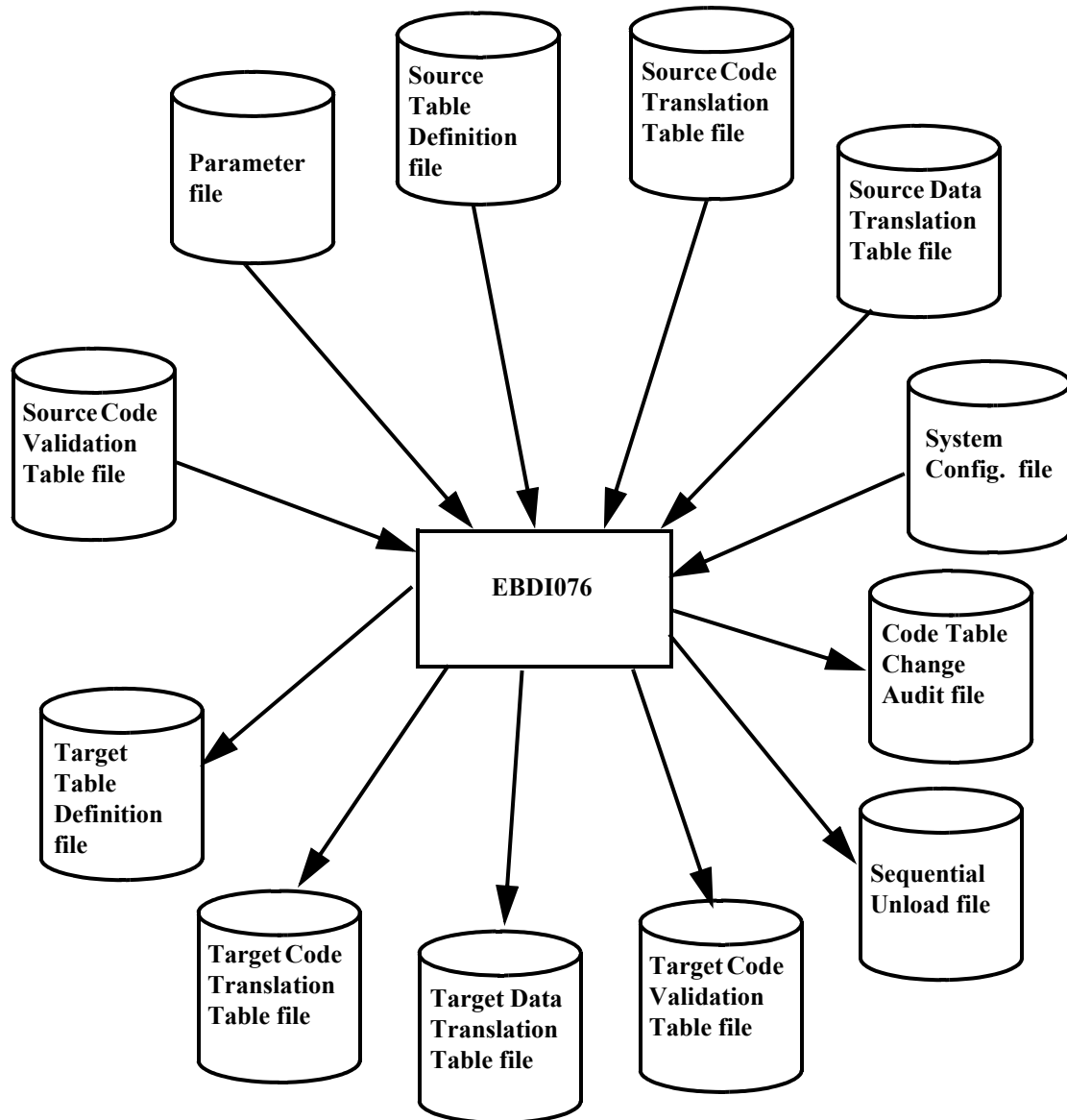
Position	Field Content	COBOL Picture	Format	Length	Description
001 – 005	UNLD6	X(05)	AN	05	Constant value of UNLD6.
006 – 015	Copy from Table ID	X(10)	AN	10	Copy from Table to be copied.
016 – 016	Space	X(01)		01	
017 – 031	Copy from User ID	X(15)	AN	15	Copy from User ID of table to be copied.
032 – 032	Space	X(01)		01	
033 – 047	Copy from Partner ID	X(15)	AN	14	Copy from Partner ID of table to be copied.
048 – 080	Filler	X(33)		33	

**Second Parameter Record**

Position	Field Content	COBOL Picture	Format	Length	Description
001 – 003	TO6	X(03)	AN	03	Constant value of TO6.
014 – 013	Copy to Table ID	X(10)	AN	10	Copy to Table ID of table to be created.
014 – 014	Space	X(01)		01	
015 – 029	Copy to User ID	X(15)	AN	15	Copy of User ID of table to be created.
030 – 030	Space	X(01)		01	
031 – 045	Copy to Partner ID	X(15)	AN	15	Copy to Partner ID of table to be created.
046 – 080	Spaces	X(35)		35	

*Code Table Copy From One File to Another*

Figure 4.5 illustrates how to copy a translation table from one file to another.



**Figure 4.5 Program EBDI076 – Code Table Copy**

---

## Configuration Change Audit Maintenance Program

**EBDI102**

The Configuration Change Audit Maintenance program (EBDI102) maintains the Configuration Change Audit file. The program functions include deleting, archiving, and restoring change audit information.

The Configuration change audit file contains records that track changes made to the System Configuration file. Update functions via the online Configuration subsystem result in records being written to the Configuration Change Audit file.

### Operation

The Configuration Change Audit Maintenance program operates in the following general modes:

#### **Periodic Housekeeping**

On a periodic basis (anywhere from daily to monthly depending on volume and retention requirements), the Housekeeping operation deletes and optionally archives data based on specified retention periods.

#### **Ad Hoc Maintenance (Purge and Restore)**

In addition to the regularly scheduled housekeeping runs, the Configuration Change Audit Maintenance Purge and Restore functions delete and optionally archive selected data and restore data from archives.

### Processing

The Configuration Change Audit Maintenance program processes documents using the following three operations:

- Housekeeping
- Purge
- Restore

#### *Housekeeping Operation*

If the Housekeeping operation is requested, Configuration Change Audit Maintenance deletes and optionally archives the change audit records that exceed the retention periods specified in the control cards. An option of the Housekeeping operation is to archive all selected change audit records to a sequential output file.

#### *Purge Operation*

If the Purge operation is requested, change records are removed if they match *all* criteria specified. An option of the Purge operation is to archive selected change audit records to a sequential output file.

#### *Restore Operation*

If the Restore operation is requested, change audit records are restored from the Archive file to the Change Audit if the records match *all* selection criteria specified.

## Files Used

### Input

Control cards and the Configuration Change Audit files are the principal input to Configuration Change Audit Maintenance.

Filename	Description
CFGARCH	Configuration Change Audit Archive File
EDIYCHA	6.4 VSAM Transaction Change Audit File
EDICNTL	Control Card File
ERRCTL	Error Message and Control File
EDICFG	System Configuration File

### Output

The following files are used as output:

Filename	Description
CFGARCH	Configuration Change Audit Archive File
EDIYCHA	6.4 VSAM Configuration Change Audit File

### Reports

Program EBDI102 produces the following reports:

Report Name	Description
CFGAUDT	Configuration Change Audit Maintenance Report
EDILOG	Configuration Change Audit Processing Log
EDISUM	Configuration Change Audit Summary Report

## Control Card Options

### Operations

- HOUSEKEEPING (Default Operation)
- PURGE
- RESTORE

*Selection Criteria*

**Defaults**

For Housekeeping operation:

RETENTION-DAYS

For Purge and Restore operations:

FROM-UPDATE-DATE 01/01/1900  
 FROM-UPDATE-TIME 00:00  
 TO-UPDATE-APPLIED-DATE 12/31/2099  
 TO-UPDATE-APPLIED-TIME 23:59  
 RECORD-TYPE ALL

For Purge and Housekeeping operations:

ARCHIVE NO

*Valid Criteria Values*

**Archive**

This controls whether an archive is to be created for deleted change records.

NO  
 YES

**From-Update-Date**

Specifies the date that the update was applied to the System Configuration record. Starts the range. Used with To-Update-Applied-Date.

MM/DD/YYYY

**From-Update-Time**

Specifies the time that the update was applied to the System Configuration record. Starts the range. Used with To-Update-Applied-Time.

HH:MM

**Record-Type**

A 10-character identification used to restrict maintenance by Configuration Record type.

ALL  
 B Gentran:Realtime processing options  
 E Gentran:Control processing options  
 T Clear Key processing options  
 V Commerce:Advantage processing options  
 0 Online processing options  
 1 Additional online processing options  
 2 Databank processing options  
 3 Gentran:Plus processing options  
 4 Gentran:Viewpoint processing options  
 7 Gentran:Structure processing options

**Retention-Days**

Specifies the number of days that have elapsed since the update was applied.

000 – 999

**To-Update-Applied-Date**

Specifies the date that the update was applied to the System Configuration record. Ends the range. Used with From-Update-Date.

MM/DD/YYYY

**To-Update-Applied-Time**

Specifies the time that the update was applied to the System Configuration record. Ends the range. Used with From-Update-Time.

HH:MM



## Data Router

**EBDI095**

### Purpose

The Data Router (EBDI095) program routes application or EDI data files based on data values within the input file. Selected sections of data (e.g., application documents, EDI transactions, or EDI interchanges) are written to specified output files based on user-supplied conditional expressions.

### Delimited “Variable Format” EDI Data Routing

**Note:** Delimited EDI data refers to the following types of data:  
ANSI, TDCC, UCS, EDIFACT, and ANA  
TRADACOMS.

The Data Router program is an optional step within the inbound flow. The data router replaces or works in conjunction with the existing Inbound Split (EBDI015) program. The following table explains when you can or cannot use the data router:

Running Condition	When to Use the Data Router
If error rejection is being used...	The data router cannot replace the Inbound Split program.
If the Inbound Split program is not being used...	The data router should run after the Inbound Editor (EBDI001).
If further data separation is needed beyond what the Inbound Split program provides...	The data router should run after EBDI015.

The output of the Data Router program is directly processed by the Inbound Mapping (EBDI041) program.

### Input/Output File Characteristics

When using delimited EDI data routing, the input file to the data router is an expanded EDI data file. All fields have been expanded to their maximum lengths and all numerics have been formatted to the Gentran:Basic floating point notation by the Inbound Editor. The input file has a variable-block record format with a 32756-byte record length. The following output files must also have a variable-block record format with a 32756-byte record length:

DD Name	Description
TEMP	Temporary Data Work file
EDIRTE00	Non-Routed Data file
EDIRTE01-EDIRTE99	Output Route files

**Caution:** For every output file (EDIRTE01-EDIRTE99) you put in the JCL, you *must* have a corresponding Output File Record parameter in the format selection and conditional expressions. However, no conditional expressions are required. This parameter is required to open and close the output file.

### Routing Control

The level of routing is predetermined based on the type of EDI data. The ROUTING LEVEL parameter record does not apply when routing delimited EDI data. For ANA TRADACOMS, the default routing level is by interchange (transmission). This is due to the logical documents used within the ANA Standards. With a routing level by interchange, all conditional expressions are evaluated at the end of each interchange. All other standards default to a routing level by transaction. At this level, all conditional expressions are evaluated at the end of each transaction.

### Non-Delimited “Fixed Format” EDI Data Routing (Gentran:Structure only)

**Note:** Non-delimited EDI data refers to the following types of data: GENCOD, COMPORD, NSF, and UB92, or any other fixed format types.

The Data Router (EBDI095) program is the only split and/or routing process available for non-delimited EDI data. The data router is intended to run after the Inbound Structure Preprocessor (EBDI083) program, a Gentran:Structure program. Delimited and non-delimited EDI data cannot be mixed as input to the data router. The output of the data router is directly processed by the Inbound Mapping (EBDI041) program.

### Input/Output File Characteristics

The input to the data router is the output from the Structure Preprocessor program. This file has a variable block record format with 32756-byte records. All fields are at fixed locations within the non-delimited EDI data record. The following output files must also have a variable-block record format with 32756-byte record:

DD Name	Description
TEMP	Temporary Data Work file
EDIRTE00	Non-Routed Data file
EDIRTE01-EDIRTE99	Output Route files

**Caution:** For every output file (EDIRTE01-EDIRTE99) you put in the JCL, you *must* have a corresponding Output File Record parameter in the format selection and conditional expressions. However, no conditional expressions are required. This parameter is required to open and close the output file.

*Routing Control*

The default routing level is by transaction. The ROUTING LEVEL parameter record can be used. When there is routing control by interchange, interchange headers must be present in the input data. When there is routing control by transaction, transaction headers must be present in the input data.

**Non-Delimited EDI Data Control Headers**

<b>Level</b>	<b>Control Headers</b>
Interchange	*BI, #BI, BI
Group	*BG, #BG, BG
Transaction	*BT, #BT, BT

**Non-Delimited EDI Data Control Trailers**

<b>Level</b>	<b>Control Trailers</b>
Interchange	*EI, #EI, EI
Group	*EG, #EG, EG
Transaction	*ET, #ET, ET

For interchange-level routing, conditional expressions are evaluated at the end of each interchange. For transaction-level routing, conditional expressions are evaluated at the end of each transaction.

## Data Router Parameters

**EBDI095**

### Purpose

Using the standard 72-byte parameter cards either within the job flow or with a file, the input parameters for the Data Router (EBDI095) program are listed in this section.

#### *ROUTING TYPE Record – All Routing Types (Mandatory)*

The following table describes the ROUTING TYPE record parameter layout:

Position	Field Content	COBOL Picture	Format	Length	Description
1 – 12	ROUTING TYPE	X(12)	AN	12	Indicates type of data being used as input. Valid value: ROUTING TYPE
13 – 13	Filler	X(01)	AN	01	Blank
14 – 26	Input Data Type	X(13)	AN	13	Valid values: DELIMITED NON-DELIMITED APPLICATION

#### *SEGMENT ID START Record – Non-Delimited EDI and Application Routing Only (Mandatory)*

The following table describes the SEGMENT ID START record parameter layout:

Position	Field Content	COBOL Picture	Format	Length	Description
1 – 16	SEGMENT ID START	X(16)	AN	16	Identifies start position of the Segment ID within all input records. Valid value: SEGMENT ID START' <b>Note:</b> For application routing, Segment ID may also be known as <i>record type</i> .
17 – 17	Filler	X(01)	AN	01	Blank
18 – 22	Starting Position	9(05)	N0	05	Segment ID starting position. Valid values: +1 through +32742

**Note:** For non-delimited EDI data, use the segment ID starting position before the data runs through (EBDI083) program. The Structure Preprocessor appends a sequence number to the beginning of each EDI record. The data router takes that sequence number into account when locating the Segment ID.

**SEGMENT ID LENGTH Record – Non-Delimited EDI and Application Routing Only (Mandatory)**

The following table describes the SEGMENT ID LENGTH record parameter layout:

Position	Field Content	COBOL Picture	Format	Length	Description
1 – 17	SEGMENT ID LENGTH	X(17)	AN	17	Identifies length of the segment ID within all input records. Valid value: SEGMENT ID LENGTH
18 – 18	Filler	X(01)	AN	01	Blank
19 – 20	Segment ID length	9(02)	N0	02	Segment ID length number. Valid values: <b>01 – 10.</b>

**Application Headers**

Level 1, Level 2, and Level 3 record types can have 10 record types associated with each level. These record types can be nested in three distinct levels within the application file. Level 1, Level 2, and Level 3 HEADER RECORD TYPE parameter records can be repeated 10 times within the parameter file, each with a record type.

**LEVEL 1 HEADER RECORD TYPE Record — Application Routing Only (Mandatory)**

Position	Field Content	COBOL Picture	Format	Length	Description
01 – 26	LEVEL 1 HEADER RECORD TYPE	X(26)	AN	26	Application record type that indicates new first level of document. Valid value: LEVEL 1 HEADER RECORD TYPE'
27 – 27	Filler	X(01)	AN	01	Blank
28 – 37	Record Type	X(10)	AN	10	Record type value

**LEVEL 2 HEADER RECORD TYPE Record — Application Routing Only (Optional)**

Position	Field Content	COBOL Picture	Format	Length	Description
01 – 26	LEVEL 2 HEADER RECORD TYPE	X(26)	AN	26	Application record type that indicates new second level of document. Valid value: LEVEL 2 HEADER RECORD TYPE'
27 – 27	Filler	X(01)	AN	01	Blank
28 – 37	Record Type	X(10)	AN	10	Record type value

**LEVEL 3 HEADER RECORD TYPE Record — Application Routing Only (Optional)**

Position	Field Content	COBOL Picture	Format	Length	Description
01 – 26	LEVEL 3 HEADER RECORD TYPE	X(26)	AN	26	Application record type that indicates new third level of document. Valid value: LEVEL 3 HEADER RECORD TYPE
27 – 27	Filler	X(01)	AN	01	Blank
28 – 37	Record Type	X(10)	AN	10	Record type value

**Application Trailers**

Three unique application record trailer levels can be identified. These trailers are associated with each application header. Nesting of application records can be ended with these record types. Level 1, Level 2, Level 3 TRAILER RECORD TYPE parameter records can be used once within each parameter file, each with a record type.

**LEVEL 1 TRAILER RECORD TYPE Record — Application Routing Only (Optional)**

Position	Field Content	COBOL Picture	Format	Length	Description
01 – 27	LEVEL 1 TRAILER RECORD TYPE	X(27)	AN	27	Application record type that indicates end of first level of document. Valid value: LEVEL 1 TRAILER RECORD TYPE
28 – 28	Filler	X(01)	AN	01	Blank
29 – 38	Record Type	X(10)	AN	10	Record type value

**LEVEL 2 TRAILER RECORD TYPE Record — Application Routing Only (Optional)**

Position	Field Content	COBOL Picture	Format	Length	Description
01 – 27	LEVEL 2 TRAILER RECORD TYPE	X(27)	AN	27	Application record type that indicates end of second level of document. Valid value: LEVEL 2 TRAILER RECORD TYPE
28 – 28	Filler	X(01)	AN	01	Blank
29 – 38	Record Type	X(10)	AN	10	Record type value

**LEVEL 3 TRAILER RECORD TYPE Record — Application Routing Only (Optional)**

Position	Field Content	COBOL Picture	Format	Length	Description
01 – 27	LEVEL 3 TRAILER RECORD TYPE	X(27)	AN	27	Application record type that indicates end of third level of document. Valid value: LEVEL 3 TRAILER RECORD TYPE
28 – 28	Filler	X(01)	AN	01	Blank
29 – 38	Record Type	X(10)	AN	10	Record type value

**Routing Level Record — Non-delimited EDI Data Routing Only (Optional)**

The following table describes the Non-Delimited ROUTING LEVEL record.

Position	Field Content	COBOL Picture	Format	Length	Description
01 – 13	ROUTING LEVEL	X(13)	AN	13	Route level control card for control when input data is routed to the output files. Valid value: ROUTING LEVEL
14 – 14	Filler	X(01)	AN	01	Blank
15 – 25	Routing Level	X(11)	AN	11	Valid values: TRANSACTION (default) INTERCHANGE

**Note:** This parameter applies only to non-delimited EDI data. ROUTING LEVEL INTERCHANGE evaluates all conditional expressions at the ends of interchanges. Interchange headers must be present in data for ROUTING LEVEL INTERCHANGE. ROUTING LEVEL TRANSACTION evaluates all conditional expressions at the ends of transactions. ROUTING LEVEL TRANSACTION is the default routing level. Transaction headers must be present in data for ROUTING LEVEL TRANSACTION.

*Routing Level Record — Application Data Routing Only (Optional)*

The following table describes the Application Data ROUTING LEVEL record.

Position	Field Content	COBOL Picture	Format	Length	Description
01 – 13	ROUTING LEVEL	X(11)	AN	13	Route level control card for control when input data is routed to the output files. Valid value: ROUTING LEVEL
14 – 14	Filler	X(01)	AN	01	Blank
15 – 25	Routing Level	X(01)	AN	01	Valid values: 1 – similar to interchange (default) 3 – similar to transaction

*Input Record Format Record — Application Routing Only (Mandatory)*

The following table describes the INPUT RECORD FORMAT record.

Position	Field Content	COBOL Picture	Format	Length	Description
01 – 19	INPUT RECORD FORMAT	X(19)	AN	19	Input file physical record format. Valid value: INPUT RECORD FORMAT
20 – 20	Filler	X(01)	AN	01	Blank
21 – 23	Record Format	X(03)	AN	03	Valid values: (V) – Variable (F) – Fixed

*Input Record Length Record — Application Routing Only (Mandatory)*

The following table describes the INPUT RECORD LENGTH record.

Position	Field Content	COBOL Picture	Format	Length	Description
01 – 13	RECORD LENGTH	X(13)	AN	13	Input record length indicator. Valid value: RECORD LENGTH
14 – 14	Filler	X(01)	AN	01	Blank
15 – 19	Record Length	9(05)	N0	05	Valid values: 1 – 32760.



**Output File Record — All Routing Types (Mandatory)**


The following table describes the OUTPUT FILE record.

Position	Field Content	COBOL Picture	Format	Length	Description
01 – 11	OUTPUT FILE	X(11)	AN	11	Output file number associated with conditional expressions. Valid value: OUTPUT FILE
12 – 12	Filler	X(01)	AN	01	Blank
13 – 14	Output file number	9(02)	N0	02	File number for data routing. Valid values: <b>01 – 99</b> .

**Caution:** For every output file (EDIRTE01-EDIRTE99) you put in the JCL, you *must* have a corresponding Output File Record parameter in the format selection and conditional expressions. However, no conditional expressions are required. This parameter is required to open and close the output file.

**Conditional Expression Record — All Routing Types (Mandatory)**

The following table describes the CONDITIONAL EXPRESSION record.

Position	Field Content	COBOL Picture	Format	Length	Description
1 – 80	Conditional Expression	X(80)	AN	80	Condition expression information for identification of elements involved with data routing. Consists of:  <div style="text-align: center;">  </div>

**Conditional Expression And/or Record — All Routing Types (Optional)**

The following table describes the CONDITIONAL EXPRESSION AND/OR record.

Position	Field Content	COBOL Picture	Format	Length	Description
1 – 80	AND, OR	X(80)	AN	80	Free format logic operand value used between conditional expressions. Valid values: AND, OR

## Data Routing

EBDI095

### Purpose

The Data Router (EBDI095) program can take an application data file as input and route specific documents to designated router output files. It is intended to run after the Inbound Mapper (EBDI041) program or before the Outbound Mapper (EBDI042) program.

### Input/Output File Characteristics

When using application data routing, the input data can have either a fixed or a variable record format. Fixed and variable data cannot be mixed as the input to a single run of the data router. The input file must contain more than one application record type. The data router will fail if only one record type is present in the entire input file. The following output files must have the same characteristics as the input application file:

DD Name	Description
TEMP	Temporary Data Work file
EDIRTE00	Non-Routed Data file
EDIRTE01-EDIRTE99	Output Route files

**Caution:** For every output file (EDIRTE01-EDIRTE99) you put in the JCL, you *must* have a corresponding Output File Record parameter in the format selection and conditional expressions. However, no conditional expressions are required. This parameter is required to open and close the output file.

### Routing Control

The application record(s) designating the beginning of a document or document grouping must be defined to the data router. This is done with the LEVEL # HEADER RECORD TYPE parameters. Three levels of application record *nesting* can be defined to the data router. Ten different application record types can be associated with each of the three header record levels.

An example of application record nesting is the instance of one application record defining the beginning of a batch of purchase orders and another application record defining the beginning of a specific purchase order. When each specific purchase order is routed to its designated route file, the application record defining the purchase order batch must precede the specific purchase order application record(s). This is an example of two levels of application record nesting.

The LEVEL 1 HEADER RECORD TYPE parameter must be specified to define the beginning of a new document or document grouping. The LEVEL 2 HEADER RECORD TYPE and the LEVEL 3 HEADER RECORD TYPE parameters are needed only if application record nesting occurs in the application file.

Trailer record types can be associated with each nesting of application header record types. Trailer record types are optional. The LEVEL # TRAILER RECORD TYPE parameters can be used to finish nested application record loops.

## Files Used

### Input

The following files are used as input:

Filename	Description
ERRCTL	Error Message file This file contains descriptions for every error number within Gentran:Basic.
SYS003	Input file This file contains the variable format, fixed format, or application data for routing.
TEMP	Temporary Data Work file This file is used as a temporary repository for EDI data while routing conditions are evaluated within the input. The file characteristics of this file must match the file characteristics of the input file (SYS003).
SYS030	Input Parameter file This file supplies the Data Router program with format selection and conditional expressions used in routing data to the output files.
SYS031	Mapping Code Validation file This file can be used within conditional expressions for checking the presence of an individual code with a code validation table.
EDICFG	System Configuration file

### Output

The following files are used as output:

Filename	Description
TEMP	Temporary Data Work file This file is used as a temporary repository for EDI data while routing conditions are evaluated within the input. The file characteristics of this file must match the file characteristics of the input file (SYS003).
SYS006	Summary Report This report contains summary confirmation of input parameters used by the data router.
SYS010	Error Report This report contains error information that the router detected in processing.
EDIRTE00	Non-Routed Data file This file is used as a "slide" file that contains data that was not routed to any other output file. The file characteristics of this file must match the file characteristics of the input file (SYS003).
EDIRTE01 through EDIRTE99	Output Route files These files contain the data directed to them by user-defined conditional expressions within the Input Parameter file. The number of files used depends on the conditional expressions used. The file characteristics of these files must match the file characteristics of the input file (SYS003).

## Defining the Rules for the Split

### Purpose

Conditional expressions testing the value of elements or fields in the input data determine to which Route file a document is written. Routing should be based on unique information within the input file. If the element(s) or field(s) used in the conditional expression(s) can occur multiple times in the document, every occurrence of the element(s) or field(s) is reviewed and if any meet the criteria specified in the conditional expression(s) the data is routed to the requested file.

### Conditional Expression Operand 1 Formats

#### Delimited EDI Data Routing

Segment ID/Element Number

This applies to segments using simple elements, without the use of component element separators. This applies to most X12, TDCC, UCS US DOMESTIC Standards.

Example

N1/1 EQ 'ST'

Segment ID/Element Number/Subelement Number

This applies to segments using composite elements, using the component element separators. This applies to recent X12, EDIFACT, TRADACOMS Standards.

Example

RFF/02/01 EQ '12345'

#### Non-Delimited EDI Data and Application Data Routing

Segment ID/Element Start/Element Length

The element start position and element length apply to non-delimited EDI data and application input only. Element start position can range from 1 to 32742. Element length can be up to 80 bytes, but operand 1 only evaluates the first 20 characters or 18 digits.

Example

HO1/5/3 EQ '001'

### Conditional Expression Operator Values

Operator	Meaning
EQ	Equals
NE	Not equals
LT	Less than
LE	Less than or equals
GT	Greater than
GE	Greater than or equals
VT	Within Code Validation table
NT	Not within Code Validation table

### Conditional Expression Operand 2 Format

'BILL NUMBER'	Alphanumeric literal	20 characters
33333.00	Numeric literal	18 digits, decimal, sign
'testtable'	Code Validation Table name	10 characters

Element length is based on a 20-character maximum for alphanumeric literals or an 18-digit maximum for numeric literals, including signs and decimal points. Data validation table names have a 10-character maximum.

See the topic "Data Validation Screen" in Chapter 6 of the *Gentran:Basic for zSeries Release 6.4 User's Guide* for information on how to set up the data validation table.

### Conditional Expression Numeric Check option

The value NUMERIC used in combination with OPERAND 2 numeric literals uses internal numeric testing on OPERAND 1 and OPERAND 2. This option applies only to delimited EDI format data.

### Compound Conditional Expressions

When using conditional expressions, the OUTPUT FILE record starts the conditional expression parameters. Conditional expressions can be compounded together by using the Conditional Expression AND/OR record. Compound conditional expressions cannot be mixed. Conditional Expression AND/OR records are associated with the same OUTPUT FILE record.

OUTPUT FILE 01

[OPERAND 1] [OPERATOR] [OPERAND 2] {NUMERIC}

OUTPUT FILE 50

[OPERAND 1] [OPERATOR] [OPERAND 2] {NUMERIC}

OR

[OPERAND 1] [OPERATOR] [OPERAND 2] {NUMERIC}

OUTPUT FILE 99

[OPERAND 1] [OPERATOR] [OPERAND 2] {NUMERIC}

OR

[OPERAND 1] [OPERATOR] [OPERAND 2] {NUMERIC}

Operand number 1 describes the standards element or application field involved with condition expression. Operand number 2 describes the alphanumeric literal, numeric literal, or code validation table name.

The operator describes the logic or code validation operation used between Operand number 1 and Operand number 2.

The numeric check is optional and is required only when using numeric literals. The numeric check can be used when routing delimited EDI data only. Operand 1 and Operand 2 are truncated at the first 20 characters for alphanumeric literals, and at the numeric precision of 18 digits, including signs and decimal points, for numeric literals.

### Conditional Expression Formats

Conditional expressions have three formats. Format 1, literal format, is used when comparing elements with alphanumeric literals. Format 2, numeric format, is used when comparing standard element with numeric values. Format 3, Code Validation, is used when comparing standard element with code validation table values.

#### Format 1 — Literal Format

##### ANSI, TDCC Example

```
OUTPUT FILE 99
  M13/04 EQ 'BILL NUMBER'
  AND
  M13/06 NE '10'
```

Format 1 tests the M13/04 element and M13/06 element for being equal to the alphanumeric literal BILL NUMBER and not equal to alphanumeric literal 10, respectively. If both of these are true, transaction is sent to output file 99.

##### EDIFACT Example

```
OUTPUT FILE 25
  RFF/01 GT 'LA '
```

Format 1 tests the RFF/01 element for characters greater than alphanumeric literal 'LA '. If this is true, the transaction is sent to output file 25.

```
OUTPUT FILE 26
  RFF/02/01 EQ 'PO123456'
```

Format 1 tests the RFF/02/01 component element for characters equal to the alphanumeric literal 'PO123456'. If this is true, the transaction is sent to output file 26.

##### TRADACOMS Example

```
OUTPUT FILE 13
  DIN/03/02 GT '1200'
```

Format 1 tests the DIN/03/02 component element for characters greater than alphanumeric literal '1200'. If this is true, the interchange is sent to output file 13.

##### GENCOD Example

```
SEGMENT ID START 01
SEGMENT ID LENGTH 03
OUTPUT FILE 27
  100/04/13 GE '3000000000000'
```

Format 1 tests the 100 segment, starting in position 16 for a length of 13 positions, for characters greater than or equal to the alphanumeric literal '3000000000000'. If this is true, the transaction or interchange is sent to output file 27.

### *Format 2 — Numeric Format*

#### **ANSI, TDCC Example**

```
OUTPUT FILE 01
  M13/05 GE 33333.00 NUMERIC
  OR
  M13/05 EQ 30000    NUMERIC
```

Format 2 tests the 013/05 element. If the element is greater than or equal to 33333.00 or the element equals 30000, the document is sent to output file 01.

#### **TRADACOMS Example**

```
OUTPUT FILE 05
  OLD/01 GT 0.0 NUMERIC
```

Format 2 tests the OLD/01 element. If the element is greater than 0.0, the interchange is sent to output file 05.

### *Format 3 — Data Validation Format*

```
OUTPUT FILE 05
  M13/04 NT 'TESTTABLE'
  OR
  M13/06 VT 'TESTTABLE'
```

Format 3 tests the M13/04 and M13/06 elements against data validation table TESTTABLE. If the value in the M13/04 element is not found in TESTTABLE or the value in the M13/06 element is found in TESTTABLE, the transaction is sent to output file 05.

See the topic “Data Validation Screen” in Chapter 6 of the *Gentran:Basic for zSeries Release 6.4 User's Guide* for information on how to set up a data validation table.

## Error Message Change Audit Maintenance Program

**EBDI105**

The Error Message Change Audit Maintenance program (EBDI105) maintains the Error Message Change Audit file. The program functions include deleting, archiving, and restoring change audit information.

The Error Message change audit file contains records that track changes made to the Error Message records that reside on the Error Message and Control file. Update functions for Gentran messages and Add/Update functions for User messages (via the online Error Message subsystem) result in records being written to the Error Message Change Audit file. The Batch utility program EBDIT13 will also generate Error Message Change Audit records when error message records are added or replaced.

### Operation

The Error Message Change Audit Maintenance program operates in the following general modes:

#### **Periodic Housekeeping**

On a periodic basis (anywhere from daily to monthly depending on volume and retention requirements), the Housekeeping operation deletes and optionally archives data based on specified retention periods.

#### **Ad Hoc Maintenance (Purge and Restore)**

In addition to the regularly scheduled housekeeping runs, the Error Message Change Audit Maintenance Purge and Restore functions delete and optionally archive selected data and restore data from archives.

### Processing

The Error Message Change Audit Maintenance program processes documents using the following three operations:

- Housekeeping
- Purge
- Restore

#### *Housekeeping Operation*

If the Housekeeping operation is requested, Error Message Change Audit Maintenance deletes and optionally archives the change audit records that exceed the retention periods specified in the control cards. An option of the Housekeeping operation is to archive all selected change audit records to a sequential output file.

#### *Purge Operation*

If the Purge operation is requested, change records are removed if they match *all* criteria specified. An option of the Purge operation is to archive selected change audit records to a sequential output file.



## Restore Operation

If the Restore operation is requested, change audit records are restored from the Archive file to the Change Audit if the records match *all* selection criteria specified.

## Files Used

### Input

Control cards and the Error Message Change Audit files are the principal input to Security Change Audit Maintenance.

Filename	Description
ERRARCH	Error Message Change Audit Archive File
EDIECHA	6.4 VSAM Error Message Change Audit File
EDICNTL	Control Card File
ERRCTL	Error Message and Control File
EDICFG	System Configuration File

### Output

The following files are used as output:

Filename	Description
ERRARCH	Error Message Change Audit Archive File
EDIECHA	6.4 VSAM Error Message Change Audit File

### Reports

Program EBDI105 produces the following reports:

Report Name	Description
ERRAUDT	Error Message Change Audit Maintenance Report
EDILOG	Error Message Change Audit Processing Log
EDISUM	Error Message Change Audit Summary Report

## Control Card Options

### Operations

- HOUSEKEEPING (Default Operation)
- PURGE
- RESTORE

**Selection Criteria****Defaults**

For Housekeeping operation:

RETENTION-DAYS

For Purge and Restore operations:

FROM-UPDATE-DATE 01/01/1900

FROM-UPDATE-TIME 00:00

TO-UPDATE-APPLIED-DATE 12/31/2099

TO-UPDATE-APPLIED-TIME 23:59

MESSAGE-NUMBER ALL

RECORD-TYPE ALL

For Purge and Housekeeping operations:

ARCHIVE NO

**Valid Criteria Values****Archive**

This controls whether an archive is to be created for deleted change records.

NO

YES

**From-Update-Date**

Specifies the date that the update was applied to the Error Message record. Starts the range. Used with To-Update-Applied-Date.

MM/DD/YYYY

**From-Update-Time**

Specifies the time that the update was applied to the Error Message record. Starts the range. Used with To-Update-Applied-Time.

HH:MM

**Message-Number**

A 5-character identification used to restrict maintenance by Error Message number.

ALL

XXXXX

**Record-Type**

A 1-character identification used to identify the Message number as either a Gentran message or a User message. With Message-Number this field will restrict maintenance.

**G** Gentran Message

**U** User Message

**Retention-Days**

Specifies the number of days that have elapsed since the update was applied.

000 – 999

**To-Update-Applied-Date**

Specifies the date that the update was applied to the Error Messages. Ends the range. Used with From-Update-Date.

MM/DD/YYYY

**To-Update-Applied-Time**

Specifies the time that the update was applied to the Error Messages. Ends the range. Used with From-Update-Time.

HH:MM

## Error Message Maintenance

EBDIT13

### Purpose

Error Message Maintenance (EBDIT13) program is a batch program and is executed using JCL member UPDTMSG. The Error Message Control file must be closed to CICS before running this job.

**Note:** We recommend that you back up your VSAM Error Message Control file.

### Process Control

The Error Message Maintenance program performs two functions: add and update. An add operation occurs when the input ERRSEQ file does not find a matching record in the ERRCTL file. An update operation occurs when the input ERRSEQ file matches an existing record in the ERRCTL file.

The update function only occurs on five fields if there is a difference in the value. The fields updated are: Error type, Direction, Severity, Text, and Note codes. The Audit Summary report identifies before and after images on updates and new records added.

### Files Used

#### Input

The following files are used as input:

Filename	Description
ERRSEQ	Sequential Cumulative Fix Error Message and Control file
EDICFG	System Configuration file

#### Input/Output

The following file is used as input and output:

Filename	Description
ERRCTL	Error Message and Control file (VSAM)

#### Report

The EBDIT13 program produces the following report:

Filename	Description
SUMRPT	Error Message and Control file Audit Summary Report

Example Report

Figure 4.6 is an example of the EBDIT13 program output report.

EBDIT13	DATE: 12/01/2005	TIME: 12:00	ERROR MESSAGE AND CONTROL FILE AUDIT SUMMARY REPORT	PAGE: 1
Error #	Field Name	(B)efore (A)fter (N)ew Msg	Values	
00030	ERROR TYPE	N:	G	
	DIRECTION	N:		
	SEVERITY	N:	I	
	TEXT	N:	FOUND INDICATED NUMBER OF SEGMENTS. DOES NOT AGREE WITH COUNT IN TRANSACTION.	
	NOTE CODES	N:	000112	325 228
00031	ERROR TYPE	N:	V	
	DIRECTION	N:	B	
	SEVERITY	N:	W	
	TEXT	N:	MONTH GREATER THAN 12 OR LESS THAN 1 IN DATE INPUT	
	NOTE CODES	N:	000	225 218
00032	ERROR TYPE	B:	V	
		A:	G	
	DIRECTION	B:	B	
		A:		
	SEVERITY	B:	W	
		A:	F	
TOTAL RECORDS READ.....				2626
TOTAL RECORDS ADDED.....				2
TOTAL RECORDS UPDATED.....				1
* * NORMAL END OF JOB * * *				

Figure 4.6 EBDIT13 Program Output Report

### Field Definitions

**Error #**

This field displays the error code number associated with the error message.

**Field Name:**

This field identifies the field name from the error message and control file.

**Error Type**

<b>D</b>	Databank
<b>E</b>	Editor
<b>G</b>	General
<b>M</b>	Mapping
<b>P</b>	Gentran:Plus
<b>S</b>	Gentran:Structure
<b>V</b>	Gentran:Viewpoint

**Direction**

<b>B</b>	Message is used in both inbound and outbound processes.
<b>I</b>	Message is used in inbound process only.
<b>O</b>	Message is used in outbound processes only.
<b>SP</b>	Message is not used inbound or outbound.

**Severity**

<b>I</b>	Informational Message
<b>W</b>	Warning Only
<b>E</b>	Error
<b>S</b>	Severe Error
<b>F</b>	Fatal Error

**Text**

Descriptive information for the error.

**Note Codes**

Three-byte numeric values used during acknowledgment generation for reporting compliance errors.

<b>B</b>	This field indicates the data value before an update.
<b>A</b>	This field indicates the data value after an update.
<b>N</b>	This field indicates the data value is new due to an add.

**Value**

This field displays the data value for the field name indicated.

**Total Records Read**

This field displays the total input sequential error control message records read.

**Total Records Added**

This field displays the total records added to the VSAM error control and message file.

**Total Records Updated**

This field displays the total number of updated records to the VSAM error control and message file.

---

## Global Parameter Change Audit Maintenance Program

**EBDI103**

The Global Parameter Change Audit Maintenance program (EBDI103) maintains the Global Parameter Change Audit file. The program functions include deleting, archiving, and restoring change audit information.

The Global Parameter change audit file contains records that track changes made to the Global Parameter records that reside on the System Configuration file. Update functions via the online Global Parameter subsystem result in records being written to the Global Parameter Change Audit file.

### Operation

The Global Parameter Change Audit Maintenance program operates in the following general modes:

#### **Periodic Housekeeping**

On a periodic basis (anywhere from daily to monthly depending on volume and retention requirements), the Housekeeping operation deletes and optionally archives data based on specified retention periods.

#### **Ad Hoc Maintenance (Purge and Restore)**

In addition to the regularly scheduled housekeeping runs, the Global Parameter Change Audit Maintenance Purge and Restore functions delete and optionally archive selected data and restore data from archives.

### Processing

The Global Parameter Change Audit Maintenance program processes documents using the following three operations:

- Housekeeping
- Purge
- Restore

#### *Housekeeping Operation*

If the Housekeeping operation is requested, Global Parameter Change Audit Maintenance deletes and optionally archives the change audit records that exceed the retention periods specified in the control cards. An option of the Housekeeping operation is to archive all selected change audit records to a sequential output file.

#### *Purge Operation*

If the Purge operation is requested, change records are removed if they match *all* criteria specified. An option of the Purge operation is to archive selected change audit records to a sequential output file.

### Restore Operation

If the Restore operation is requested, change audit records are restored from the Archive file to the Change Audit if the records match *all* selection criteria specified.

### Files Used

#### Input

Control cards and the Global Parameter Change Audit files are the principal input to Global Parameter Change Audit Maintenance.

Filename	Description
GBLARCH	Global Parameter Change Audit Archive File
EDIGCHA	6.4 VSAM Global Parameter Change Audit File
EDICNTL	Control Card File
ERRCTL	Error Message and Control File
EDICFG	System Configuration File

#### Output

The following files are used as output:

Filename	Description
GBLARCH	Global Parameter Change Audit Archive File
EDIGCHA	6.4 VSAM Global Parameter Change Audit File

#### Reports

Program EBDI103 produces the following reports:

Report Name	Description
GBLAUDT	Global Parameter Change Audit Maintenance Report
EDILOG	Global Parameter Change Audit Processing Log
EDISUM	Global Parameter Change Audit Summary Report

### Control Card Options

#### Operations

- HOUSEKEEPING (Default Operation)
- PURGE
- RESTORE



*Selection Criteria***Defaults**

For Housekeeping operation:

RETENTION-DAYS

For Purge and Restore operations:

FROM-UPDATE-DATE 01/01/1900

FROM-UPDATE-TIME 00:00

TO-UPDATE-APPLIED-DATE 12/31/2099

TO-UPDATE-APPLIED-TIME 23:59

RECORD-TYPE ALL

For Purge and Housekeeping operations:

ARCHIVE NO

*Valid Criteria Values***Archive**

This controls whether an archive is to be created for deleted change records.

NO

YES

**From-Update-Date**

Specifies the date that the update was applied to the Global Parameter record. Starts the range. Used with To-Update-Applied-Date.

MM/DD/YYYY

**From-Update-Time**

Specifies the time that the update was applied to the Global Parameter record. Starts the range. Used with To-Update-Applied-Time.

HH:MM

**Record-Type**

A 10-character identification used to restrict maintenance by Global Parameter Record type.

ALL

GI Inbound Global parameters

GO Outbound Global Parameters

**Retention-Days**

Specifies the number of days that have elapsed since the update was applied.

000 – 999

**To-Update-Applied-Date**

Specifies the date that the update was applied to the Global Parameter record. Ends the range. Used with From-Update-Date.

MM/DD/YYYY

**To-Update-Applied-Time**

Specifies the time that the update was applied to the Global Parameter record. Ends the range. Used with From-Update-Time.

HH:MM

## JCL File Maintenance

## EDIRJCLX

The JCL File Maintenance program (EDIRJCLX) allows you to add, replace, get, and delete JCL from the EDIRJCL file. The JCL that resides on this file is used when submitting batch jobs from the Gentran online subsystem. The Gentran:Basic Online Batch Initiator (EDIBOBI) or the Gentran:Control/Gentran:Realtime Online Batch Initiator (EDIEOBI) submit the JCL for processing.

### Operation

Control cards enable the JCL File Maintenance program to add new JCL statements, replace existing JCL statements, get existing JCL statements, and delete existing JCL statements from the EDIRJCL file. The sample JCL to run this program is located in member EXECJCLX in the JCL library.

- When using the **Add** function, EDIRJCLX will read the JCL in the SEQJCL file and add the specified JCL to the EDIRJCL file.
- When using the **Replace** function, EDIRJCLX will read the JCL in the SEQJCL and use it to replace the specified JCL in the EDIRJCL file.
- When using the **Get** function, EDIRJCLX will read the EDIRJCL file and write the specified JCL to the SEQJCL file.
- When using the **Delete** function, EDIRJCLX will delete the specified JCL from the EDIRJCL file.

### Inputs

The following are the inputs for the JCL file maintenance program:

DDname	Filename
EDICFG	System Configuration file
ERRCTL	Error message and control file
PARMS	Control Card file
SEQJCL	Sequential file that contains JCL to be added or replaced
EDIRJCL	Online JCL file

### Outputs

The following are the outputs for the JCL file maintenance program:

DDname	Filename
SEQJCL	Sequential JCL file that will contain JCL pulled from the EDIRJCL file (when using the GET option)
EDIRJCL	Online JCL file

## Reports

The following are the reports for the JCL file maintenance program:

<b>DDname</b>	<b>Filename</b>
EDISUM	Processing Summary report
EDILOG	Processing Log report

### *Processing Summary Report*

The Process Summary report is contained in the EDISUM output file. The report displays the processing results for the execution, the number of records read, and the return code for the execution.

Figure 4.7 illustrates the Processing Summary Report.

```

PAGE : 00001
VERSION: 6.4

GENTRAN: BASIC
JCL FILE MAINTENANCE
SUMMARY REPORT

REPORT DATE: 12/01/2005
REPORT TIME: 12:00:00
REPORT ID : EDIRJCL-EDISUM

OPTIONS USED THIS RUN
-----
REQUESTED-OPERATION = ADD
JCL-NAME             = EXEC0060
RECORDS ADDED       :          168

PROCESSING SUMMARY
-----
TOTAL RECORDS DELETED      :          0
TOTAL RECORDS ADDED       :          168
TOTAL RECORDS RETRIEVED   :          0
TOTAL RECORDS REPLACED    :          0
NUMBER OF ERRORS THIS RUN :          0
HIGHEST RETURN CODE THIS RUN :          0
    
```

Figure 4.7 Processing Summary Report

*Processing Log Report*

The Process Log report is contained in the EDILOG output file. The report contains information and error messages; it is created by the subroutine EDILOG, which is included in the Gentran:Basic system.

Figure 4.8 illustrates the Processing Log Report.

```

                                PAGE : 00001
                                VERSION: 6.3
                                COMPILE DATE: 12/01/05

                                GENTRAN: BASIC
                                JCL FILE MAINTENANCE
                                PROCESSING LOG

REPORT DATE: 12/01/2005
REPORT TIME: 12:05:38
REPORT ID : EDIRJCL-EDILOG
MESSAGES
-----
EDI-010608-I 00 JCL PROGRAM MAINTENANCE PROCESSING BEGINS . . . DATE: 12/01/2005, TIME: 12:05:39
EDI-010609-I 00 JCL PROGRAM MAINTENANCE PROCESSING ENDS . . . DATE: 12/01/2005, TIME: 12:05:39

```

**Figure 4.8 Processing Log Report**

## Control Card Options

Control cards specify the operation to be completed when the job processes and the name of the JCL as it is stored on the EDIRJCL file.

Position	Length	Format	Field Name	Value
01 – 03	03	AN	Operation	ADD DEL REP GET
04 – 04	01	AN	Filler	spaces
05 – 12	08	AN	JCL Name	8-character name to identify the JCL
13 – 80	68	AN	Filler	spaces

**Note:** For queue file processing in Gentran:Realtime or Gentran:Control, the JCL name (in columns 5 – 12) should use the following convention:

**EDIJnnn**

Where:

**EDI** is the system image.

**J** is a constant.

**nnn** is a three-character name that is entered on the Queue Options Maintenance screen in the Batch JCL name field.

## Mapping Version Comparison and Conversion Report

**EBDI088**

### Purpose

The Mapping Version Comparison and Conversion Report (EBDI088) program takes an existing map and converts it into a new map to be used in translating EDI data of a more current standard. Almost all the mapping that was done in the old map is copied to the new map. A parameter record identifies the old Translation ID to copy from, the new Translation ID to copy to, and the version you want to convert to. Optional Loop ID Conversion parameter records identify differences in Loop IDs in the existing map versus the new standard. The program generates a report that displays all fields that were updated in the Transaction Segment and Element records (it shows what the field value was both before and after the change).

This program can be run two ways. The first converts a map to resemble the new version of the standard. This includes moving segments and elements to where they are located in the transaction set under the new version. New elements are added to the map if they do not exist in the old map. This first method should be run before trying the second method. The second method (Straight Copy Option) copies each segment in a map to the new map, as long as the segment still exists in the new standard. The order of the segments does change although they may be different in the new standard. If any segments changed their position in the transaction set, the segments in the map have to be rearranged using the online Copy Segments from Transaction screen. However, the elements in the map are moved to the new version, and new elements are added to the map if they do not exist in the old map. To run the second method, enter **Y** in position 35 of the parameter record.

The following fields are used to match the Transaction Segment records to the Standard Segment records: Transaction Segment ID, Type of Segment, and Transaction Loop ID. The Transaction Element Reference field in the transaction element record layout is used to match the standard element records to the transaction element records.

The conditions for specifying the optional Loop ID Conversion parameter records can be determined by evaluating the Loop IDs in the existing map and the new standard being converted to. If no differences occur, Loop ID Conversion parameter records are not required. If differences occur, Loop ID Conversion parameter records should be created to identify the relationship of the Loop IDs in the existing map to the new standard.

If you have Change Audit for Transaction enabled on the "1" record of the EDICFG file, a new record will be written to the Transaction Change Audit file (EDITCHA) for the transaction map that is created.

The following files are updated by this program in the manner indicated:

**An Application/Translation Link record is added after the new Transaction Header record has been added to the Transaction Header file.**

**All fields in the new Transaction Header record are copied from the old Transaction Header record with the exception of:**

Translation ID	Taken from parameter record.
Standard Version	Taken from parameter record.
Agency	Taken from parameter record.

**Note:** This report is not designed for EDIFACT to ANSI conversion.



**All fields in the new Transaction Segment record are copied over from the old Transaction Segment record with the exception of:**

Translation ID	Taken from parameter record.
Loop ID	Updated with the value specified in the optional Loop ID Conversion parameter record. A straight copy is performed if Loop ID Conversion is not specified.
Segment Sequence	The segment is resequenced.
Mandatory Code	Update with value found on Standard Segment file.
Description	Update with value found on Standard Segment file.

**All fields in the new Transaction Element record are copied over from the old Transaction Element record with the exception of:**

Translation ID	Taken from the parameter record.
Element Sequence	The element is re-sequenced.
Segment Sequence	The segments are re-sequenced.
Mandatory Code	Update with value found on Standard Element file.
Element Reference	Update with value found on Standard Element file.
Element Type	Update with value found on Standard Dictionary file.
Min Length	Update with value found on Standard Dictionary file.
Max Length	Update with value found on Standard Dictionary file.
Description	Update with value found on Standard Element file.
Conditional &	The mapping fields in the Transaction Element Arithmetic record mapped with an element identification Mapping Fields N1/10/100) are updated if the segment (Inbound only) sequence or element sequence has changed.

**The following conditions are checked by the program and an appropriate message is written on the output report for each condition identified:**

- When a Loop ID to be converted is not found in the map, a warning message is printed at the beginning of the report identifying the parameter record that contained the Loop ID. This message is generated only when optional Loop ID Conversion is specified.
- When a Transaction Segment is not copied to the new map, a message is printed on the report identifying the segment that was not copied.
- If a Transaction Segment has been mapped and the position of that segment within its transaction set has changed, a message is printed on the report and the segment is copied to the Transaction Segment file with a new sequence number.
- If a Transaction Element has been mapped and the position of that element within its segment has changed, a message is printed on the report and the element is copied to the Transaction Element file with a new sequence number.
- When adding a Transaction Element that does not exist under the old Transaction Segment, a message is printed on the report identifying the new element and the new element is added to the Transaction Element file.
- When a Transaction Element is not copied to the new map, a message is printed on the report identifying the element that was not copied.

- If a Transaction Element has been mapped with a conditional value using the N1/10/100 format, and if this element identification has changed in the new map, a message is printed and the element is copied to the Transaction Element file with a new conditional element identification.
- If a Transaction Element has been mapped with a constant and it is an ID-type field, the Standard Code file is read to determine if the code is valid in the new version of the standard. A message is printed if the code is not found on the standard codes file and the element is copied to the Transaction Element file with the invalid code.
- The following fields are reported if there has been any change in the value: segment mandatory code, element mandatory code, element type, element minimum length, element maximum length, conditional and arithmetic mapping fields.

**Processing terminates if any of the following conditions are true:**

- The old Translation ID given in the parameter file does not exist on the Transaction Header file.
- The new Translation ID given in the parameter file does exist on the Transaction Header file.
- No translation segments exist under the old Translation ID.
- The version given in the parameter record and the transaction given on the Transaction Header file does not exist on the Standard Transaction file.
- All files were not successfully opened.
- An invalid return code is received on a READ or WRITE statement.
- Incomplete parameter record found.
- More than 50 Loop ID Conversion parameter records were specified.

**The following conditions may prevent a segment or element from being copied over to the new Translation ID:**

- The segment Loop ID changed in the standard and an optional conversion parameter record was not specified.
- One or more segments in the standard were moved to another area of the document (i.e., Header, Detail, or Summary).
- The element reference number in the standard was changed.
- A segment that exists in a loop structure may not get copied if it was not originally copied from the loop structure where it now resides.
- The segment or element no longer exists under the new version and transaction set.

**The following is a list of messages that may appear on the Batch Copy Transaction Report and their meanings.**

Warning: This Loop ID to be converted was not found in the map.        xxxxxxxxx  
The displayed Loop ID Conversion parameter was specified but not found in the map. This condition does not prevent the conversion from being completed. It may point out a potential problem.

**Mandatory Code field has changed.**

The mandatory code field on the Transaction Segment or Element record was updated with the mandatory code field on the Standard Segment or Element record.

**The position of this segment within its transaction has changed.**

The segment's position in the transaction mapping definition was changed in the conversion.

**Segment not copied from old translation – could not be match with segment under Standard Segment file.**

The segment was not copied to the new transaction map. It could not be matched with a segment on the Standard Segment file. If a segment is not copied, the user can use the online copy screen to add the segment(s) that were not originally copied.

**Min length field has changed.**

The minimum length field on the Transaction Element record was updated with the minimum length field on the Standard Dictionary record.

**Max length field has changed.**

The maximum length field on the Transaction Element record was updated with the maximum length field on the Standard Dictionary record.

**Element Type field has changed.**

The element type field on the Transaction Element record was updated with the element type field on the Standard Dictionary record.

**Conditional element identification has changed.**

An element was mapped with a conditional value using the (N1/10/400) format and the segment sequence or element sequence number of that element was changed, so the conditional element identification was changed to correctly identify the location of that element.

**Conditional element identification no longer valid. Segment no longer exists under new translation.**

This message appears if a segment was not copied over to the new map and somewhere in the new map a conditional value is used, pointing to that segment. The element record with the invalid conditional value is copied over to the new translation with the mapping fields unchanged, and the user must evaluate the element record and make proper changes. This message only appears on inbound maps.

**Conditional element identification no longer valid. Element no longer exists under new translation.**

This message appears if an element was not copied over to the new map and somewhere in the new map a conditional value is used, pointing to that element. The element record with the invalid conditional value is copied over to the new translation with the mapping fields unchanged, and the user must evaluate the element record and make proper changes. This message only appears on inbound maps.

**Code no longer exists on Standard Code file.**

This message only appears on inbound maps. If an element record has been mapped with a constant and it is an ID-type field, the Standard Code file is read to verify that the code still exists. This message appears if the code was not found on the Standard Code file. The element record is copied over with the invalid code and the user must evaluate the record and make proper changes.

**The position of this element within its segment has changed.**

The transaction element's position within its segment was changed in the conversion.

**Element not copied from old translation – could not be matched with element under Standard Element file.**

The transaction element could not be matched with an element on the Standard Element file and was not copied to the new transaction map. Usually the element cannot be matched with an element on the Standard Element file because the element reference number was changed in the standard or the element was dropped from the standard. If the element reference number was changed, it is copied over from the standard element file and the user has to re-enter the mapping fields. The following message are displayed if this is the case.

**Element copied from Standard Element file – could not identify element under old translation.**

The transaction element record was copied from the Standard Element file because it could not identify the element under the old transaction map. Usually an element is copied from the Standard Element file because either it is a new element or the element reference number was changed in the Standard. If the element reference number was changed, the user has to re-enter the mapping fields from the old transaction element record. The previous message is displayed along with this message if the element reference number changed on the standard.

**Invalid conditional element identification found – could not identify this element in the old translation.**

This message appears if a conditional element identification was found that did not have a valid location of a Transaction Element record. The element record with the invalid conditional value is copied over to the new translation with the mapping fields unchanged, and the user must evaluate the element record and make proper changes. This message only appears on inbound maps.

**Files Used***Input*

The following files are used as input:

Filename	Description
SYS001	Parameters file This file contains the parameter record.
EDISTRN	Standard Transaction file This file contains all transaction sets defined for a standard.
EDISSEG	Standard Segment file This file contains all segments defined for a standard.
EDISSGD	Standard Segment Description file This file contains all segment descriptions.
EDISELE	Standard Element file This file contains all the elements defined for a standard.
EDISELD	Standard Element Description This file contains all element descriptions.

Filename	Description
EDISDIC	Standard Dictionary file This file contains all data elements defined for a standard, identified by reference number.
EDISCD1	Standard Code File 1 This file contains all the codes defined for a standard with a maximum length less than 5.
EDISCD2	Standard Code File 2 This file contains all the codes defined for a standard with a maximum length between 6 and 10.
EDICFG	System Configuration file This file contains all system configuration information.

### *Input/Output*

The following files are used as input and output:

Filename	Description
SYS040	Translation Header file This file contains all the information from the transaction header, and the corresponding information from the Standards file.
SYS041	Translation Segment file This file contains all the translation segments defined in a transaction.
SYS042	Translation Element file This file contains all the translation elements defined in a segment.
SYS023	Application/Trans. Link This file contains records linking the application to the Translation ID.

### *Output*

The following file is used as output:

Filename	Description
SYS010	Output Report file This is a spool file for the output of this program. The report is presented hierarchically — by segment and then element. It displays all fields that were changed in the Transaction Segment and Element records, showing the field values before and after the update.
EDITCHA	Transaction Change Audit File

*Parameter Record Layout*

The following table describes the parameter record layout:

Position	Field Content	COBOL Picture	Format	Length	Description
001 – 010	Old Translation ID	X(10)	AN	10	Existing Translation ID
011 – 011	Send/Receive	X(01)	AN	01	Specifies send or receive criteria. Valid values: <b>S</b> – Send <b>R</b> – Receive
012 – 021	New Translation ID	X(10)	AN	10	New Translation ID name
022 – 033	Version ID	X(12)	AN	12	The standard version you want to convert to
034 – 034	Release Number	X(01)	AN	01	The release number of the transaction set. Valid values: <b>0 – 9</b> (for TRADACOMS only). Default is <b>0</b> .
035 – 035	Straight Copy Option	X(01)	AN	01	Specifies whether a straight copy is performed. Valid values: <b>Y</b> or space.
036 – 038	Agency	X(03)	AN	03	The agency associated with Version ID.

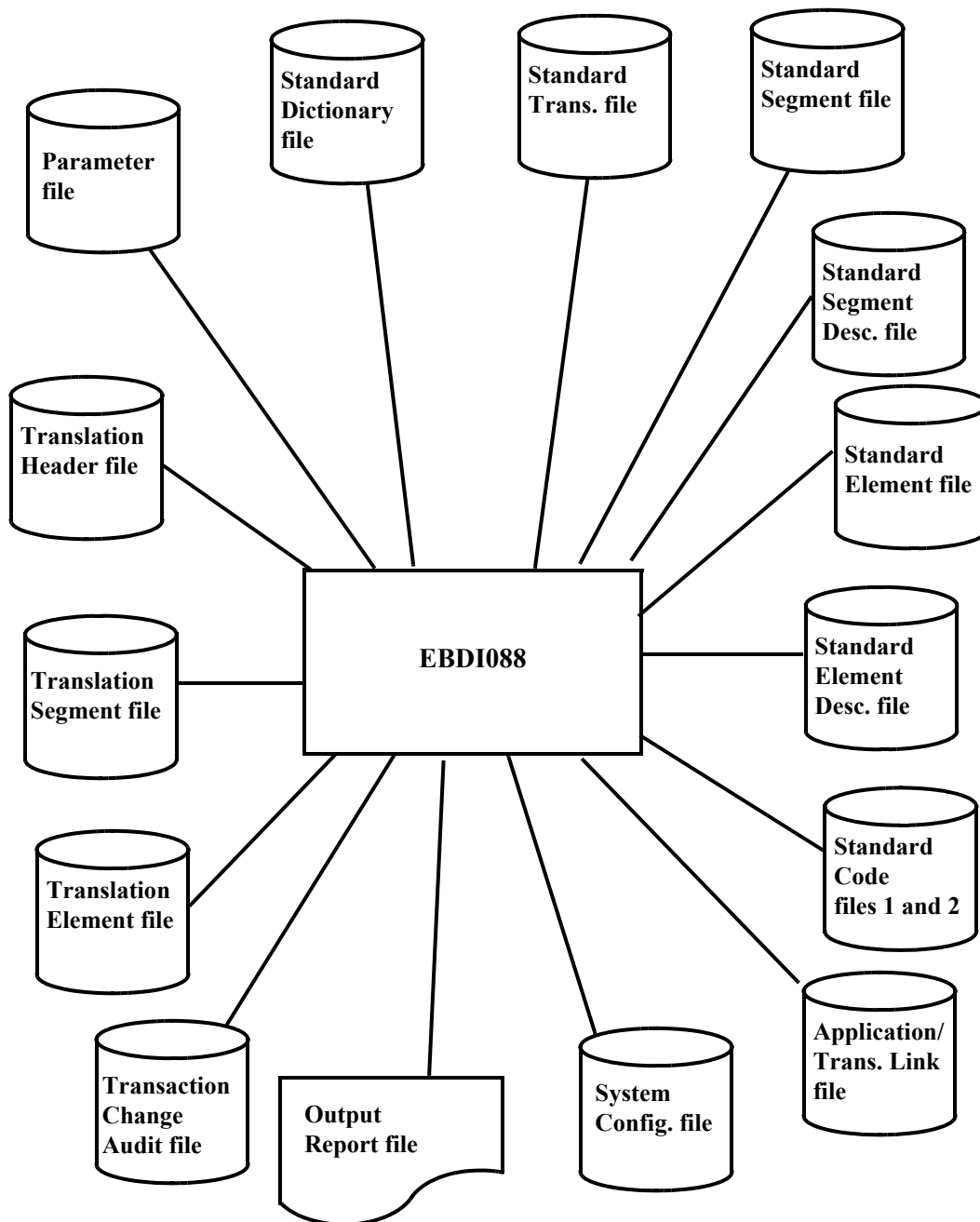
*Optional Loop ID Conversion Parameter Record Layout*

The following table describes the Optional Loop ID Conversion parameter record layout:

Position	Field	COBOL Picture	Format	Length	Description
001 – 001	Segment Type	X(01)	A	01	Valid values: <b>H</b> – header area <b>D</b> – detail area <b>S</b> – summary area
002 – 007	Old Loop ID	X(06)	AN	06	Existing Loop ID
008 – 013	New Loop ID	X(06)	AN	06	The Loop ID to which to convert
014 – 080	Filler	X(67)	AN	67	Filler

Mapping Version Comparison and Conversion Report (EBDI088)

Figure 4.9 illustrates the Mapping Version Comparison and Conversion report:



**Figure 4.9 Program EBDI088 – Mapping Version Comparison and Conversion Report**

## Example Reports

Figure 4.10 through Figure 4.12 are examples of the Mapping Version Comparison and Conversion report.

```

EBDI088      DATE: 12/01/2005      TIME: 12:16      PAGE 1
GENTRAN MAPPING INTEGRATION

          MAPPING VERSION COMPARISON & CONVERSION REPORT
          SEGMENTS

TRANSLATION ID: ANSI3010FO      SEND OR RECEIVE: S      DESCRIPTION: ANSI 002040 OUTBOUND FOS      COPY FROM TRAN ID: ANSI2040FO
NEW VERSION: 003010      X TRANSACTION SET: 850      USE CODE: G (GENERAL)      OLD VERSION: 002040      X
APPLICATION ID: POF1LE      SEND OR RECEIVE: S      ENVELOPE TYPE: X (X12)

NEW SEGMENT      SEG M      MAX LOOP MAX      DESCRIPTION      MESSAGE
REC ID VER H/D SEQ C USE ID LOOP
-----
PID 00 D 04205 0 1 P01      PRODUCT/ITEM DESCRIPTION      SEGMENT NOT COPIED FROM OLD TRANSLATION - COULD NOT
TRNSEG ORIGINAL LOOP ID: P01      BE MATCHED WITH SEGMENT UNDER STANDARD SEGMENT FILE

***** SEGMENT PROCESSING COMPLETED NORMALLY *****

```

**Figure 4.10 Mapping Version Comparison and Conversion Report (Segments)**



PAGE 2

TIME: 12:16

EBDI088 DATE: 12/01/2005 MAPPING VERSION COMPARISON & CONVERSION REPORT  
 GENTRAN MAPPING INTEGRATION ELEMENTS

TRANSLATION ID: ANSI3010FO SEND OR RECEIVE: S DESCRIPTION: ANSI 002040 OUTBOUND POS COPY FROM TRAN ID: ANSI2040FO  
 NEW VERSION: 003010 X TRANSACTION SET: 850 USE CODE: G (GENERAL) OLD VERSION: 002040 X  
 APPLICATION ID: POFIE SEND OR RECEIVE: S ENVELOPE TYPE: X (X12)

NEW SEGMENT	SEG	ELEM	MAP	M	ELEM	ELEM	MIN	MAX	MESSAGE				
REC ID	VER	H/D	SEQ	SEQ	NO	C	REF	VER	TYPE	LEN	LEN	DESCRIPTION / CONDITIONAL	
04200	020	00	M	0330	00	R	1	9	QUANTITY ORDERED			ELEMENT TYPE FIELD HAS CHANGED	
*	F01	00	D	02940	020	00	M	0330	00	R0	1	9	QUANTITY ORDERED
CTT	00	S	08800	020	00	O	0347	00	R	1	10	HASH TOTAL	ELEMENT TYPE FIELD HAS CHANGED
*	CTT	00	S	08800	020	00	O	0347	00	R0	1	10	HASH TOTAL

\*\*\*\*\* ELEMENT PROCESSING COMPLETED NORMALLY \*\*\*\*\*

Figure 4.11 Mapping Version Comparison and Conversion Report (Elements)

```

EBDI088      DATE: 12/01/2005      MAPPING VERSION COMPARISON & CONVERSION REPORT      TIME: 12:16      PAGE      3
GENTRAN MAPPING INTEGRATION      ELEMENTS

TRANSLATION ID: ANSI3010PO      SEND OR RECEIVE: S      DESCRIPTION: ANSI 002040 OUTBOUND POS      COPY FROM TRAN ID: ANSI2040FO
NEW VERSION: 003010      X TRANSACTION SFT: 850      USE CODE: G (GENERAL)      OLD VERSION: 002040      X
APPLICATION ID: FOFIE      SEND OR RECEIVE: S      ENVELOPE TYPE: X (X12)

NEW SEGMENT      SEG      ELEM      MAP      M      ELEM      ELEM      MIN      MAX
REC ID VER      H/D      SEQ      SEQ      NO      C      REF      TYPE      LEN      LEN      DESCRIPTION / CONDITIONAL      MESSAGE
-----
SEGMENT TOTALS:
OLD TRANSLATION SEGMENTS SUCCESSFULLY MATCHED AND COPIED      14
OLD TRANSLATION SEGMENTS NOT MATCHED AND NOT COPIED      01
NEW TRANSLATION SEGMENTS WRITTEN      13

ELEMENT TOTALS:
OLD TRANSLATION ELEMENTS READ      81
OLD TRANSLATION ELEMENTS SUCCESSFULLY MATCHED AND COPIED      81
OLD TRANSLATION ELEMENTS NOT MATCHED AND NOT COPIED      00
STANDARD ELEMENTS NOT MATCHED BUT WERE COPIED      00
NEW TRANSLATION ELEMENTS WRITTEN      81

CHANGE AUDIT RECORDS WRITTEN      01
    
```

Figure 4.12 Mapping Version Comparison and Conversion Report (Summary Page)

---

## Field Definitions

### Segments

**Translation ID:**

This field identifies the user-defined translation map name.

**Send or Receive:**

This field specifies whether the translation map is an inbound or outbound map. Valid values are:

<b>S</b>	Outbound Translation map
<b>R</b>	Inbound Translation map

**Description:**

This field is used to display a text description of the Translation ID.

**Copy From Tran ID:**

This field identifies the translation ID from which the new Translation ID is copied from.

**New Version:**

This field displays the new version and agency of the EDI Standard defined by the new translation map.

**Transaction Set:**

This identifies the transaction set, as defined by the new translation map.

**Use Code:**

This field defines the transaction map use code, followed by a mapped description. The use code identifies whether the transaction map definition may be used for general transaction purposes or is used only for translating data for a particular group of trading partners. Valid values are:

<b>G</b>	General – (may be used with any trading partner)
<b>P</b>	Partner-specific – (must be used for specified trading partners.)

**Old Version:**

This field displays the version and agency of the EDI Standard defined by the Copy From Translation map.

**Application ID:**

This field identifies the user defined application definition name associated with the Translation ID.

**Send or Receive:**

This field specifies whether the application definition is defined for inbound or an outbound processing. Valid values are:

<b>S</b>	Outbound application definition
<b>R</b>	Inbound application definition

**Envelope Type:**

This field defines the data envelope type used with this transaction map header definition. It is followed by a description of the envelope type. Valid values are:

<b>E</b>	EDIFACT
<b>X</b>	X12

<b>U</b>	UCS
<b>G</b>	GS
<b>A</b>	ANA TRADACOMS

**(Straight Copy Option Selected):**

This field is a message that specifies whether or not the Straight Copy Option was selected in the parameter record. If **Y** appears in position 35 of the parameter record, the message Straight Copy Option Selected is displayed. If the field is left blank, the Straight Copy Option was not selected, and this field is also left blank.

**New Rec:**

This indicates whether the displayed records is new or old (from the new or old map of the transaction). Valid values are:

<b>*</b>	New record from translation
<b>blank</b>	Old record from the Copy From translation map

**Segment ID:**

This field displays the segment identifier.

**Segment VER:**

This field displays the segment version associated with the segment ID.

**H/D:**

This field indicates the origin of the segment. Valid values are:

<b>H</b>	Header
<b>D</b>	Detail
<b>S</b>	Summary
<b>F</b>	Float

**Seg Seq:**

This field displays the segment sequence number that is assigned to the transaction segment record when it is copied over.

**MC:**

This field displays the Segment Requirement Designator. This field is printed from the transaction segment record. Valid values are:

<b>M</b>	Mandatory
<b>O</b>	Optional
<b>C</b>	Conditional

**Max Use:**

This field displays the maximum number of times the segment may occur at that specific location within the transaction. This field is printed from the transaction segment record.

**Loop ID:**

This field displays the identification of the loop. This field is printed from the transaction segment record.

**Max Loop:**

This field displays the maximum number of times the loop may occur at that specific location within the transaction. This field is printed from the transaction segment record.

**Description:**

This field displays the segment description, as defined by the specified version of the EDI Standard. The description provides additional text to describe the Segment ID.

**Message:**

This field displays the message associated with the conversion of the segment.

See “List of messages that may appear on the Version Comparison and Conversion report and their meaning,” in this section for detailed information.

**Elements****Translation ID:**

This field identifies the user defined translation map name.

**Send or Receive:**

This field specifies whether the translation map is an inbound or an outbound map. Valid values are:

<b>S</b>	Outbound Translation map
<b>R</b>	Inbound Translation map

**Description:**

This field displays a text description of the Translation ID.

**Copy From Tran ID:**

This field identifies the Translation ID from which the new Translation ID is copied from.

**New Version:**

This field displays the new version of the EDI Standard defined by the new translation map.

**Transaction Set:**

This identifies the transaction set, as defined by the new translation map.

**Use Code:**

This field defines the transaction map use code, followed by a mapped description. The use code identifies whether the transaction map definition may be used for general transaction purposes or is used only for translating data for a particular group of trading partners. Valid values are:

<b>G</b>	General (may be used with any trading partner)
<b>P</b>	Partner-specific (must be used for specified trading partners).

**Old Version:**

This field displays the version of the EDI Standard defined by the Copy From translation map.

**Application ID:**

This field identifies the user defined application definition name associated with the translation map.

**Send or Receive:**

This field specifies whether the application definition is defined for inbound or outbound processing. Valid values are:

<b>S</b>	Outbound application definition
<b>R</b>	Inbound application definition

**Envelope Type:**

This field defines the data envelope type used with this transaction map header definition. It is followed by a description of the envelope type. Valid values are:

<b>E</b>	EDIFACT
<b>X</b>	X12
<b>U</b>	UCS
<b>G</b>	GS
<b>A</b>	ANA TRADACOMS

**(Straight Copy Option Selected):**

This specifies whether or not the Straight Copy Option was selected in the parameter record. If **Y** appears in position 35 of the parameter record, the message Straight Copy Option Selected is displayed. If the parameter record field is blank, the Straight Copy Option was not selected, and this field is also blank.

**New Rec:**

This indicates whether the displayed records are new or old (from the new or old map of the transaction). Valid values are:

<b>*</b>	New record from translation
<b>blank</b>	Old record from the Copy From translation map

**Segment ID:**

This field displays the segment identifier.

**Segment VER:**

This field displays the segment version associated with the Segment ID.

**H/D:**

This field indicates the origin of the element. Valid values are:

<b>H</b>	Header
<b>D</b>	Detail
<b>S</b>	Summary

**Seg Seq:**

This field displays the segment sequence number which is assigned to the transaction segment record.

**Elem Seq:**

This field displays the element sequence number which is assigned to the transaction element record.

**Map No:**

This field indicates the number of the map. This field is printed from the transaction element file.

**MC:**

This field displays the Element Requirement Designator. This field is printed from the transaction element file. Valid values are:

<b>M</b>	Mandatory
<b>O</b>	Optional
<b>F</b>	Floating
<b>C</b>	Conditional

**Element Ref:**

This field displays the dictionary reference number of the element. This field is printed from the transaction element file.

**Element VER:**

This field displays the element version associated with the element reference number.

**Elem Type:**

This field displays the code for the element type. Valid values are:

<b>AN</b>	String Type
<b>ID</b>	Identifier Type
<b>R</b>	Decimal Type
<b>Rn</b>	Decimal type with “n” explicit decimal places
<b>Nn</b>	Numeric with “n” decimal places implied
<b>DT</b>	Date
<b>D8</b>	Date
<b>TM</b>	Time
<b>AS</b>	Application Select

**Min Len:**

This field displays the minimum length for the element. This field is printed from the transaction element file.

**Max Len:**

This field displays the maximum length for the element. This field is printed from the transaction element file.

**Description/Conditional:**

This field displays the element description, as defined by the specified version of the EDI Standard. This field also displays the element conditional code, if one exists for the element. The description provides additional text to describe the Element ID.

**Message:**

This field displays the message associated with the conversion of the element.

See the “List of messages that may appear on the Version Comparison and Conversion report and their meaning” in this section for detailed information.

**Segment Totals****Old Translation Segments Read:**

This field displays the total number of old translation segments read by the program.

**Old Translation Segments Successfully Matched and Copied:**

This field displays the total number of old translation segments matched to standard segments, and copied to the new map.

**Old Translation Segments Not Matched and Not Copied:**

This field displays the total number of old translation segments that could not be matched to standard segments (for any reason), and therefore were not copied to the new map.

**New Translation Segments Written:**

This field displays the total number of new translation segments written to the new map.

**Element Totals****Old Translation Elements Read:**

This field displays the total number of old translation elements read by the program before translation.

**Old Translation Elements Successfully Matched and Copied:**

This field displays the total number of old translation elements matched to standard elements, and copied to the new map.

**Old Translation Elements Not Matched and Not Copied:**

This field displays the total number of old translation elements that could not be matched to standard elements (for any reason), and therefore were not copied to the new map.

**Standard Elements Not Matched But Were Copied:**

This field displays the total number of old translation elements that could not be matched to standard elements (for any reason), but were copied and added to the new map.

**New Translation Elements Written:**

This field displays the total number of new translation elements written to the new map.

**Other Totals****Change Audit Records Written:**

This field displays the total number of Transaction Change Audit Records written. This will only be displayed if you have Change Audit for Transaction enabled on the EDICFG file.



## Object Insert Program

**EBDI065**

The Object Insert (EBDI065) program can be used to merge Objects into outbound packages, such as the (EDIFACT) UNO/UNP or (ASC X12) 102 transaction packages.

This program reads the output from the Outbound Mapper (EBDI042) program or the Outbound Envelope Generator (EBDI011M) program and searches for ###OBJ records that identify locations to insert objects. The Objects must reside in a sequential file (recfm=vb, blksize=3004) and be preceded by a control card that supplies the package reference number. The program uses the package reference number on the ###OBJ record to search for the corresponding number in the Objects file.

During processing, the Object Insert program runs as batch job step in the outbound process. Use this **optional** program only when you need to insert objects into your data.

### Files Used

#### *Input*

The following files are used as input:

DDname	Filename
PARMIN	Parameter file
EDICFG	Configuration file
ERRCTL	Error Control file
EDIIN	EDI Input Data file
OBJECTIN	Object Data file

#### *Output*

The following files are used as output:

DDname	Filename
EDIOUT	Merged EDI file
EDILOG	Processing Log Report
EDISUM	Error/Summary Report

*Parameter File Layout*

The following table describes the parameter file layout:

Position	Field Content	COBOL Picture	Format	Length	Description
001 – 010	TRACE ON	X(10)	AN	10	Constant value

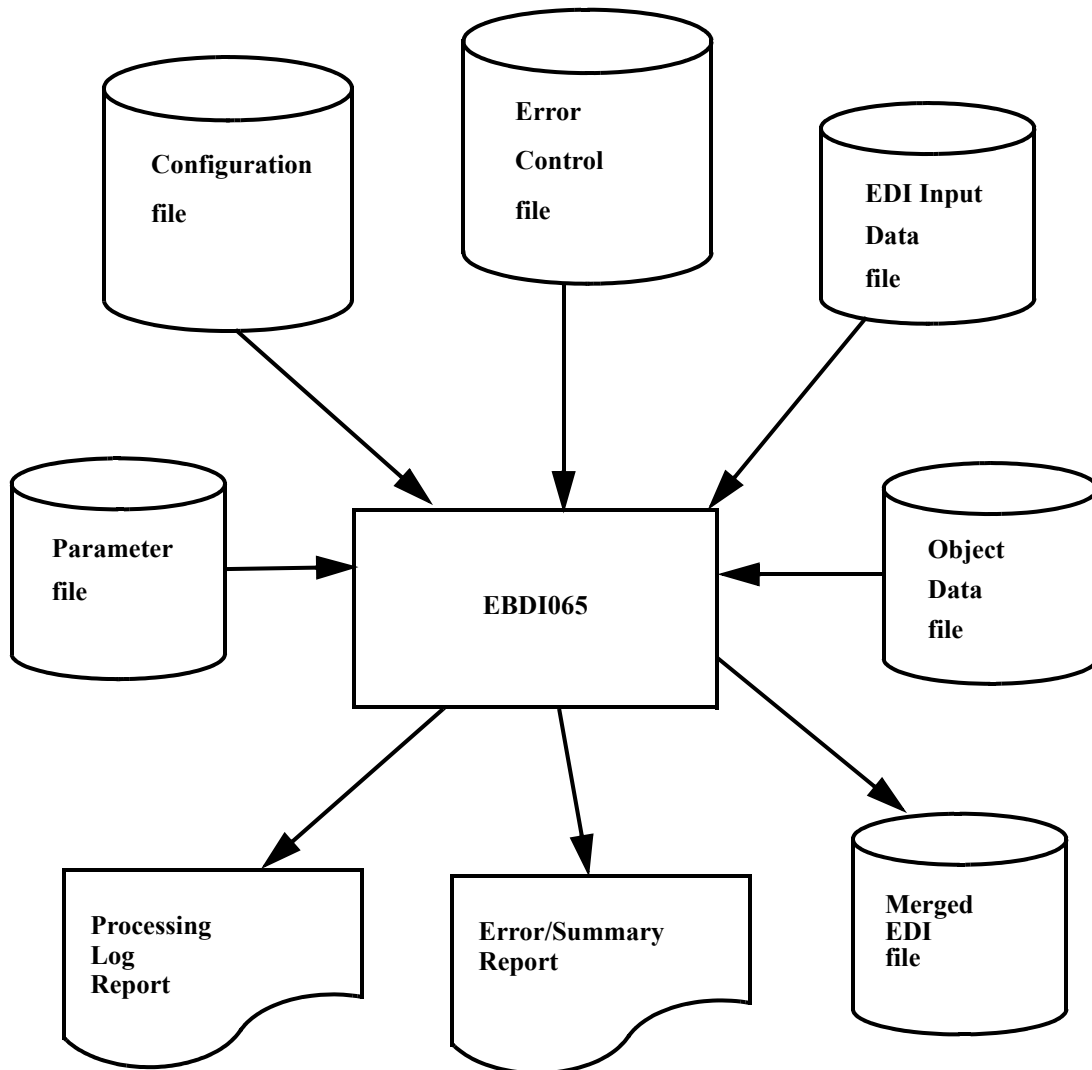
**Note:** Up to 20 parameter records may be entered. If more than 20 parameter records are entered, they are ignored, and processing continues.

*Object Data File Parameters*

Position	Length	Description
01 – 10	10	Value “Object ID”
11 – 45	35	Package reference number from field UNO01.

*Object Insert (EBDI065) Program*

Figure 4.13 illustrates the Object Insert Program.



**Figure 4.13 Program EBDI065 – Object Insert Program**



Sample Object Insert Error/Summary Report

Figure 4.15 represents the Object Error/Summary Report.

```

EBDI065  RUN 12/01/2005  TIME 14:42  ERRORS ENCOUNTERED PROCESSING OBJECTS  PAGE 1
ERROR **RECORD**
NUMBER NBR      INFORMATION      ERROR MESSAGE
NO ERRORS OCCURRED DURING PROCESSING
PROCESSING ENDED NORMALLY - PROCESSING COUNTS BELOW
      INPUT RECORDS READ ----- 10
      OBJECT IDENTIFIERS READ ----- 1
      OBJECT FILE RECORDS READ ----- 4
      OBJECT RECORDS INSERTED ----- 2
      EDI SEGMENTS WRITTEN ----- 11
      RETURN CODE FOR GENERATION ----- 0

The following figure represents the Object Insert Error/Summary Report with errors.

EBDI065  RUN 12/01/2005  TIME 15:30  ERRORS ENCOUNTERED PROCESSING OBJECTS  PAGE 1
ERROR **RECORD**
NUMBER NBR      INFORMATION      ERROR MESSAGE
900      9          LENGTH ERR OBJECT DATA LENGTH DOES NOT MATCH CALCULATED LENGTH 860
          CALCULATED LENGTH: 760 SUPPLIED LENGTH:
PROCESSING ENDED ABNORMALLY - CORRECT ERRORS/RE-RUN JOB
          RETURN CODE FOR GENERATION ----- 16
    
```

Figure 4.15 Object Insert Error/Summary Report Without Errors

## Online Log File Utility

EDIELMT

### Purpose

The Online Log File Utility (EDIELMT) program formats and prints information from the Online Log file. After the Online Log file is defined, this program formats the file for usage by the online portion of Gentran:Basic. This file can then be used by Gentran:Basic (Gentran:Control<sup>®</sup>, Gentran:Realtime<sup>®</sup>, Gentran:Viewpoint<sup>®</sup>) to hold messages written from online background (non-terminal) transactions. An additional function of the program is to print a report of all the messages for a particular transaction. This facility lists the messages in a form that is easy to read and easy to fax to Gentran:Basic support.

### Operation

The Log File Utility program runs as a batch job. A sample JCL to run this program is located in member EXECLMT in the JCL library.

### Files Used

#### Input

The following files are used as input:

Filename	Description
EDICNTL	Parameter file This file contains the control parameters that tell the Utility program which operation to perform.
ERRCTL	Error Message and Control file This file contains the text for all messages created during the program execution.
EDIOLF	Online Log file This is the file upon which the operation is to be performed.
EDICFG	System Configuration file

#### Output

The following files are used as output:

Filename	Description
EDISUM	Summary file This is the print file that the program writes the echo of the operation to be performed, the listing of the Online Log file and any summary statistics about the process.
EDILOG	Log file This is the print file that the program lists all messages that are generated during the process.

Filename	Description
SYSOUT	Output file This is the print file on which the program lists information such as the compile date and version number.

## Parameter File Layout

### *Format Process Records*

The following tables describe the FORMAT process record layouts.

#### Parameter Record 1

Position	Field Content	Format	Length	Description
001-006	FORMAT	AN	6	Identifies the process to be performed is to format (initialize) the Online Log file.
007-080	Unused	AN	74	Spaces

#### Parameter Record 2

Position	Field Content	Format	Length	Description
001-002	Unused	AN	2	Spaces
003-007	LOG	AN	5	Identifies the number of log records to be formatted.
008-016	Count	N	9	The count of the number of log records to be formatted.
017-080	Unused	AN	64	Spaces

*Print Process Records*

The following tables describe the PRINT process record layouts.

**Parameter Record 1**

Position	Field Content	Format	Length	Description
001-006	PRINT	AN	6	Print the messages for the identified transaction.
007-080	Unused	AN	74	Spaces

**Parameter Record 2**

Position	Field Content	Format	Length	Description
001-002	Unused	AN	2	Spaces
003-007	TASK	AN	5	Identifies the task number to be printed (used only with the PRINT process).
008-014	Task Number	N	7	The task number to be printed. Leading zeros are required.
015-080	Unused	AN	66	Spaces

**Parameter Record 3**

Position	Field Content	Format	Length	Description
001-002	Unused	AN	2	Spaces
003-007	DATE	AN	5	Identifies the date of the task to be printed (used only with the PRINT process).
008-017	Task Date	N	10	The Date the task was processed with a format of MM/DD/YYYY.
018-080	Unused	AN	67	Spaces



---

## Partner Change Audit Maintenance Program

**EBDI093**

The Partner Change Audit Maintenance program (EBDI093) maintains the Partner Change Audit file. The program functions include deleting, archiving, and restoring change audit information.

The partner change audit file contain records that track changes made to the Partner file and either the Partner Xref file or the Partrel file, depending on the trading profile mode indicated in the system configuration file. Add, delete and update functions via the online partner subsystem result in records being written to the Partner Change Audit file. The Batch copy utility (EBDI087) and its online counterpart EDIR087 will also write records to the Partner change audit file indicating add, delete and replace events.

The Partner Change Audit file contains two different types of change audit records: Partner file information and Xref/Partrel file information. Housekeeping will delete or archive all data—both Partner and Xref/Partrel—in a single run because it is based on specified retention periods. Purge and restore maintenance of the different types of Partner records (partner, xref, partrel) requires separate runs of the program. Each execution will process the type of Partner Change audit records that are indicated by the RECORD-TYPE control card. The default for Record-Type is "P" - partner. To maintain Xref records you must use Record-Type of "X". In order to maintain Partrel records you must use Record-type of "R".

**Note:** You must close the Partner Change Audit file (SIMPCHA) to CICS before running maintenance.

### Operation

The Partner Change Audit Maintenance program operates in the following general modes:

- **Periodic Housekeeping**  
On a periodic basis (anywhere from daily to monthly depending on volume and retention requirements) the Housekeeping operation deletes and optionally archives data based on specified retention periods.
- **Ad Hoc Maintenance (Purge and Restore)**  
In addition to the regularly scheduled housekeeping runs, the Partner Change Audit Maintenance's Purge and Restore functions deletes and optionally archives selected data and restores data from archives.

### Processing

The Partner Change Audit Maintenance program processes documents using the following three operations:

- Housekeeping
- Purge
- Restore

### *Housekeeping Operation*

If the Housekeeping operation is requested, Partner Change Audit Maintenance deletes and optionally archives the change audit records that exceed the retention periods specified in the control cards. An option of the Housekeeping operation is to archive all selected change audit records to a sequential output file. Partner and Xref/Partrel records will be deleted or archived by a housekeeping run.

### *Purge Operation*

If the Purge operation is requested, change records are removed if they match ALL criteria specified. An option of the Purge operation is to archive selected change audit records to a sequential output file.

### *Restore Operation*

If the Restore operation is requested, change audit records are restored from the Archive file to the Change Audit if the records match ALL selection criteria specified.

## **Files Used**

### *Input*

Control cards and the Partner Change Audit files are the principal input to Partner Change Audit Maintenance.

PRTARCH	Partner Change Audit Archive File
EDIPCHA	6.4 VSAM Partner Change Audit File
EDICNTL	Control Card File
ERRCTL	Error Message and Control File
EDICFG	System Configuration File

### *Output*

The following files are used as output:

PRTARCH	Partner Change Audit Archive File
EDIPCHA	6.4 VSAM Partner Change Audit File

### *Reports*

Program EBDI093 produces the following reports:

PRTAUDT	Partner Change Audit Maintenance Report
EDILOG	Partner Change Audit Processing Log
EDISUM	Partner Change Audit Summary Report

## Control Card Options

### Operations

HOUSEKEEPING (Default Operation)  
 PURGE  
 RESTORE

### Selection Criteria

#### Defaults

For Housekeeping operation:

RETENTION-DAYS            30

For Purge and Restore operations:

FROM-UPDATE-DATE        01/01/1900  
 FROM-UPDATE-TIME        00:00  
 TO-UPDATE-DATE          12/31/2099  
 TO-UPDATE-TIME          23:59  
 RECORD-TYPE             P  
 PARTNER-ID                ALL  
 PARTNER-QUALIFIER        ALL  
 USER-PARTNER            ALL  
 XREF-ID                    ALL  
 XREF-ID-QUAL             ALL  
 USER-ID                    ALL  
 USER-QUALIFIER          ALL

For Purge and Housekeeping operations:

ARCHIVE                    NO

### Valid Criteria Values

#### Archive

This controls whether an archive is to be created for deleted change records.

NO

YES

#### From-Update-Date

Specifies the date that the update was applied to the partner. Starts the range. Used with To-Update-Date.

MM/DD/YYYY

#### From-Update-Time

Specifies the time that the update was applied to the Partner. Starts the range. Used with To-Update-Time.

HH:MM

**Partner-ID**

A 35-character partner identification used to restrict maintenance by Partner ID. Used in Trading profile mode Partner/Qualifier to identify the desired Partner ID. Used in Relationship Mode to identify the Partner EDI ID to identify the desired PARTREL.

ALL  
XX

**Partner-Qualifier**

The 4-character Qualifier for the Partner ID. You can specify a Partner/Qualifier or ALL for all Partner/Qualifiers. Used in Trading profile mode Partner/Qualifier to identify the desired Partner ID. Used in Relationship Mode to identify the Partner EDI ID to identify the desired PARTREL.

ALL  
XXXX

**Record-Type**

A 1-character field that describes the records on the Partner Change Audit file that are to be affected by this maintenance.

- P** - Partner records      Used for both Trading modes
- X** - Xref records        Used for Partner Qualifier mode only
- R** - Partrel records.    Used for Relationship mode only.

**Retention-Days**

Specifies the number of days that have elapsed since the update was applied.

000 - 999

**To-Update-Date**

Specifies the date that the update was applied to the partner. Ends the range. Used with From-Update-Date.

MM/DD/YYYY

**To-Update-Time**

Specifies the time that the update was applied to the Partner. Ends the range. Used with From-Update-Time.

HH:MM

**User-ID**

A 35-character partner identification used to restrict maintenance by Partrel. Used in Relationship Mode to identify the User EDI ID to identify the desired PARTREL.

ALL  
XX

**User-Partner**

A 31-character partner identification used to restrict maintenance by User/Partner Profile. Positions 1 - 15 represent the User portion of User/Partner profile. Position 16 contains '/'. Position 17 - 31 represents the Partner portion of the User/partner profile.Used in Relationship Mode to restrict maintenance by Partner ID.

ALL  
XXXXXXXXXXXXXXXXXX/YYYYYYYYYYYYYYYY

Where:      X-----X represents the User  
              Y-----Y represents the Partner

**User-Qualifier**

The 4-character Qualifier for the User ID. You can specify a User/Qualifier or ALL for all USER-ID. Used in Relationship Mode to identify the Partner EDI ID to identify the desired PARTREL.

ALL  
XXXX

**Xref-ID**

A 35-character partner identification used to restrict maintenance by Xref ID. Used in Trading profile mode Partner/Qualifier only.

ALL  
XX

**Xref-ID-Qual**

The 4-character Qualifier for the Xref- ID. You can specify a Xref Qualifier or ALL for all Xref-IDs. Used in Trading profile mode Partner/Qualifier only.

ALL  
XXXX

## Partners Copy From One File to Another

EBDI087

### Purpose

The Partners Copy (EBDI087) program is one of the five Test/Production Integration Copy programs. The trading partners to be copied are specified in a parameter file. More than one partner may be copied with each execution of the Partners Copy program.

Partner profile records can be directly copied (i.e., VSAM to VSAM) or copied to a sequential upload file for later upload. The formats of partner profiles vary, depending upon the trading partner processing mode specified by the System Configuration file. Therefore, you should copy only between parallel processing modes (e.g., Partner/Qualifier to Partner/Qualifier or Relationship to Relationship). When processing in the Partner/Qualifier mode, the Partner ID and the Partner/Qualifier data make up a partner profile. For the Relationship mode, the User/Partner ID data makes up the partner profile.

If you have Change Audit for Partner enabled on the “1” record of the EDICFG file, a new record will be written to the Partner Change Audit file (EDIPCHA) for each partner that is added, deleted, or replaced.

**Note:** If the Copy From Partner Profile (ID) is different from Copy To Partner Profile (ID), no cross-reference records for the new Partner ID (Copy To) are created.

### Files Used

#### Input

The following files are used as input:

Filename	Description
SYS030	Parameter file The partner identifier and partner qualifier for all partners to be copied. Each record contains ‘to’ and ‘from’ fields to allow the partner to be renamed on the new file. The Unload function can not be intermixed with the Copy function.
SYS080	Source Partner Profile Partner records from which the partners are copied.
SYS085	Source Cross Reference file This file contains Partner/Qualifier X-Ref records.
SYS086	Source Partner Relationship file This file contains Partner Relationship records.
SYS081	Source Outbound Control file This file contains the Outbound Control records for the source partner profile.

Filename	Description
SYS082	Source Inbound Control file This file contains the Inbound Control records for the source partner profile.
EDICFG	System Configuration file <b>Note:</b> If using the Change Audit feature, make sure that the DD statement in your EXEC087 JCL for this file contains the dataset name as it appears in your target CICS region.

### Output

The following files are used as output:

Filename	Description
SYS090	Target Partner Profile This file contains partner records to which the partners are copied.
SYS095	Target Cross Reference file This file contains Partner/Qualifier X-Ref records.
EDIPREL	Target Partner Relationship file This file contains relationship cross-reference records.
SYS091	Target Outbound Control file This file contains the records to which the test outbound control records are copied.
SYS092	Target Inbound Control file This file contains the records to which the test inbound control records are copied.
SYS060	Sequential Unload file
EDIPCHA	Partner Change Audit file <b>Note:</b> If using the Change Audit feature, make sure that the DD statement in your EXEC087 JCL for this file contains the dataset name as it appears in your target CICS region.

### Parameter File Notes

The trading processing mode of the source and target files must be the same. The Copy and Unload function use the same parameter records. For the Unload function, however, a records consisting of "UNLOAD PROCESS" in columns 1 through 15 must be placed at the beginning of the parameter input.

*Parameter File Layout*

The following table describes the parameter file layout for the Partner/Qualifier mode.

Position	Field Layout	COBOL Picture	Format	Length	Description
001 – 035	Copy from Partner ID	X(35)	AN	35	Copy from Partner ID of partner to be copied.
036 – 039	Copy from Partner/Qualifier	X(04)	AN	04	Copy from Partner ID Qualifier of partner to be copied.
040 – 040	Spaces	X(01)	AN	01	
041 – 075	Copy to Partner ID	X(35)	AN	35	Copy to Partner ID of partner to be copied.
076 – 079	Copy to Partner Qualifier	X(04)	AN	04	Copy to Partner/Qualifier
080 – 080	Replace Switch (Y/N)	X(01)	AN	01	<p><b>Y</b> – Deletes the existing partner from the Copy To files.</p> <p><b>C</b> – Deletes all existing partner records from the Copy To files except the Inbound and Outbound Control files so you can maintain the control numbers on the Copy To partner.</p> <p><b>N</b> – Process data, but system indicates the partners that need to be deleted.</p> <p><b>Note:</b> The system does not copy if partner exists on Copy To file and the replace switch is N.</p>

The following table describes the parameter file layout for the Relationship mode.

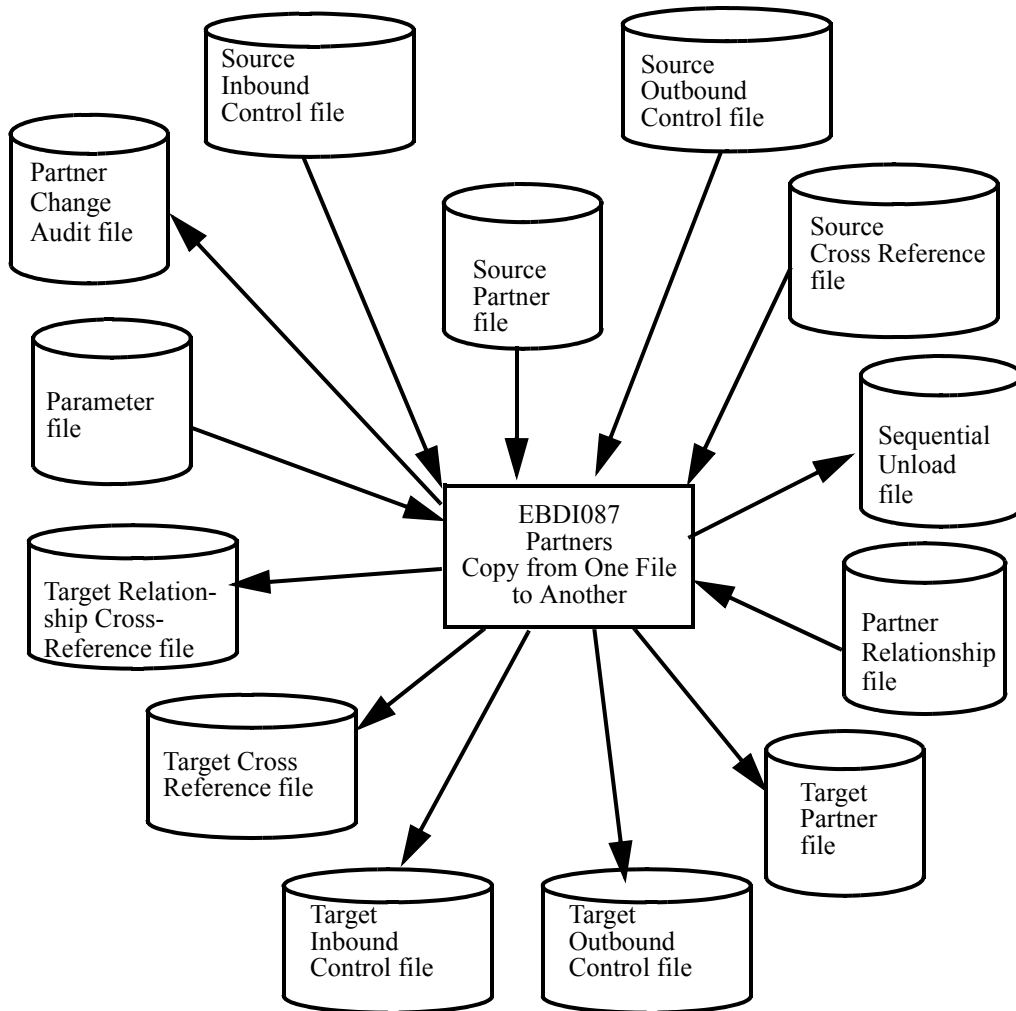
Position	Field Layout	COBOL Picture	Format	Length	Description
001 – 015	Copy from User ID	X(15)	AN	15	Copy from User ID of records to be copied.
016 – 030	Copy from Partner ID	X(15)	AN	15	Copy from Partner ID of records to be copied.
031 – 040	Spaces	X(10)	AN	10	Not used
041 – 055	Copy to User ID	X(15)	AN	15	Copy of User ID of records to be created.
056 – 070	Copy to Partner ID	X(15)	AN	15	Copy to Partner ID of records to be created.



Position	Field Layout	COBOL Picture	Format	Length	Description
071 – 079	Spaces	X(09)	AN	09	
080 – 080	Replace Switch (Y/N)	X(01)	AN	01	<p><b>Y</b> – Deletes the existing partner from the Copy To files.</p> <p><b>C</b> – Deletes all existing partner records from the Copy To files except the Inbound and Outbound Control files so you can maintain the control numbers on the Copy To partner.</p> <p><b>N</b> – Process data, but system indicates the partners that need to be deleted.</p> <p><b>Note:</b> The system does not copy if partner exists on Copy To file and the replace switch is <b>N</b>.</p>

*Partners Copy From One File to Another*

Figure 4.16 illustrates how to copy partners from one file to another.



**Figure 4.16 Program EBDI087 – Partners Copy**

---

## Security Change Audit Maintenance Program

**EBDI100**

The Security Change Audit Maintenance program (EBDI100) maintains the Security Change Audit file. The program functions include deleting, archiving, and restoring change audit information.

The Security change audit file contains records that track changes made to the Security profiles that reside on the Security file. Add, delete, and update functions via the online Security subsystem result in records being written to the Security Change Audit file.

### Operation

The Security Change Audit Maintenance program operates in the following general modes:

#### **Periodic Housekeeping**

On a periodic basis (anywhere from daily to monthly depending on volume and retention requirements), the Housekeeping operation deletes and optionally archives data based on specified retention periods.

#### **Ad Hoc Maintenance (Purge and Restore)**

In addition to the regularly scheduled housekeeping runs, the Security Change Audit Maintenance Purge and Restore functions delete and optionally archive selected data and restore data from archives.

### Processing

The Security Change Audit Maintenance program processes documents using the following three operations:

- Housekeeping
- Purge
- Restore

#### *Housekeeping Operation*

If the Housekeeping operation is requested, Security Change Audit Maintenance deletes and optionally archives the change audit records that exceed the retention periods specified in the control cards. An option of the Housekeeping operation is to archive all selected change audit records to a sequential output file.

#### *Purge Operation*

If the Purge operation is requested, change records are removed if they match *all* criteria specified. An option of the Purge operation is to archive selected change audit records to a sequential output file.

#### *Restore Operation*

If the Restore operation is requested, change audit records are restored from the Archive file to the Change Audit if the records match *all* selection criteria specified.

## Files Used

### Input

Control cards and the Security Change Audit files are the principal input to Security Change Audit Maintenance.

Filename	Description
SECARCH	Security Change Audit Archive File
EDIXCHA	6.4 VSAM Security Change Audit File
EDICNTL	Control Card File
ERRCTL	Error Message and Control File
EDICFG	System Configuration File

### Output

The following files are used as output:

Filename	Description
SECARCH	Security Change Audit Archive File
EDIXCHA	6.4 VSAM Security Change Audit File

### Reports

Program EBDI100 produces the following reports:

Report Name	Description
SECAUDT	Security Change Audit Maintenance Report
EDILOG	Security Change Audit Processing Log
EDISUM	Security Change Audit Summary Report

## Control Card Options

### Operations

- HOUSEKEEPING (Default Operation)
- PURGE
- RESTORE

**Selection Criteria****Defaults**

For Housekeeping operation:

RETENTION-DAYS

For Purge and Restore operations:

FROM-UPDATE-DATE 01/01/1900

FROM-UPDATE-TIME 00:00

TO-UPDATE-APPLIED-DATE 12/31/2099

TO-UPDATE-APPLIED-TIME 23:59

USER-ID ALL

For Purge and Housekeeping operations:

ARCHIVE NO

**Valid Criteria Values****Archive**

This controls whether an archive is to be created for deleted change records.

NO

YES

**From-Update-Date**

Specifies the date that the update was applied to the Security record. Starts the range. Used with To-Update-Applied-Date.

MM/DD/YYYY

**From-Update-Time**

Specifies the time that the update was applied to the Security record. Starts the range. Used with To-Update-Applied-Time.

HH:MM

**User-ID**

An 8-character identification used to restrict maintenance by Security User ID.

ALL

XXXXXXXX

**Retention-Days**

Specifies the number of days that have elapsed since the update was applied.

000 – 999

**To-Update-Applied-Date**

Specifies the date that the update was applied to the Security record. Ends the range. Used with From-Update-Date.

MM/DD/YYYY

**To-Update-Applied-Time**

Specifies the time that the update was applied to the Security record. Ends the range. Used with From-Update-Time.

HH:MM

## Separator Change Audit Maintenance Program

**EBDI104**

The Separator Change Audit Maintenance program (EBDI104) maintains the Separator Change Audit file. The program functions include deleting, archiving, and restoring change audit information.

The Separator change audit file contains records that track changes made to the System and Priority Options records that reside on the Separator file. Add, delete, and update functions for the separator records via the online Separator subsystem result in records being written to the Separator Change Audit file.

### Operation

The Separator Change Audit Maintenance program operates in the following general modes:

#### **Periodic Housekeeping**

On a periodic basis (anywhere from daily to monthly depending on volume and retention requirements), the Housekeeping operation deletes and optionally archives data based on specified retention periods.

#### **Ad Hoc Maintenance (Purge and Restore)**

In addition to the regularly scheduled housekeeping runs, the Separator Change Audit Maintenance Purge and Restore functions delete and optionally archive selected data and restore data from archives.

### Processing

The Separator Change Audit Maintenance program processes documents using the following three operations:

- Housekeeping
- Purge
- Restore

#### *Housekeeping Operation*

If the Housekeeping operation is requested, Separator Change Audit Maintenance deletes and optionally archives the change audit records that exceed the retention periods specified in the control cards. An option of the Housekeeping operation is to archive all selected change audit records to a sequential output file.

#### *Purge Operation*

If the Purge operation is requested, change records are removed if they match *all* criteria specified. An option of the Purge operation is to archive selected change audit records to a sequential output file.

#### *Restore Operation*

If the Restore operation is requested, change audit records are restored from the Archive file to the Change Audit if the records match *all* selection criteria specified.

## Files Used

### Input

Control cards and the Separator Change Audit files are the principal input to Separator Change Audit Maintenance.

Filename	Description
SEPARCH	Separator Change Audit Archive File
EDIZCHA	6.4 VSAM Separator Change Audit File
EDICNTL	Control Card File
ERRCTL	Error Message and Control File
EDICFG	System Configuration File

### Output

The following files are used as output:

Filename	Description
SEPARCH	Separator Change Audit Archive File
EDIZCHA	6.4 VSAM Separator Change Audit File

### Reports

Program EBDI104 produces the following reports:

Report Name	Description
SEPAUDT	Separator Change Audit Maintenance Report
EDILOG	Separator Change Audit Processing Log
EDISUM	Separator Change Audit Summary Report

## Control Card Options

### Operations

- HOUSEKEEPING (Default Operation)
- PURGE
- RESTORE

**Selection Criteria****Defaults**

For Housekeeping operation:

RETENTION-DAYS

For Purge and Restore operations:

FROM-UPDATE-DATE	01/01/1900
FROM-UPDATE-TIME	00:00
TO-UPDATE-APPLIED-DATE	12/31/2099
TO-UPDATE-APPLIED-TIME	23:59
TRANS-GROUP-ID	ALL
SENDER-ID	ALL
SENDER-QUAL	ALL
RECEIVER-ID	ALL
RECEIVER-QUAL	ALL
VERSION	ALL
TEST-PROD-IND	ALL
IGT-LEVEL-IND	ALL

For Purge and Housekeeping operations:

ARCHIVE	NO
---------	----

**Valid Criteria Values****Archive**

This controls whether an archive is to be created for deleted change records.

NO

YES

**From-Update-Date**

Specifies the date that the update was applied to the Separator record. Starts the range. Used with To-Update-Applied-Date.

MM/DD/YYYY

**From-Update-Time**

Specifies the time that the update was applied to the Separator record. Starts the range. Used with To-Update-Applied-Time.

HH:MM

**IGT-Level-Ind**

A 3-character identification used to restrict maintenance by the Interchange/Group/Transaction Level Indicator.

ALL

I	Interchange
---	-------------

G	Group
---	-------

T	Transaction
---	-------------



**Receiver-Id**

A 35-character identification used to restrict maintenance by Receiver ID.

ALL  
XXXXX

**Receiver-Qual**

A 4-character identification used with the Receiver ID to restrict maintenance of Separator Change Audit records.

ALL  
XXXX

**Retention-Days**

Specifies the number of days that have elapsed since the update was applied.

000 – 999

**Sender-Id**

A 35-character identification used to restrict maintenance by Sender Id.

ALL  
XXXXX

**Sender-Qual**

A 4-character identification used with the Sender Id to restrict maintenance of Separator Change Audit records.

ALL  
XXXX

**Test-Prod-Ind**

A 3-character identification used to restrict maintenance by Test/Production Indicator.

ALL  
T            Test  
P            Production

**Trans-Group-ID**

A 6-character identification used to restrict maintenance by Transaction or Group ID.

ALL  
XXXXXX

**To-Update-Applied-Date**

Specifies the date that the update was applied to the Separator record. Ends the range. Used with From-Update-Date.

MM/DD/YYYY

**To-Update-Applied-Time**

Specifies the time that the update was applied to the Separator record. Ends the range. Used with From-Update-Time.

HH:MM

**Version**

A 12-character identification used to restrict maintenance by Version Id.

ALL  
XXXXXXXXXXXX

## Sort Generated Acknowledgments Program

EBDI110

### Purpose

The Sort Generated Acknowledgments (EBDI110) program is the first step in the acknowledgment process. This program is responsible for resequencing and reformatting the Acknowledgment file generated by the Inbound Editor. Because of the sequence numbers assigned to the acknowledgment records, it is important that this program be run immediately after the Inbound Editor (EBDI001).

The Sort Generated Acknowledgment program produces an Acknowledgment file that may or may not contain envelopes. If the global parameter, Use CNTL for Acknowledgments, is used in the Inbound Editor, there will be corresponding CNTL records for the acknowledgments. If the output file contains envelopes, the acknowledgments are ready for processing through the Outbound Editor (EBDI002). If the Acknowledgment file contains CNTL control records, processing must continue through the Acknowledgment Generation (EBDI011A) program before running through the Outbound Editor.

### Processing Summary

When generating functional acknowledgments for ASC X12, the TA1 interchange acknowledgment must appear after the ISA and before the GS envelope. However, the TA1 represents the status of the entire interchange processed by the Inbound Editor.

When generating functional acknowledgments for EDIFACT, the UCI interchange acknowledgment segment is required and must be the first segment in the CONTRL transaction. However, the UCI represents the status of the entire interchange processed by the Inbound Editor. The UCF group acknowledgment segment has a similar problem in EDIFACT.

In both cases, the standards define the acknowledgment segments in a sequence that differs from the actual point of generation within the Inbound Editor. To handle this sequence conflict, the Inbound Editor tacks a sort key onto each acknowledgment record generated.

The Sort Generated Acknowledgments (EBDI110) program sorts the Acknowledgment file created by the Inbound Editor using the tacked-on sort key, which resequences the segments into their proper locations, as defined by their respective standards.

After the Acknowledgment file is resequenced, the sort key that was added by the Inbound Editor must be removed. This program strips the sort key from each acknowledgment record.

### Files Used

The following are files used by the EBDI110 program:

Filename	Description
SYS050	Sort Work file for output file extraction. All records are sorted based on sequencing a key within the output file.
SYS030	Input file (30013-character variable input) This file contains the acknowledgment records generated by the Inbound Editor.
SYS040	Output file (30009-character variable output) This file is the output containing sorted acknowledgments with no sort key.

---

## Standard Change Audit Maintenance Program

**EBDI033**

The Standard Change Audit Maintenance program (EBDI033) maintains the Standard Change Audit file. The program functions include deleting, archiving, and restoring change audit information.

The Standard change audit file contain records that track changes made to the thirteen Standards files. Add, delete and update functions via the online Standard subsystem result in records being written to the Standard Change Audit file. The Batch Copy utility (EBDI032) and Batch Standards Maintenance (EBDI039) will also write records to the Standard change audit file indicating add, delete and replace functions.

**Note:** You must close the Standard Change Audit file (SIMSCHA) to CICS before running maintenance.

### Operation

The Standard Change Audit Maintenance program operates in the following general modes:

- **Periodic Housekeeping**  
On a periodic basis (anywhere from daily to monthly depending on volume and retention requirements) the Housekeeping operation deletes and optionally archives data based on specified retention periods.
- **Ad Hoc Maintenance (Purge and Restore)**  
In addition to the regularly scheduled housekeeping runs, the Standard Change Audit Maintenance's Purge and Restore functions deletes and optionally archives selected data and restores data from archives.

### Processing

The Standard Change Audit Maintenance program processes documents using the following three operations:

- Housekeeping
- Purge
- Restore

#### *Housekeeping Operation*

If the Housekeeping operation is requested, Standard Change Audit Maintenance deletes and optionally archives the change audit records that exceed the retention periods specified in the control cards. An option of the Housekeeping operation is to archive all selected change audit records to a sequential output file.

#### *Purge Operation*

If the Purge operation is requested, change records are removed if they match ALL criteria specified. An option of the Purge operation is to archive selected change audit records to a sequential output file.

### *Restore Operation*

If the Restore operation is requested, change audit records are restored from the Archive file to the Change Audit if the records match ALL selection criteria specified.

### **Files Used**

#### *Input*

Control cards and the Standard Change Audit files are the principal input to Standard Change Audit Maintenance.

STDARCH	Standard Change Audit Archive File
EDISCHA	6.4 VSAM Standard Change Audit File
EDICNTL	Control Card File
ERRCTL	Error Message and Control File
EDICFG	System Configuration File

#### *Output*

The following files are used as output:

STDARCH	Standard Change Audit Archive File
EDISCHA	6.4 VSAM Standard Change Audit File

### *Reports*

Program EBDI033 produces the following reports:

STDAUDT	Standard Change Audit Maintenance Report
EDILOG	Standard Change Audit Processing Log
EDISUM	Standard Change Audit Summary Report

### **Control Card Options**

#### *Operations*

HOUSEKEEPING (Default Operation)  
PURGE  
RESTORE

#### *Selection Criteria*

##### **Defaults**

For Housekeeping operation:

RETENTION-DAYS

For Purge and Restore operations:

FROM-UPDATE-DATE	01/01/1900
FROM-UPDATE--TIME	00:00

TO-UPDATE-APPLIED-DATE 12/31/2099

TO-UPDATE-APPLIED-TIME 23:59

VERSION ALL

AGENCY ALL

For Purge and Housekeeping operations:

ARCHIVE NO

### *Valid Criteria Values*

#### **Archive**

This controls whether an archive is to be created for deleted change records:

NO

YES

#### **From-Update--Date**

Specifies the date that the update was applied to the Standard. Starts the range. Used with To-Update-Date.

MM/DD/YYYY

#### **From-Update-Time**

Specifies the time that the update was applied to the Standard. Starts the range. Used with To-Update-Time.

HH:MM

#### **Retention-Days**

Specifies the number of days that have elapsed since the update was applied.

000 - 999

#### **To-Update--Date**

Specifies the date that the update was applied to the Standard. Ends the range. Used with From-Update-Date.

MM/DD/YYYY

#### **To-Update-Time**

Specifies the time that the update was applied to the Standard. Ends the range. Used with From-Update-Time.

HH:MM

#### **Version**

A 12-character identification used to restrict maintenance by version Id.

ALL

XXXXXXXXXXXX

#### **Agency**

A 3-character value associated with the Version Id.

ALL

XXX

## Standards Copy Utility

EBDI032

### Purpose

The Standards Copy Utility (EBDI032) program is used to copy standard versions or transactions from one set of VSAM files to another set of VSAM files. The program can also be used to delete standard versions from the target file. The ADD and REPLACE actions are used to copy. The DELETE action is used for deletions.

If you have Change Audit for Standards enabled on the “1” record of the EDICFG file, a new record will be written to the Standard Change Audit file (EDISCHA) for each version that is added, deleted, or replaced.

### Files Used

#### Input

The files in the following table are used as input.

File	Description
SYS030	Sorted Parameter File Parameters are sorted together by action type, version ID, agency, rename version ID, rename agency, transaction ID, and release. Up to 300 parameter records can be processed at one time.
ERRCTL	Error Message File
EDICFG	System Configuration File <b>Note:</b> If using the Change Audit feature, make sure that the DD statement in your EXEC032 JCL for this file contains the dataset name as it appears in your target CICS region.
INSVER	Source Standard Version File The versions from which the standard is copied
INSTRN	Source Standard Transaction File The transactions from which the standard is copied
INSSEG	Source Standard Segment File The segments from which the standard is copied
INSSGD	Source Standard Segment Description File The segment descriptions from which the standard is copied
INSACT	Source Standard Activity File The activity from which the standard is copied
INSELE	Source Standard Element File The element from which the standard is copied
INSELD	Source Standard Element Description File The element description from which the standard is copied
INSDIC	Source Standard Dictionary File The dictionary from which the standard is copied

File	Description
INSCDE1	Source Standard Code File 1 The code 1 records from which the standard is copied
INSCDE2	Source Standard Code File 2 The code 2 records from which the standard is copied
INSCDE3	Source Standard Code File 3 The code 3 records from which the standard is copied
INSCDE4	Source Standard Code File 4 The code 4 records from which the standard is copied

*Input/Output*

The files in the following table are used for output and input during processing.

File	Description
EDISVER	Target Standard Version File The versions to which the standard is copied
EDISTRN	Target Standard Transaction File The transactions to which the standard is copied
EDISSEG	Target Standard Segment File The segments to which the standard is copied
EDISSGD	Target Standard Segment Description File The segment descriptions to which the standard is copied
EDISACT	Target Standard Activity File The activity to which the standard is copied
EDISELE	Target Standard Element File The element to which the standard is copied
EDISELD	Target Standard Element Description File The element description to which the standard is copied
EDISDIC	Target Standard Dictionary File The dictionary to which the standard is copied
EDISCD1	Target Standard Code File 1 The code 1 file to which the standard is copied
EDISCD2	Target Standard Code File 2 The code 2 file to which the standard is copied
EDISCD3	Target Standard Code File 3 The code 3 file to which the standard is copied
EDISCD4	Target Standard Code File 4 The code 4 file to which the standard is copied

## Output

The following files are used for output:

File	Description
SYS006	Standards Copy Utility – Summary Reports A separate report is issued for each action type performed.
SYS010	Standards Copy Utility – Processing Log Shows error messages, if any.
EDISCHA	Standard Change Audit file <b>Note:</b> If using the Change Audit feature, make sure that the DD statement in your EXEC032 JCL for this file contains the dataset name as it appears in your target CICS region.

## Parameter Record Layout

The following table describes the parameter record layout:

Position	Field Content	COBOL Picture	Format	Length	Description
01 – 08	Action	X(08)	AN	8	Use ADD, REPLACE, or DELETE.
09 – 20	Version ID	X(12)	AN	12	Version ID selected
21 – 23	Agency	X(03)	AN	03	Agency associated with the Version ID
24 – 29	Transaction Set	X(06)	AN	06	Transaction Set ID selected, or *ALL
30	Release Number	X(01)	AN	01	Release number of the transaction set
31 – 42	Rename Version ID	X(12)	AN	12	Optional. Used with ADD and REPLACE only. Used in conjunction with Rename Agency to define the target version/agency when it is not the same as the source version/agency. Not used with DELETE.
43 – 45	Rename Agency	X(03)	AN	03	Only used when Rename Version ID is used.

## Parameter Usage

Multiple parameters can be used to perform an operation for a particular version/agency and multiple operations can be carried out in a single run. Parameter records are sorted in the SORTPARM step of EXEC032 so that ADD operations are performed first, REPLACE operations are performed next, and DELETE operations are performed last.



When multiple parameters are used for an operation, all fields other than the transaction field must be consistent from one parameter record to the next. Parameter records with like fields are sorted together in the SORTPARM step of EXEC032.

### *The ADD Parameter*

A single ADD parameter with \*ALL in the transaction field can be used to copy an entire version from the source files to the target files. If only a subset of transactions is desired, multiple parameters can be used where each parameter record lists one of the transactions to be copied. All fields other than the transaction field must be consistent from one parameter record to the next if they are to define a single ADD operation. Control transactions (such as 997, CONTRL, or the “blank” transaction) that may be needed for a given envelope type are always copied, even if they are not requested in the parameter records.

An ADD operation, whether using one parameter or many, will not be carried out if the target version/agency is already present in the target version file.

**Recommendation:** If you need to ADD multiple transactions for a single version, specify all transactions using multiple parameters during a single job run. You cannot ADD some transactions and then ADD more to an existing version in a later run.

### *Special note regarding the REPLACE and DELETE parameters*

Individual transactions cannot be replaced or deleted in a target version/agency because different transactions in a version/agency use common segments, elements, and codes. Replacing or deleting individual transactions could result in conflicts; only the whole version/agency can be replaced or deleted.

### *The REPLACE Parameter*

A REPLACE operation can be performed using a single parameter with “\*ALL” in the transaction field or with multiple parameters that list the transactions to be copied. Transactions indicated in the parameter record(s) for the REPLACE operation must match the transactions found in the version/agency to be replaced in the target files. If an exact match is not found, the REPLACE operation will not take place. If a single parameter record with \*ALL is used, then the transactions found in the source version/agency must match the transactions found in the target version/agency in the target files.

If the target version/agency is not found in the target version file, the REPLACE operation will be treated as an ADD operation.

**Recommendation:** To replace a version/agency regardless of the transactions present, use a DELETE parameter with \*ALL in the transaction field to clear the version/agency from the target files. Then use ADD parameter(s) to copy transactions for the desired version/agency to the target files. The DELETE and ADD operations should be done in separate runs.

### The DELETE Parameter

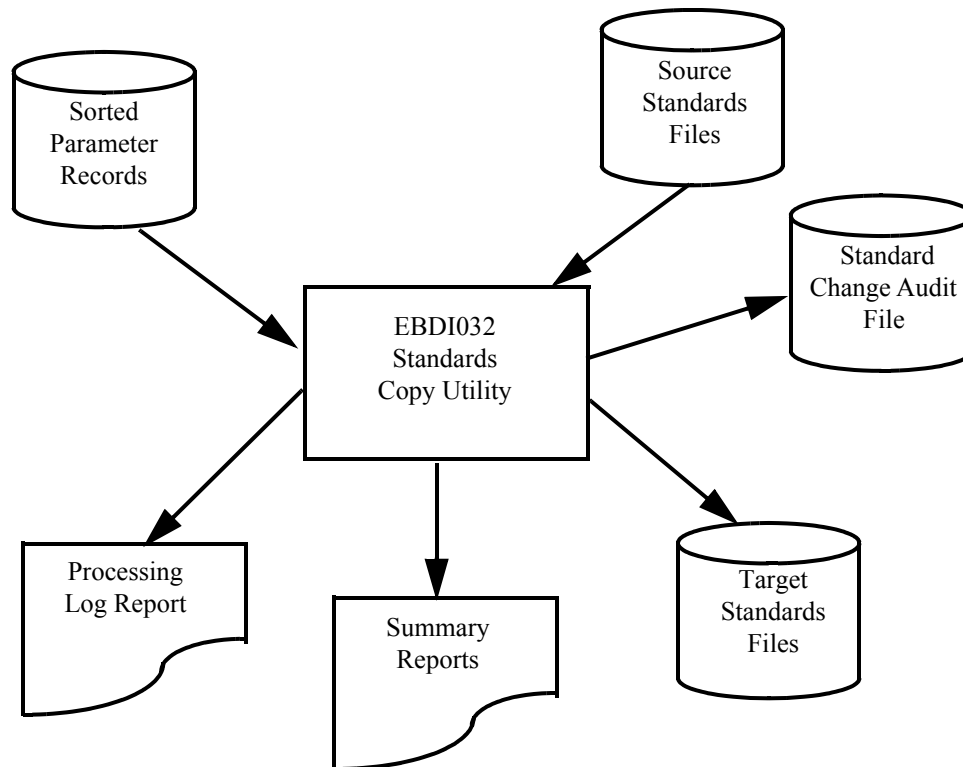
A DELETE operation can be accomplished using a single parameter with \*ALL in the transaction field or with parameter(s) that list individual transactions to be deleted. If individual transaction(s) are entered, the transactions indicated in the parameter record(s) must match the transactions found in the version/agency to be deleted from the target files. If the transactions do not match, the DELETE operation will not take place.

If \*ALL is used in the transaction field, no safety checks will be made. All records for the version/agency indicated will be removed from the target files even if a version record or transaction record does not exist.

**Recommendation:** If you wish to delete all transactions along with the version, specify only the transactions that are on the target version. Too many or too few transactions in the parameters will result in an error message. The message indicates which transactions need to be included in the DELETE parameters to aid you in correcting the error. Alternatively, DELETE with \*ALL eliminates the need to list specific transactions.

### Standards Copy Utility Process

Figure 4.17 illustrates the Standards Copy Utility process.



**Figure 4.17 Program EBDI032 – Standards Copy Utility**

## Standards Selective Download

## EBDI030

### Purpose

The Standards Selective Download (EBDI030) program builds the Optimized Table file for the Inbound and Outbound Editors. The Optimized Table file is the repository for all information that describes the ASC X12, TDCC, UCS, WINS, ANA, TRADACOMS, ODETTE, and EDIFACT Standards.

In the online Standards subsystem, the Transaction In Use (EDIM180) screen enables you to select the version and transactions that you process. Only those versions with transactions selected through the Transaction In Use screen are processed by this program. This feature enables you to create a customized Optimized Table file from the online Standards files.

Any time that you alter the versions and transactions selected on the Transaction In Use screen, you must run the EXEC030 job to update your Optimized Table file.

Use the following tables to determine which versions and transactions to select.

### Default

Always use the default version and transactions for all documents.

Version	Agency	Transaction to be Selected	Description
Default	SC	Blank Transaction CONTROL 997 999	<ul style="list-style-type: none"> <li>Envelope information</li> <li>Acknowledgment for EDIFACT</li> <li>Acknowledgment for X12</li> <li>Acknowledgment for UCS</li> </ul>

### For ANSI Only

Use the following versions and transactions for ANSI documents you have selected:

Version	Agency	Transaction to be Selected	Description
00100	X	Blank Transaction 997 999	<ul style="list-style-type: none"> <li>Envelope information</li> <li>Acknowledgment for X12</li> <li>Acknowledgment for UCS</li> </ul>
00200	X	Blank Transaction 997 999	<ul style="list-style-type: none"> <li>Envelope information</li> <li>Acknowledgment for X12</li> <li>Acknowledgment for UCS</li> </ul>
00300	X	Blank Transaction 997 999	<ul style="list-style-type: none"> <li>Envelope information</li> <li>Acknowledgment for X12</li> <li>Acknowledgment for UCS</li> </ul>

Version	Agency	Transaction to be Selected	Description
00400	X	Blank Transaction 997 999	<ul style="list-style-type: none"> <li>Envelope information</li> <li>Acknowledgment for X12</li> <li>Acknowledgment for UCS</li> </ul>

*For EDIFACT Only*

Use the following versions and transactions for EDIFACT documents you have selected:

Version	Agency	Transaction to be Selected	Description
002001	UN	CONTRL	Acknowledgment for EDIFACT
002002	UN	CONTRL	Acknowledgment for EDIFACT
004001	UN	CONTRL	Acknowledgment for EDIFACT

*For TDCC Only*

The “general” TDCC version associated with an TDCC version that your are processing should also be selected. For example, if you have selected versions U2/5 and UG/6, the following “general” versions should also be selected:

Version	Agency	Transaction to be Selected	Description
G2/5	T	Blank Transaction 990 994 995 996 997 998 999	If using U2/5 <ul style="list-style-type: none"> <li>Generalized feedback</li> <li>Administrative Message</li> <li>Advisory Information</li> <li>File Transfer</li> <li>Acknowledgment for ANSI</li> <li>Set Cancellation</li> <li>Acknowledgment for UCS</li> </ul>
G2/6	T	Blank Transaction 990 994 995 996 997 998 999	If using U2/6 <ul style="list-style-type: none"> <li>Generalized feedback</li> <li>Administrative Message</li> <li>Advisory Information</li> <li>File Transfer</li> <li>Acknowledgment for ANSI</li> <li>Set Cancellation</li> <li>Acknowledgment for UCS</li> </ul>

The EBDI030 program cross-checks and verifies that necessary parts (e.g., number of elements for a Segment ID) are present. If an error is detected, the program displays a message, bypasses the version/transaction in error, and continues to process the remaining selected versions/transactions.

## Files Used

### Input

The following are files used as input:

Filename	Description
EDISVER	VSAM Version file This file contains all versions of the standards supported by Gentran:Basic.
EDISTRN	VSAM Transaction file This file contains the transaction information for the versions of the standards.
EDISSEG	VSAM Segment file This file contains the segment information for the transactions of the standards.
EDISACT	VSAM Segment Element Activity file This file contains the segment element activity information for the elements of the standards.
EDISLD	VSAM Element Description file This file contains the descriptions for the elements in the Segment Element file.
EDISELE	VSAM Segment Element file This file contains the segment element information for the segments of the standards.
EDISASC	VSAM Standards Association file This file contains the association/agency relationships for standard versions.
EDISDIC	Online Data Element Definition file This file contains the data element definition information for the elements of the standards.
ERRCTL	Error Message and Control file This file contains all error message information.

### Output

The following file is used as output:

Filename	Description
SYS001	Sequential Optimized Table (Output) This file contains all standard records that were selected for the standards download.

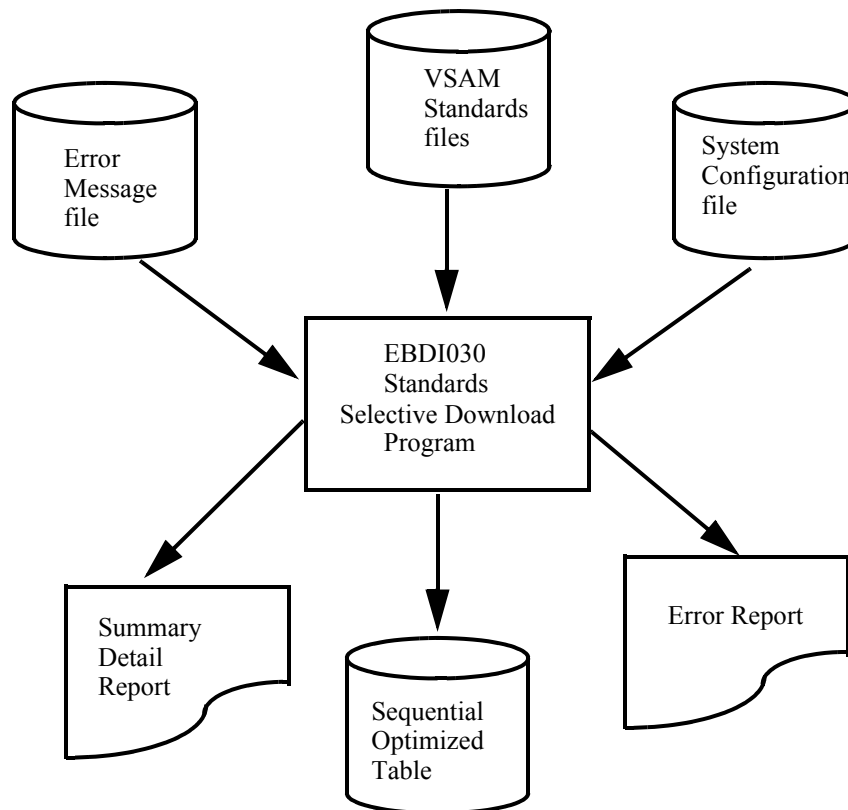
## Reports

The EBDI030 program produces the following reports:

Filename	Description
SYS006	Summary Detail Report
SYS010	Error Report
EDICFG	System Configuration file

## Standards Selective Download (EBDI030)

Figure 4.18 illustrates the Standards Selective Download process:



**Figure 4.18 Program EBDI030 – Standards Selective Download**

## Standards Update

## EBDI038, EBDI039

### Purpose

The Standards Update feature enables you to limit the update of your online standard files to a few selected versions. With this procedure, you can also keep existing standards unchanged on your online standards files. This procedure enables you to replace an existing standard, add a new standard, or delete a standard from your standard files.

If you have Change Audit for Standards enabled on the “1” record of the EDICFG file, a new record will be written to the Standard Change Audit file (EDISCHA) for each version that is added, deleted, or replaced.

See the “Standards Updates” topic in Chapter 7 of the *Gentran: Basic for zSeries Release 6.4 User's Guide* for a detailed worksheet designed to help you perform standards updates.

### Files Used (EBDI038)

#### Input

The following files are used as input:

Filename	Description
SYS030	Parameter file This file contains all the parameters used to run EBDI038.
EDICFG	System Configuration file
ERRCTL	Error Message and Control file
EDISASC	Standard Association file VSAM file that contains the association/agency relationship.
SEQASC	Standard Association file
SEQVER	Standard Version file This is a CD file containing all the standard versions.
SEQTRN	Standard Transaction file This is a CD file containing all the standard transactions.
SEQSEG	Standard Segment file This is a CD file containing all the standard segment information.
SEQSGD	Standard Segment and Description file This is a CD file containing all segment description information.
SEQELE	Standard Element file This is a CD file containing all the standard elements.
SEQELD	Standard Element Description file This is a CD file containing all the element description information.
SEQACT	Standard Activity file This is a CD file containing all the standard activity information.
SEQDIC	Standard Dictionary file This is a CD file containing all the standard dictionary information.

Filename	Description
SEQCD1	Standard Code File 1 This is a CD file containing all the standard code file 1 information.
SEQCD2	Standard Code File 2 This is a CD file containing all the standard code file 2 information.
SEQCD3	Standard Code File 3 This is a CD file containing all the standard code file 3 information.
SEQCD4	Standard Code File 4 This is a CD file containing all the standard code file 4 information.

### Output

The following files are used as output:

Filename	Description
EXTASC	Extract Association file This file contains all extracted association information.
EXTVER	Extract Version file This file contains all the extracted version information.
EXTTRN	Extract Transaction file This file contains all the extracted transaction information.
EXTSEG	Extract Segment file This file contains all the extracted segment information.
EXTELE	Extract Element file This file contains all the extracted element information.
EXTACT	Extract Activity file This file contains all the extracted activity information.
EXTDIC	Extract Dictionary file This file contains all the extracted dictionary information.
EXTCD1	Extract Code File 1 This file contains all the extracted code file 1 information.
EXTCD2	Extract Code File 2 This file contains all the extracted code file 2 information.
EXTCD3	Extract Code File 3 This file contains all the extracted code file 3 information.
EXTCD4	Extract Code File 4 This file contains all the extracted code file 4 information.
SYS010	Output Report file This file contains the Standards Extract Report.
EXTSGD	Extract Segment Description file This file contains all the extracted segment description information.



Filename	Description
EXTELD	Extract Element Description file This file contains all the extracted element description information.
CTLOUT	Control Records file This file contains all the parameters used to run EBDI039.

### Files Used (EBDI039)

#### Input

The following files are used as input:

Filename	Description
SYS030	Control Records file This file contains all the parameters used to run EBDI039.
EDICFG	System Configuration file <b>Note:</b> If using the Change Audit feature, make sure that the DD statement in your UPDSTD2 JCL for this file contains the dataset name as it appears in your target CICS region.
ERRCTL	Error Message and Control file
SEQVER	Sequential Standard Version file This file contains a backup of your VSAM standard versions.
SEQTRN	Sequential Standard Transaction file This file contains a backup of your VSAM standard transaction information.
SEQSEG	Sequential Standard Segment file This file contains a backup of your VSAM standard segment information.
SEQSGD	Sequential Segment Description file This file contains a backup of your VSAM segment description information.
SEQELE	Sequential Standard Element file This file contains a backup of your VSAM standard element information.
SEQELD	Sequential Element Description file This file contains a backup of your VSAM element description information.
SEQACT	Sequential Standard Activity file This file contains a backup of your VSAM standard activity information.
SEQDIC	Sequential Standard Dictionary file This file contains a backup of your VSAM standard dictionary information.

Filename	Description
SEQCD1	Sequential Standard Code File 1 This file contains a backup of your VSAM standard code file 1 information (length 1-5).
SEQCD2	Sequential Standard Code File 2 This file contains a backup of your VSAM standard code file 2 information (length 6-10).
SEQCD3	Sequential Standard Code File 3 This file contains a backup of your VSAM standard code file 3 information (length 11-20).
SEQCD4	Sequential Standard Code File 4 This file contains a backup of your VSAM standard code file 4 information (length 21-40).
EXTVER	Extract Version file This file contains the extracted version information.
EXTTRN	Extract Transaction file This file contains the extracted transaction information.
EXTSEG	Extract Segment file This file contains the extracted segment information.
EXTSGD	Extract Segment Description This file contains the extracted segment description information.
EXTELE	Extract Element file This file contains the extracted element information.
EXTELD	Extract Element Description file This file contains the extracted element description information.
EXTACT	Extract Activity file This file contains the extracted activity information.
EXTDIC	Extract Dictionary file This file contains all the extracted dictionary information.
EXTCD1	Extract Code File 1 This file contains all the extracted code file 1 information.
EXTCD2	Extract Code File 2 This file contains all the extracted code file 2 information.
EXTCD3	Extract Code File 3 This file contains all the extracted code file 3 information.
EXTCD4	Extract Code File 4 This file contains all the extracted code file 4 information.

*Output*

The following files are used as output:

Filename	Description
NEWVER	New Sequential Standard Version file This file contains all the updated sequential standard versions.
NEWTRN	New Sequential Standard Transaction file This file contains all the updated sequential standard transaction information.
NEWSEG	New Sequential Standard Segment file This file contains all the updated sequential standard segment information.
NEWSGD	New Sequential Standard Segment Description file This file contains all the updated sequential standard segment description information.
NEWELE	New Sequential Standard Element file This file contains all the updated sequential standard element information.
NEWELD	New Sequential Standard Element Description file This file contains all the updated sequential standard element description information.
NEWACT	New Sequential Standard Activity file This file contains all the updated sequential standard activity information.
NEWDIC	New Sequential Standard Dictionary file This file contains all the updated sequential standard dictionary information.
NEWCD1	New Sequential Standard Code File 1 This file contains the updated sequential standard code file 1 information.
NEWCD2	New Sequential Standard Code File 2 This file contains the updated sequential standard code file 2 information.
NEWCD3	New Sequential Standard Code File 3 This file contains the updated sequential standard code file 3 information.
NEWCD4	New Sequential Standard Code File 4 This file contains the updated sequential standard code file 4 information.
SYS010	Output Report file This file contains the Standards Merge report.
EDISCHA	Standard Change Audit file <b>Note:</b> If using the Change Audit feature, make sure that the DD statement in your UPDSTD2 JCL for this file contains the dataset name as it appears in your target CICS region.

*Parameter Layout (EBDI038)*

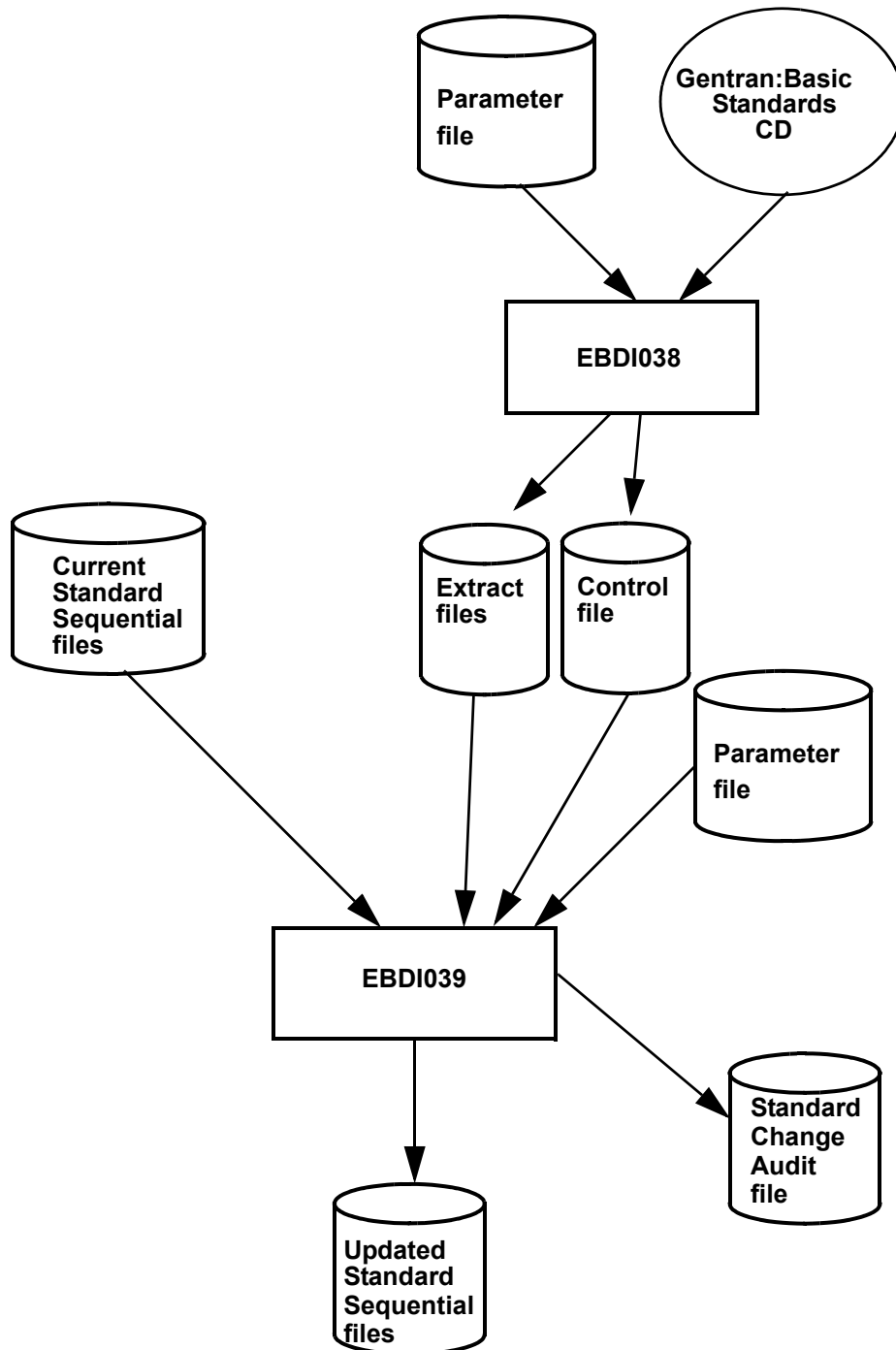
The following table describes the parameter layouts for the EBDI038 program.

**Select Versions**

<b>Position</b>	<b>Field Content</b>	<b>COBOL Picture</b>	<b>Format</b>	<b>Length</b>	<b>Description</b>
01 – 12	Version ID	X(12)	AN	12	Version ID of standard
13 – 13	Filler	X(01)	AN	01	Filler
14 – 16	Agency	X(03)	AN	03	Agency associated with Version
17 – 17	Filler	X(01)	AN	01	Filler
18 – 24	Action	X(07)	AN	07	ADD – Add new version. REPLACE – Replace existing version. DELETE – Delete existing version.
25	Filler	X(01)	AN	01	Filler
26	Bypass Codes	X(01)	AN	01	Y – Do not unload codes. N – Codes will be unloaded. Space – Codes will be unloaded.
27 – 80	Filler	X(54)	AN	54	Filler

*Standards Update (EBDI038 and EBDI039 Programs)*

Figure 4.19 illustrates the Standards Update process.



**Figure 4.19 Programs EBDI038 and EBDI039 – Standards Update**

## Transaction Change Audit Maintenance Program

**EBDI067**

The Transaction Change Audit Maintenance program (EBDI067) maintains the Transaction Change audit file. The program functions include deleting, archiving, and restoring change audit information.

The Transaction change audit file contain records that track changes made to the Transaction file. Add, delete and update functions via the online Transaction subsystem result in records being written to the Transaction Change Audit file. A change audit record may also be written by batch utility programs EBDI080 (Copy), EBDI088 (Version Comparison and Conversion), and EBDI089 (Translation Resynch).

### Operation

The Transaction Change Audit Maintenance program operates in the following general modes:

#### **Periodic Housekeeping**

On a periodic basis (anywhere from daily to monthly depending on volume and retention requirements) the Housekeeping operation deletes and optionally archives data based on specified retention periods.

#### **Ad Hoc Maintenance (Purge and Restore)**

In addition to the regularly scheduled housekeeping runs, the Transaction Change Audit Maintenance's Purge and Restore functions deletes and optionally archives selected data and restores data from archives.

### Processing

The Transaction Change Audit Maintenance program processes documents using the following three operations:

- Housekeeping
- Purge
- Restore

#### *Housekeeping Operation*

If the Housekeeping operation is requested, Transaction Change Audit Maintenance Deletes and optionally archives the change audit records that exceed the retention periods specified in the control cards. An option of the Housekeeping operation is to archive all selected change audit records to a sequential output file.

#### *Purge Operation*

If the Purge operation is requested, change records are removed if they match ALL criteria specified. An option of the Purge operation is to archive selected change audit records to a sequential output file.

### Restore Operation

If the Restore operation is requested, change audit records are restored from the Archive file to the Change Audit if the records match ALL selection criteria specified.

### Files Used

#### Input

Control cards and the Transaction Change Audit files are the principal input to Transaction Change Audit Maintenance.

Filename	Description
TRNARCH	Transaction Change Audit Archive File
EDITCHA	6.4 VSAM Transaction Change Audit File
EDICNTL	Control Card File
ERRCTL	Error Message and Control File
EDICFG	System Configuration File

#### Output

The following files are used as output:

Filename	Description
TRNARCH	Transaction Change Audit Archive File
EDITCHA	6.4 VSAM Transaction Change Audit File

#### Reports

Program EBDI067 produces the following reports:

Report Name	Description
TRNAUDT	Transaction Change Audit Maintenance Report
EDILOG	Transaction Change Audit Processing Log
EDISUM	Transaction Change Audit Summary Report

### Control Card Options

#### Operations

- HOUSEKEEPING (Default Operation)
- PURGE
- RESTORE

*Selection Criteria***Defaults**

For Housekeeping operation:

RETENTION-DAYS

For Purge and Restore operations:

FROM-UPDATE-DATE	01/01/1900
FROM-UPDATE--TIME	00:00
TO-UPDATE-DATE	12/31/2099
TO-UPDATE-TIME	23:59
RECORD-TYPE	P
TRANSACTION-ID	ALL
SEND-RECEIVE	ALL

For Purge and Housekeeping operations:

ARCHIVE	NO
---------	----

*Valid Criteria Values***Archive**

This controls whether an archive is to be created for deleted change records.

NO

YES

**From-Update--Date**

Specifies the date that the update was applied to the Transaction. Starts the range. Used with To-Update-Date.

MM/DD/YYYY

**From-Update-Time**

Specifies the time that the update was applied to the Transaction. Starts the range. Used with To-Update-Time.

HH:MM

**Retention-Days**

Specifies the number of days that have elapsed since the update was applied.

000 – 999

**Send-Receive**

Specifies the direction of the map. Used with Transaction-Id to restrict the records available for maintenance.

S	–	Send (outbound)
R	–	Receive (outbound)
ALL	–	Both Send and Receive



**To-Update--Date**

Specifies the date that the update was applied to the Transaction. Ends the range. Used with From-Update-Date.

MM/DD/YYYY

**To-Update-Time**

Specifies the time that the update was applied to the Transaction. Ends the range. Used with From-Update-Time.

HH:MM

**Transaction-ID**

A 10-character Transaction Id used to restrict maintenance by Transaction ID.

ALL

XXXXXXXXXX

## Transaction Copy From One Set of VSAM Files to Another EBDI080

### Purpose

The Transaction Definition Copy (EBDI080) program is one of the five Test/Production Integration Copy programs. This program is used to move a transaction from the test environment to the production environment. It can also be used to delete a transaction definition from the target files.

The transaction definitions can be directly copied (i.e., VSAM to VSAM) or to a sequential unload file for later upload.

The transaction to be copied or deleted is specified in a parameter record. Also, the Header record must exist on the Source Header file before it may be copied (unless the DELETE function is specified).

If you have Change Audit for Transaction enabled on the "1" record of the EDICFG file, a new record will be written to the Transaction Change Audit file (EDITCHA) for each transaction map that has been added, deleted, or replaced.

### Files Used

#### Input

The following files are used as input:

Filename	Description
SYS001	Parameter file The identity of the translation to be copied. Each record contains 'From' and 'To' fields to allow the transaction to be renamed on the new file. Each delete record identifies the transaction being deleted from the target files. The Unload function can not be intermixed with the Copy and Delete functions
SYS020	Source Transaction Header file The header records from which the transaction is copied.
SYS021	Source Transaction Segment file The segments records from which the transaction is copied.
SYS211	Source Transaction Segment Path file This file is a secondary allocation for the Source Transaction Segment file.
SYS022	Source Transaction Element file The element records from which the transaction is copied.
SYS221	Source Transaction Element Path file This file is a secondary allocation for the Source Transaction Element file.
SYS023	Source Application/Transaction Link file This file contains the cross reference records that link transactions to application Definitions.

Filename	Description
EDICFG	System Configuration file <b>Note:</b> If using the Change Audit feature, make sure the DD statement in your EXEC080 JCL for this file contains the dataset name as it appears in your target CICS region.

### Output

The following files are used as output:

Filename	Description
SYS030	Target Transaction Header file The header records to which the transaction is copied, or from which it is deleted.
SYS031	Target Transaction Segment file The segment records to which the transaction is copied, or from which it is deleted.
SYS311	Target Transaction Segment Path file This file is a secondary allocation for the Target Transaction Segment file.
SYS032	Target Transaction Element file The element records to which the transaction is copied, or from which it is deleted.
SYS321	Target Transaction Element Path file This file is a secondary allocation for the Target Transaction Element file.
SYS033	Target Application/Transaction Link file This file contains the cross reference records to which the transaction is copied, or from which it is deleted.
SYS060	Sequential Unload file This file is used during the Unload processing. It contains the records that will be used by the CICS Upload process to update VSAM application files.
SYS070	Parameter file This file contains parameters that can be input to EBDI076 to request unload of related tables. If you unload more than one transaction definition, and these transaction definitions use the same tables, the same table IDs will be written to the SYS070 file causing the same table to be listed in the unload file multiple times. Although a duplicate error will occur while running the upload, the tables will be uploaded correctly.
EDITCHA	Transaction Change Audit file <b>Note:</b> If using the Change Audit feature, make sure the DD statement in your EXEC080 JCL for this file contains the dataset name as it appears in your target CICS region.

*Parameter File Layout (Copy Function)*

Position	Field Content	COBOL Picture	Format	Length	Description
001 – 005	FROMb	X(05)	AN	05	Constant value of FROMb.
006 – 015	Copy from Transaction ID	X(10)	AN	10	Copy from Transaction ID of transaction to be copied.
016 – 016	Spaces	X(01)		01	
017 – 017	Copy from S or R	X(01)	AN	01	Copy from Send or Receive Code of transaction to be copied. Valid values: <b>S</b> – Send <b>R</b> – Receive.
018 – 024	bbbbTOb	X(07)	AN	07	Constant value of bbbbTOb.
025 – 034	Copy to Transaction ID	X(10)	AN	10	Copy to Transaction ID of transaction to be copied.
035 – 035	Spaces	X(01)	AN	01	
036 – 036	Copy to Send/Receive	X(01)	AN	01	Copy to Send or Receive Code of transaction to be copied. Valid values: <b>S</b> – Send <b>R</b> – Receive.
037 – 039	Spaces	X(03)	AN	03	
040 – 040	Replace Switch (Y/N)	X(01)	AN	01	<b>Y</b> – Deletes the existing transaction from the Copy To files <b>N</b> – Process data, but system indicates the transactions that need to be deleted.
041 – 080	Spaces	X(40)	AN	40	

*Parameter File Layout (Delete Function)*

Position	Field Content	COBOL Picture	Format	Length	Description
001 – 005	DEL	X(05)	AN	05	Constant value of DEL.
006 – 015	TRANS MAP	X(10)	AN	10	Constant value of TRANS MAP
016 – 024	Filler	X(09)	AN	09	Not used
025 – 034	Translation ID being deleted	X(10)	AN	10	Translation ID being deleted
035 – 035	Filler	X(01)	AN	01	Not used

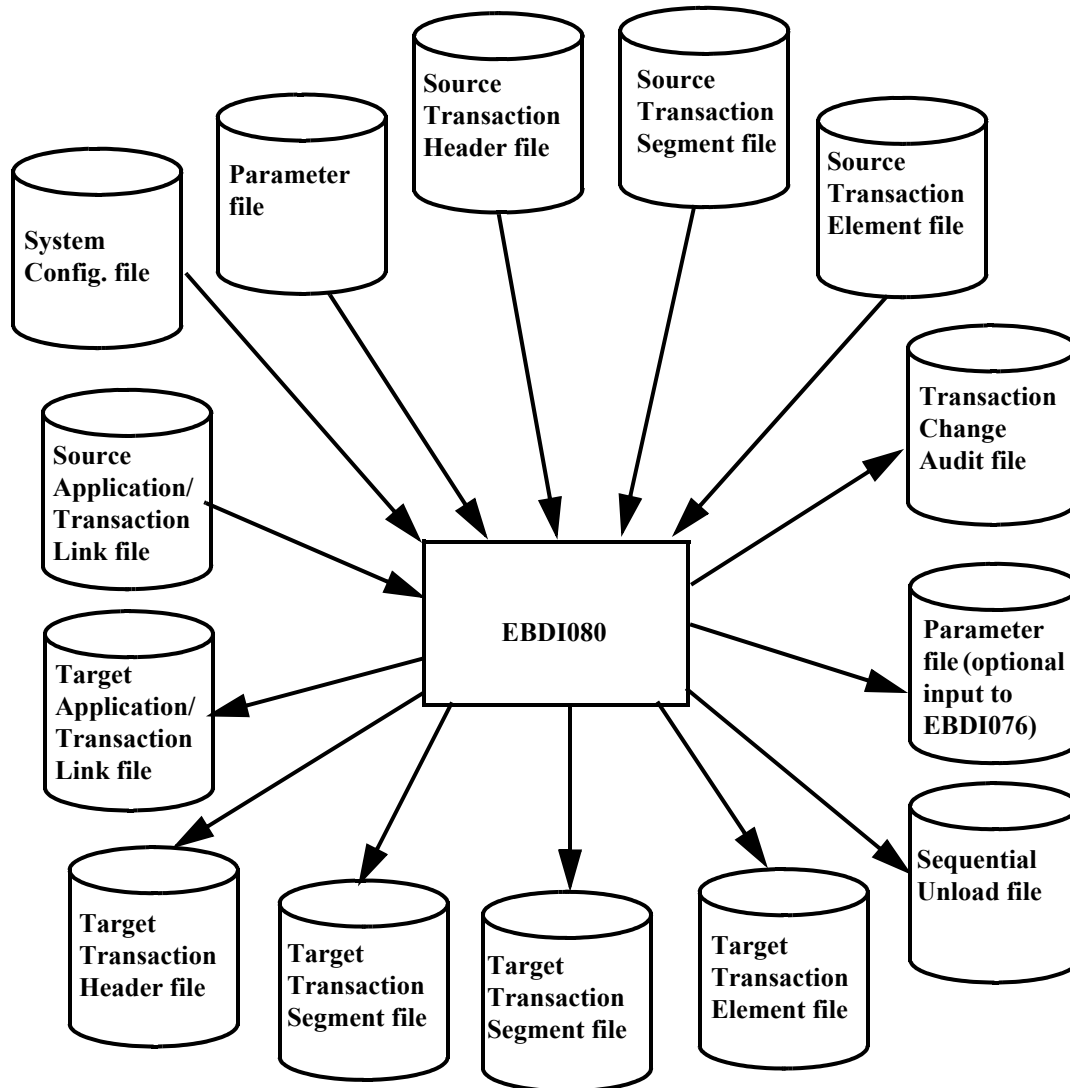
Position	Field Content	COBOL Picture	Format	Length	Description
036 – 036	Send/Receive of delete	X(01)	AN	01	Send or Receive code of transaction being deleted. Valid values: <b>S</b> – Send <b>R</b> – Receive
037 – 042	Filler	X(06)	AN	06	Not used
043 – 043	Delete Switch (Y/N)	X(01)	AN	01	<b>Y</b> – Deletes the existing transaction from the Copy To files.
044 – 080	Spaces	X(37)	AN	37	

*Parameter File Layout (Unload Function)*

Position	Field Content	COBOL Picture	Format	Length	Description
001 – 005	UNLD <b>b</b>	X(05)	AN	05	Constant value of UNLD <b>b</b> .
006 – 015	Copy from Transaction ID	X(10)	AN	10	Copy from Transaction ID of transaction to be copied.
016 – 016	Spaces	X(01)		01	
017 – 017	Copy from S or R	X(01)	AN	01	Copy from Send or Receive Code of transaction to be copied. Valid values: <b>S</b> – Send <b>R</b> – Receive
018 – 024	<del>bbbb</del> TO <b>b</b>	X(07)	AN	07	Constant value of <del>bbbb</del> TO <b>b</b> .
025 – 034	Copy to Transaction ID	X(10)	AN	10	Copy to Transaction ID of transaction to be copied.
035 – 035	Spaces	X(01)	AN	01	
036 – 036	Copy to Send/Receive	X(01)	AN	01	Copy to Send or Receive Code of transaction to be copied. Valid values: <b>S</b> – Send <b>R</b> – Receive
037 – 080	Spaces	X(44)	AN	44	

*Transaction Copy*

Figure 4.20 illustrates how to copy a transaction from one set of VSAM files to another.



**Figure 4.20 Program EBDI080 – Transaction Copy**

---

## Transaction Mapping Definition Re-synch

**EBDI089**

### Purpose

The Transaction Mapping Definition Re-Synch (EBDI089) program is the second program in the re-synch process, run after the Application Sequence Renumbering (EBDI090) program. The program reads the Translation Element file and finds the elements that are mapped, and then re-synchronizes the translation element with the Application field that is mapped to it. The output report contains all counts and errors that may have occurred.

A parameter record must be entered containing the following information: Transaction Definition ID and Send/Receive field. Up to twenty (20) parameter records may be entered. If more than 20 are entered, they are ignored and processing continues.

**The following condition(s) are checked and an appropriate message is written on the output report for each condition:**

If the Application field to be mapped cannot be found on the Application Fields file, the following message is printed on the report along with the element information that was mapped:

\*\*\* W A R N I N G: Application field not found for the following element \*\*\*

The following information is printed below the message:

- Translation element that was mapped (Seg ID and Seq).
- Description of the Translation element field.
- Application field that could not be found.

**Processing terminates if any of the following conditions are true:**

- The Transaction ID or Send/Receive field on the parameter record is left blank.
- The Transaction ID on the parameter record does not exist on the Transaction (TRNHDR) file.
- The application definition that is linked to the Translation does not exist on the Application (APPHDR) file.
- Invalid data is entered on the parameter record.
- All files were not successfully opened.
- An invalid return code is received on a READ or WRITE statement.

If you have Change Audit for Transaction enabled on the "1" record of the EDICFG file, a new record will be written to the Transaction Change Audit file (EDITCHA) for each transaction map that has been through the re-synch process.

## Files Used

### Input

The following files are used as input:

Filename	Description
SYS001	Parameter file This file contains the Translation Definition to be re-synched.
SYS040	Application Header file This file contains all the header records for applications that have been identified to the system.
SYS041	Application Fields file This file contains all the field records for applications that have been identified to the system.
SYS050	Translation Header file This file contains all the header records for translations that have been identified to the system.
EDICFG	System Configuration file

### Input/Output

The following file is used as input and output:

Filename	Description
SYS051	Translation Element file This file contains all the records for translation elements that have been identified to the system.

### Output

The following file is used as output:

Filename	Description
SYS002	Application Sequence Renumbering Report This file contains the report of the Batch Transaction Mapping Definition Re-synch.
EDITCHA	Transaction Change Audit File



*Parameter File Layout*

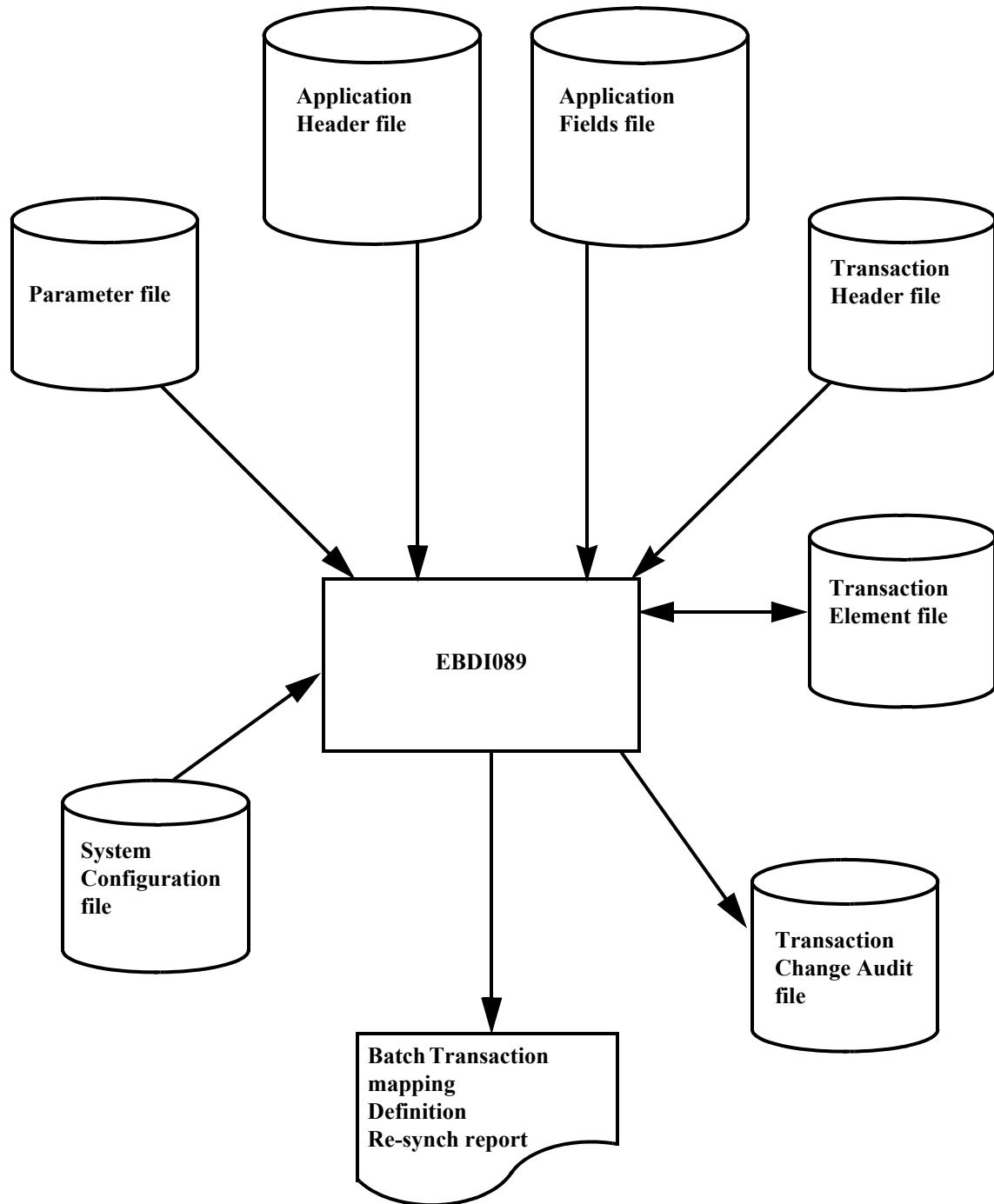
The following table describes the parameter file layout:

Position	Field Content	COBOL Picture	Format	Length	Description
001 – 010	Translation ID	X(10)	AN	10	The name assigned to transaction
011 – 011	Translation Send/Receive	X(01)	AN	01	Valid values: <b>S</b> – Send <b>R</b> – Receive
012 – 080	Filler	X(69)	AN	69	Not used

**Note:** Up to 20 parameter records may be entered. If more than 20 parameter records are entered, they are ignored, and processing continues.

*Transaction Mapping Definition Re-synch*

Figure 4.21 illustrates the Transaction Mapping Definition Re-Synch.



**Figure 4.21 Program EBDI089 – Transaction Mapping Definition Re-synch**

## Sample Report

Figure 4.22 illustrates the Transaction Mapping Definition Re-Synch report.

```

EBDI089          DATE: 12/01/2005          TRANSLATION RE-SYNCH          PAGE 1
GENTRAN MAPPING INTEGRATION          TIME: 12:00

PARAMETER RECORD:
  TRANSLATION ID: BS812      S

***** PROCESSING ENDED ABNORMALLY *****
* TRANSLATION HEADER DOES NOT EXIST      *
* HEADER KEY -> BS812 S                  *
***** JOB CANCELLED *****

*** PROCESSING ENDED ABNORMALLY -- FINAL COUNTS BELOW ***

      FINAL TOTAL:
      TRANSLATION ELEMENTS READ: 0
      TRANSLATION ELEMENTS MAPPED: 0
      APPLICATION FIELD NOT FOUND: 0
      TRANSLATION ELEMENTS UNCHANGED: 0
      TRANSLATION ELEMENTS RE-SYNCHED: 0
      CHANGE AUDIT RECORDS WRITTEN: 0
  
```

**Figure 4.22 Transaction Mapping Definition Re-Synch Report**

## Field Definitions

### Parameter Record

**Translation ID:**

This field displays the translation identifier from the parameter record.

**(Message Field):**

This field displays a message indicating any errors encountered during processing. If no errors are encountered, this field is blank.

**(Message Field):**

This field displays a message indicating either that processing ended abnormally, or the successful completion of re-synchronization.

### Final Totals

**Translation Elements Read:**

This field displays the total number of translation elements read during execution of the program.

**Translation Elements Mapped:**

This field displays the total number of translation elements mapped during execution of the program.

**Application Fields Not Found:**

This field displays the total number of application fields not found during execution of the program.

**Translation Elements Unchanged:**

This field displays the total number of translation elements unchanged during execution of the program.

**Translation Elements Re-Synched:**

This field displays the total number of translation elements that were re-synchronized during execution of the program.

**Change Audit Records Written:**

This field displays the total number of Transaction Change Audit Records written during the execution of this program. This will only be displayed if you have Change Audit for Transaction enabled on the EDICFG file.

# Databank Utility Programs and Reports

## Overview

This chapter contains the following topics related to databank utility programs and reports:

<b>Topic</b>	<b>Page</b>
Outbound Processing Flow.....	5-2
Databank Control Card Format .....	5-3
Online Databank Maintenance .....	5-4
Acknowledgment Reconciliation/Monitor Program ..... EDID510 .....	5-5
Application Databank Inquiry Report ..... EDID551 .....	5-8
EDI Archive Inquiry Program ..... EDID555 .....	5-19
Databank Change Audit Inquiry Program ..... EDID500 .....	5-27
Databank Change Audit Maintenance Program ..... EDID502 .....	5-31
EDI Databank Inquiry Program..... EDID550 .....	5-37
Inbound Application Databank Extract Program ..... EDID405 .....	5-59
Inbound Application Databank Maintenance Program ..... EDID401 .....	5-66
Inbound EDI Databank Maintenance Program ..... EDID301 .....	5-73
Outbound Application Databank Maintenance Program..... EDID101 .....	5-80
Outbound EDI Databank Extract Program..... EDID205 .....	5-87
Outbound EDI Databank Maintenance Program..... EDID201 .....	5-99
Transaction Queue Initialize Program ..... EDID520 .....	5-107
EDI Databank Orphan Record Cleanup Program..... EDIG300 .....	5-109
Program Report Names and Descriptions .....	5-112

## Outbound Processing Flow

Figure 5.1 illustrates the outbound processing flow with deferred enveloping and full databanking. Use the optional program according to your own processing needs. For detailed information on the programs illustrated below, see the appropriate topics in this guide.

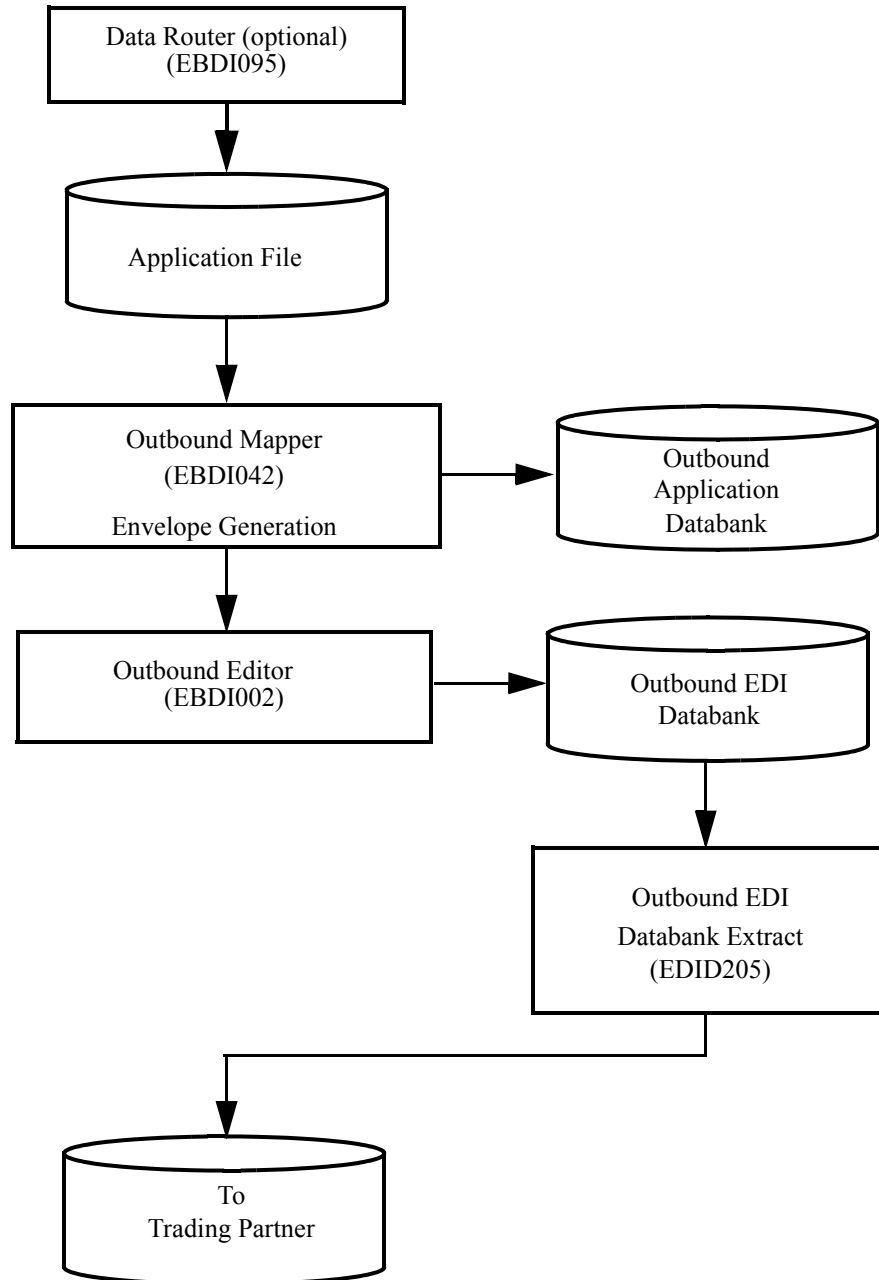


Figure 5.1 Outbound Processing – With Deferred Enveloping and Full Databanking

## Databank Control Card Format

Databank control cards consist of two parts: a request and parameters for the request. In describing control cards, the following terms are used:

- Requested operation – describes the action you want to take (e.g., purge or housekeeping).
- Criteria – defines the limits you set for the requested operation.

Control cards function as overrides to default values contained in the programs. These program defaults are used when either no control cards are specified or when an individual criteria is not included in the control cards. If an empty control card file is supplied, the defaults are used.

Only one control card request can be specified per run.

Databank control cards must adhere to the following rules:

- Comment cards can appear in any order and are denoted by an asterisk in column one.
- All control cards must be in uppercase.
- Each control card request consists of one requested operation and several selection criteria.
- The requested operation must be on the first no-comment card.
- Criteria can consist of any number of variable=value cards.
- The name of the variable begins in column 3 and the value to match begins in column 35. The criteria are specified one per card, and can be in any order.
- Except when noted, the criteria must all be true (i.e., they are “AND’d” together). For example, PARTNER is ABCCORP and APPLICATION-DATA-ID is POFIE.
- All variable names must be in uppercase, with dashes separating multi-word variable names.
- All databank status values and codes must be in uppercase.

## Online Databank Maintenance

Databank maintenance can be performed against documents through the Databank Maintenance system. This online maintenance is only a logical maintenance. A batch program must be run to physically apply the maintenance to the Databank files.

You can apply databank maintenance by running the following programs based on the databank:

Type of Databank File Maintenance	Programs to Run
Outbound application	<p>Run the following programs in the order listed:</p> <ol style="list-style-type: none"> <li>Run the Outbound Mapper program (EBDI042) with the databank Reprocess Switch set to <b>Y</b> in the first Mapping Parameter record to apply the online maintenance.</li> </ol> <p>See the topic “Processing Option – Databanking” in the “Outbound Mapper” section of Chapter 3 for further information.</p> <ol style="list-style-type: none"> <li>Run the Outbound Application Databank Maintenance (EDID101) program to physically delete the document from the Databank file.</li> </ol>
Outbound EDI	<p>Run the following programs in the order listed:</p> <ol style="list-style-type: none"> <li>Run the Outbound EDI Databank Extract (EDID205) program to apply the online maintenance.</li> <li>Run the Outbound EDI Databank Maintenance program (EDID201) to physically delete the document from the Databank file.</li> </ol>
Inbound EDI	<p>Run the following programs in the order listed:</p> <ol style="list-style-type: none"> <li>Run the Inbound Editor (EBDI001) program with the global parameter <b>INBOUND EDI DATABANK REPROCESS</b> to apply the online maintenance.</li> </ol> <p>See the topic “Databank Processing” in the “Inbound Editor” section Chapter 2 for further information.</p> <ol style="list-style-type: none"> <li>Run the Inbound EDI Databank Maintenance (EDID301) program to physically delete the document from the Databank file.</li> </ol>
Inbound application	<p>Run the following programs in the order listed:</p> <ol style="list-style-type: none"> <li>Run the Inbound Application Databank Extract (EDID405) program to apply the online maintenance.</li> <li>Run the Inbound Application Databank Maintenance (EDID401) program to physically delete the document from the Databank file.</li> </ol>



## Acknowledgment Reconciliation/Monitor Program

**EDID510**

The Acknowledgment Reconciliation/Monitor (EDID510) program performs two functions, based on the following control card use:

1. RECONCILE      Reconciliation of incoming acknowledgments with the original outbound documents on the Outbound EDI Databank.
2. MONITOR        Monitoring of expected acknowledgments to determine if the acknowledgments are overdue.

Either or both of these functions can be performed in one run.

### Operation

The Acknowledgment Reconciliation/Monitor runs as a batch job step.

### Files Used

#### *Input*

The following files are used as input:

DDname	Description
EDICFG	System Configuration File
EDIDAR	Acknowledgment Reconciliation File
EDIOEA	Outbound EDI Databank Directory
EDICNTL	Control Card File
ERRCTL	Error Message and Control File

#### *Output*

The following files are used as output:

DDname	Description
EDIOEA	Outbound EDI Databank Directory

#### *Reports*

The EDID510 program produces the following reports:

DDname	Description
EDILOG	Processing Log
EDIRPT1	Reconciliation Report

DDname	Description
EDIRPT2	Expected Acknowledgment Monitoring Report
EDISUM	Summary Report

*Gentran:Viewpoint*

The following files are used in Gentran:Viewpoint only:

DDname	Description
EDIVT	Trace ID File (Tracking Management users only)
EDIVP	Profile File
EDIVSF	Staging Files

**Control Card Options**

The following Acknowledgment Reconciliation/Monitor operations and selection criteria are specified by control cards:

**OPERATIONS**

PROCESS (Default Operation)

**SELECTION CRITERIA**

**Defaults**

MONITOR	YES
MONITOR-HOLIDAY	(no default; valid value required)
MONITOR-LEVEL	NORMAL
MONITOR-REPORT	NEW
MONITOR-SATURDAY	YES
MONITOR-SUNDAY	YES
RECONCILE	YES

**VALID CRITERIA VALUES**

**Monitor**

YES  
NO

**Monitor-Holiday**

A field in which you specify holiday dates that are *not* be considered as business days when monitoring for overdue acknowledgments. This selection criteria may be repeated up to 50 times.

Use this format when specifying dates: MM/DD/YYYY.

**Monitor-Level**

NORMAL

TRANSACTION

**Note:** If only Transaction level information is maintained on the Outbound EDI Databank and Transaction level acknowledgment reconciliation is to be performed, **TRANSACTION** must be specified on the Monitor-Level control card.

**Monitor-Report**

NEW Only report overdue acknowledgments from the run.

ALL Report past overdue acknowledgments and overdue acknowledgments as a result of the run.

**Monitor-Saturday**

YES Saturdays *are* considered business days when monitoring overdue acknowledgments.

NO Saturdays are *not* be considered business days when monitoring overdue acknowledgments.

**Monitor-Sunday**

YES Sundays *are* considered business days when monitoring overdue acknowledgments.

NO Sundays are *not* be considered business days when monitoring overdue acknowledgments.

**Reconcile**

Specifies whether or not inbound acknowledgments should be reconciled against documents on the outbound EDI databank.

YES

NO

## Application Databank Inquiry Report

**EDID551**

The Application Databank Inquiry report is generated when the the EDID551 batch program is executed. The Application Databank Inquiry report provides reporting against the inbound and outbound application databanks. The wide range of selection criteria enables you to produce reports that meet your specific requirements.

### Operation

The Application Databank Inquiry program runs as a batch job step.

### Files Used

#### *Input*

Input varies based on the databank control card criteria used. The following is a list of possible input:

DDname	File Name
EDIIAA	Inbound Application Databank Directory
EDIOAA	Outbound Application Databank Directory
EDICNTL	Control Card File
ERRCTL	Error Message and Control File
EDICFG	System Configuration File

#### *Output*

The following files may be updated when the Reported control card selection criteria is used with a value of NO or ALL:

DDname	Description
EDIIAA	Inbound Application Databank Directory
EDIOAA	Outbound Application Databank Directory

#### *Reports*

The EDID551 program produces the following reports:

DDname	Description
EDIRPT	Inquiry Report
EDILOG	Processing Log
EDISUM	Summary Report

## Control Card Options

The following Databank Inquiry operations and selection criteria are specified by control cards:

### OPERATIONS

SELECT (Default Operation)

### SELECTION CRITERIA

#### Defaults

ACKNOWLEDGEMENT-STATUS	ALL
AGE-DAYS	ALL
AGE-HOURS	ALL
APPLICATION-DATA-ID	ALL
DATABANK	ALL
DATABANK-RUN-NUMBER	ALL
DIVISION	ALL
ENVELOPE-VALIDATION-STATUS	ALL
FROM-DATE	01/01/1900
FROM-TIME	00:00
FUNCTIONAL-GROUP-ID	ALL
GROUP-ENV-REF-ID	ALL
INTERCHANGE-ENV-REF-ID	ALL
MAP-VALIDATION-STATUS	ALL
NETWORK	ALL
PARTNER	ALL
PARTNER-QUALIFIER	ALL
REFERENCE-TAG	ALL
REPORTED	ALL
STATUS	ALL
TEST-PRODUCTION-STATUS	ALL
TO-DATE	12/31/2099
TO-TIME	23:59
TRANSACTION-ENV-REF-ID	ALL
TRANSACTION-SET-ID	ALL
USER	ALL
USER-REFERENCE	ALL

**Note:** Date and time selection criteria are evaluated independently. Time-based selection criteria are used to limit the data selected using the specified time range for each day within the date range. For example, selecting data with time criteria of 9 a.m. to 5 p.m. equates to data processed between 9 and 5 on any day. Adding a date range further reduces the data selected to that which was processed between 9 and 5 on any day in the date range.

**VALID CRITERIA VALUES****Acknowledgment-Status**

**Note:** This selection criteria is used *only* for inbound application searches.

ALL  
ACCEPTED  
ACCEPTED-WITH-ERRORS  
REJECTED

**Age-Days**

The age of the data in number of days. Enter a three-digit value to request documents that are the specified age or newer. Specify 000 to get only documents added on the current date. To request documents older than a specified age, enter a greater than sign and the age-days (e.g., enter >002 to select documents more than 2 days old).

ALL  
nnn (000-999)  
>nnn (where nnn is 000-999)

**Age-Hours**

The age of the data in hours. Enter a three-digit value to request documents that are a specified age or newer. To request documents older than the specified age, enter a greater than sign and the age-hours (e.g., enter >005 to select documents more than five hours old).

ALL  
nnn (000-999)  
>nnn (where nnn is 000-999)

**Application-Data-ID**

ALL  
aaaaaaaaaaaa

**Databank**

Identifies which application databanks should be used for the inquiry.

ALL  
INBOUND-APPLICATION  
OUTBOUND-APPLICATION

**Databank-Run-Number**

An 8-digit number that identifies the run that last processed the document.

ALL  
nnnnnnnn (must include leading zeros)

**Division**

ALL  
aaa (division code)

**Envelope-Validation-Status**

Specifies the envelope validation status as assigned by the Inbound Editor. Only used for inbound application databank search.

ALL

<08, <12, 00, 04, 08, 12, >00, >04, >08

**From-Date**

For outbound application data this value specifies the last date the Outbound Mapper processed the data. For inbound application data this value specifies the date when the data was last output to an application. This could have been performed by either the Inbound Mapper or the Inbound Application Extract program. Starts the range. Used with To-Date.

MM/DD/YYYY

**From-Time**

For outbound application data this value specifies the last time the Outbound Mapper processed the data. For inbound application data this value specifies the time when the data was last output to an application. This could have been performed by either the Inbound Mapper or the Inbound Application Extract program. Starts the range. Used with To-Time.

HH:MM

**Functional-Group-ID**

Specifies the 1- to 6-character functional group identification. You can specify either a Functional Group ID or ALL for all functional groups. This selection criteria is valid only if the Report-Structure control card selection criteria has a value of GROUP.

ALL

aaaaaa (Inbound Application only)

**Group-Env-Ref-ID**

Only used for inbound application search.

aaaaaaaaaaaaa EDIFACT

nnnnnnnnn X12 and TDCC – leading zeros required

ALL

**Interchange-Env-Ref-ID**

Only used for Inbound Application search.

aaaaaaaaaaaaa EDIFACT

nnnnnnnnn X12 and TDCC – leading zeros required

ALL

**Map-Validation-Status**

Specifies the map validation status as assigned by the either the Inbound or Outbound Mapper.

ALL

<08, <12, 00, 04, 08, 12, >00, >04, >08, ALL

**Network**

Specifies the 1- to 15-character network identifier. You can specify either a network identifier or ALL for all networks. This selection criteria is only valid for outbound EDI data.

ALL

aaaaaaaaaaaaaa (Outbound Application only)

**Partner**

ALL or 15- or 35-character partner identification. The length of the field depends on the trading mode. For the Relationship mode, the length is 15 characters. If processing using the Partner/Qualifier mode, the length is 35 characters.

**Partner-Qualifier**

The 4-character Qualifier for the Partner ID. You can specify a Partner/Qualifier or ALL for all Partner/Qualifiers. This is valid only if the trading mode is Partner/Qualifier.

ALL

aaaa

**Reference-Tag**

The Reference Tag assigned to the document by Gentran:Basic.

ALL

aanntnnnnn

**Reported**

This selection criteria is used to limit selection based on what has been reported previously by an Inquiry report.

ALL

Select everything regardless of previous reports.

YES

Select only what has been reported previously.

NO

Select only what has not been reported previously.

**Status**

Specifies whether the data has been output to an application. Can be output by either the Inbound Mapper or the Inbound Application Extract program.

PROCESSED

Has been output to application.

UNPROCESSED

Has never been output to an application.

**Test-Production-Status**

Specifies the test/production status as assigned by Gentran:Basic that should be used for selecting documents.

ALL

PRODUCTION

TEST



**To-Date**

For outbound application data this value specifies the last date the Outbound Mapper processed the data. For inbound application data this value specifies the date when the data was last output to an application. This could have been performed by either the Inbound Mapper or the Inbound Application Extract program. Ends the range. Used with From-Date.

MM/DD/YYYY

**To-Time**

For outbound application data this value specifies the last time the Outbound Mapper processed the data. For inbound application data this value specifies the time when the data was last output to an application. This could have been performed by either the Inbound Mapper or the Inbound Application Extract program. Ends the range. Used with From-Time.

HH:MM

**Transaction-Env-Ref-ID**

This selection criteria is only valid if the Report-Structure control card selection criteria has a value of TRANSACTION. Only used for Inbound Application search.

aaaaaaaaaaaa	EDIFACT
nnnnnnnnnn	X12 and TDCC – leading zeros required
ALL	

**Transaction-Set-ID**

This selection criteria is only used for Inbound Application search.

ALL  
aaaaaa

**User**

This selection criteria is only valid if the trading profile mode Relationship.

ALL  
15-character User ID

**User-Reference**

Specifies the 1- to 45-character user reference value. You can enter either a user reference or ALL for all user references.

ALL  
45-character User Reference field

Example Reports

Figure 4.2 is an example of the Application Databank Inquiry (Summary report).

```

PAGE : 00001
VERSION: 6.3

GENTRAN: BASIC
APPLICATION DATABANK INQUIRY
SUMMARY REPORT

REPORT DATE: 12/01/2005
REPORT TIME: 12:00:00
REPORT ID : EDI151

OPTIONS USED THIS RUN
-----
REQUESTED-OPERATION = SELECT
ACKNOWLEDGEMENT-STATUS = ALL
AGE-DAYS = ALL
AGE-HOURS = ALL
APPLICATION-DATA-ID = ALL
DATABANK = ALL
DATABANK-RUN-NUMBER = ALL
DIVISION = ALL
ENVELOPE-VALIDATION-STATUS = ALL
FROM-DATE = 01/01/1900
FROM-TIME = 00:00
FUNCTIONAL-GROUP-ID = ALL
GROUP-ENV-REF-ID = ALL
INTERCHANGE-ENV-REF-ID = ALL
MAP-VALIDATION-STATUS = ALL
NETWORK = ALL
PARTNER = ALL
PARTNER-QUALIFIER = ALL
REALTIME-DATABANKS = NO
REFERENCE-TAG = ALL
REPORTED = ALL
STATUS = ALL
TEST-PRODUCTION-STATUS = ALL
TO-DATE = 12/31/2099
TO-TIME = 23:59
TRADING-PROFILE-MODE = PARTNER-QUALIFIER
TRANSACTION-ENV-REF-ID = ALL
TRANSACTION-SET-ID = ALL
USER-REFERENCE = ALL
CONCURRENCY-ENABLED = NO

PROCESSING SUMMARY
-----
INBOUND DOCUMENTS : 6
OUTBOUND DOCUMENTS : 4
INBOUND DOCUMENTS SELECTED : 6
OUTBOUND DOCUMENTS SELECTED : 4
NUMBER OF ERRORS THIS RUN : 0
HIGHEST RETURN CODE THIS RUN : 0
    
```

Figure 4.2 Application Databank Inquiry (Summary Report)

Figure 4.3 is an example of the Application Databank Inquiry (Processing Log).

```
REPORT DATE: 12/01/2005
REPORT TIME: 12:00:00
REPORT ID   : EDI149

GENTRAN
APPLICATION DATABANK INQUIRY
PROCESSING LOG

PAGE   : 00001
VERSION: 6.4
(12/01/2005)

MESSAGES
-----
EDI-42301-I 00 APPLICATION DATABANK INQUIRY PROCESSING BEGINS . . . DATE: 12/01/2005, TIME: 09:08:36
EDI-42302-I 00 APPLICATION DATABANK INQUIRY PROCESSING ENDS DATE: 12/01/2005, TIME: 09:08:40
```

**Figure 4.3 Application Databank Inquiry (Processing Log)**

Figure 4.4 is an example of the Application Databank Inquiry (Inquiry report).

REPORT DATE: 12/01/2005		PAGE : 00001				
REPORT TIME: 12:00:00		VERSION: 6.4				
REPORT ID : EDI150						
GENTRAN APPLICATION DATABANK INQUIRY INQUIRY REPORT						
OUTBOUND APPLICATION DATABANK						
PARTNER	USER REFERENCE	APPLICATION DATA ID	REFERENCE TAG	DATE	MAPPING TIME	ST
VENDOR-1	PONUMBER-001	POFILE	0A00000001	12/01/2005	12:00	00
VENDOR-2	PONUMBER-002	POFILE	0A00000002	12/01/2005	12:00	00
VENDOR-3	PONUMBER-003	POFILE	0A00000003	12/01/2005	12:00	00
VENDOR-4	PONUMBER-004	POFILE	0A00000004	12/01/2005	12:00	00

Figure 4.4 Application Databank Inquiry (Inquiry Report)

## Field Definitions

### Summary Report

**Options Used this Run**

This section of the report displays all the parameter options in effect for EDI Databank Inquiry. Only those parameter options used are displayed.

**Processing Summary****Inbound Documents:**

This field displays the total number of inbound documents.

**Outbound Documents:**

This field displays the total number of outbound documents.

**Inbound Documents Selected:**

This field displays the total number of inbound documents selected.

**Outbound Documents Selected:**

This field displays the total number of outbound documents selected.

**Number of Errors this Run:**

This field displays the total number of errors that occurred on this run of the report.

**Highest Return Code this Run:**

This field displays the highest return code occurring on this run of the report.

### Processing Log

**Messages:**

Displays error and system messages pertaining to EDI Databank Inquiry processing.

### Inquiry Report

**Partner:**

Displays the partner identification you have entered or selected.

**Partner Qual:**

Displays the Partner/Qualifier.

**User Reference:**

Displays the User Reference for the transaction. Depending on whether the transaction is inbound or outbound and how the user has chosen to implement the User Reference options in Data Mapping this field is created for the transaction by either Editor processing or Data Mapping processing.

**Application Data ID:**

A 12-character alphanumeric field entered to restrict the list to one application data identification for listing documents.

**User (Relationship only):**

Displays the User ID valid for trading profile relationship mode.

**Reference Tag:**

Displays the Reference Tag Gentrans:Basic has assigned to the transaction/document.

**Mapping Date:**

Displays the date and time the transaction was last mapped. Format is MM/DD/YYYY.

**Mapping Time:**

Displays the date and time the transaction was last mapped. Format is HH:MM.

**Mapping St:**

Displays the current Mapping Status for the document.

00	No errors were encountered during the mapping process for the document.
04	One or more requirement or relational condition errors were detected.
08	One or more unknown or unexpected segment errors were detected.
12	An inbound document could not be translated.

## EDI Archive Inquiry Program

**EDID555**

The EDI Archive Inquiry report is generated when the EDID555 batch program is executed. The EDI Archive Inquiry report lists the databank run number and other databank information pertaining to the trading partner. A wide range of selection criteria enables you to produce reports that meet your specific requirements. These requirements may be based on the needs of the user viewing the reports, or they may be based on what information is stored on the directories by the Inbound and Outbound Editors.

In addition, the report also produces the Archive Report Records file that consists of two sections, inbound and outbound.

### Operation

The EDI Archive Inquiry report runs as a batch job step.

### Files Used

#### *Input*

These files are used as input:

DDname	File Name
EDIARCH	EDI Databank Archive File
EDICNTL	Parameter File
EDICFG	System Configuration File
ERRCTL	Error Message and Control File

#### *Output*

This file is used as input:

DDname	Description
ARCHOUT	Archive Records

#### *Reports*

The EDID555 program produces the following reports:

DDname	Description
EDIRPT	Archive Inquiry Report
EDISUM	Summary Report
EDILOG	Processing Log Report

## Control Card Options

The following Databank Inquiry operations and selection criteria are specified by control cards:

### OPERATIONS

SELECT (Default Operation)

### SELECTION CRITERIA

#### Defaults

DATABANK	INBOUND
DATABANK-RUN-NUMBER	ALL
FROM-DATE	01/01/1900
FROM-TIME	00:00
GROUP-ENV-REF-ID	ALL
INTERCHANGE-ENV-REF-ID	ALL
NETWORK	ALL
NETWORK-STATUS	ALL
PARTNER	ALL
PARTNER-QUALIFIER	ALL
TO-DATE	12/31/2099
TO-TIME	23:59
TRANSACTION-ENV-REF-ID	ALL
USER	ALL

### VALID CRITERIA VALUES

#### Databank

Identifies which databanks should be used for the inquiry.

INBOUND	Select data from the inbound archive EDI databank
OUTBOUND	Select data from the outbound archive EDI databank

#### Databank-Run-Number

An 8-digit number that identifies the run that last processed the document.

ALL  
nnnnnnnn (must include leading zeros)

#### From-Date

For inbound EDI data, this value specifies the inbound edited date. For outbound EDI data, this value specifies the date when the data was last output to communications. This could have been performed by either the Outbound Editor or the Outbound EDI Extract programs. Starts the range. Used with To-Date.

ALL  
MM/DD/YYYY



**From-Time**

For inbound EDI data, this value specifies the inbound edited time. For outbound EDI data, this value specifies the time when the data was last output to communications. This could have been performed by either the Outbound Editor or the Outbound EDI Extract program. Starts the range. Used with To-Time.

ALL  
HH:MM

**Group-Env-Ref-ID**

This selection criteria is only valid if the Report-Structure control card selection criteria has a value of GROUP.

aaaaaaaaaaaa	EDIFACT
nnnnnnnnnn	X12 and TDCC – leading zeros required
ALL	All group envelope reference IDs

**Interchange-Env-Ref-ID**

This selection criteria is only valid if the Report-Structure control card selection criteria has a value of INTERCHANGE.

aaaaaaaaaaaa	EDIFACT
nnnnnnnnnn	X12 and TDCC – leading zeros required
ALL	All interchange envelope reference IDs

**Network**

Specifies the 1- to 15-character network identifier. You can specify either a network identifier or ALL for all networks. This selection criteria is only valid for outbound EDI data.

**Network Status**

This selection criteria is only valid if the Report-Structure control card selection criteria has a value of INTERCHANGE. This selection criteria is only valid for outbound EDI data.

ALL  
ERRORS-ENCOUNTERED  
MANUALLY-COMPLETED  
NOT-REQUIRED  
PARTNER-PICKED-UP  
RECEIVED-NETWORK  
SORTED-NETWORK  
WAITING

**Partner**

ALL or 15- or 35-character partner identification. The length of the field depends on the trading mode. For the Relationship mode, the length is 15 characters. If processing using the Partner/Qualifier mode, the length is 35 characters.

**Partner Qualifier**

The 4-character Qualifier for the Partner ID. You can specify a Partner/Qualifier or ALL for all Partner/Qualifiers. This is valid only if the trading mode is Partner/Qualifier.

**To-Date**

For inbound EDI data this value specifies the Inbound Edited Date. For outbound EDI data this value specifies the date when the data was last output to communications. This could have been performed by either the Outbound Editor or the Outbound EDI Extract program. Ends the range. Used with From-Date.

ALL  
MM/DD/YYYY

**To-Time**

For inbound EDI data this value specifies the Inbound Edited Time. For outbound EDI data this value specifies the time when the data was last output to communications. This could have been performed by either the Outbound Editor or the Outbound EDI Extract program. Ends the range. Used with From-Time.

ALL  
HH:MM

**Transaction-Env-Ref-ID**

This selection criteria is only valid if the Report-Structure control card selection criteria has a value of TRANSACTION.

aaaaaaaaaaaaa	EDIFACT
nnnnnnnnn	X12 and TDCC – leading zeros required
ALL	All transaction envelope reference IDs

**User**

Specifies the 1- to 15-character user identification. You can specify either a User ID or ALL for all users. This selection criteria is valid only if processing in the Relationship trading mode.

Example Report

Figure 4.5 is an example of the EDI Archive Inquiry report

```

REPORT DATE: 12/01/2005                GENTRAN: BASIC                PAGE : 00001
REPORT TIME: 09:32:39                 EDI ARCHIVE INQUIRY           VERSION: 6.4
REPORT ID : EDI152                   PROCESSING LOG                 COMPILE DATE: 12/01/05

  MESSAGES
  -----

EDI-042101-I 00 EDI DATABANK INQUIRY PROCESSING BEGINS . . . DATE: 12/01/2005, TIME: 09:32:40
EDI-042102-I 00 EDI DATABANK INQUIRY PROCESSING ENDS . . . DATE: 12/01/2005, TIME: 09:32:40
REPORT DATE: 12/01/2005                GENTRAN: BASIC                PAGE : 00001
REPORT TIME: 09:32:39                 EDI ARCHIVE INQUIRY           VERSION: 6.4
REPORT ID : EDI154                   SUMMARY REPORT

  OPTIONS USED THIS RUN
  -----

DATABANK                               = INBOUND
DATABANK-RUN-NUMBER                    = ALL
FROM-DATE                              = 01/02/1900
FROM-TIME                              = 00:00
GROUP-ENV-REF-ID                       = ALL
INTERCHANGE-ENV-REF-ID                 = ALL
PARTNER                                = ALL
PARTNER-QUALIFIER                      = ALL
TO-DATE                                = 12/31/2099
TO-TIME                                = 23:59
TRADING-PROFILE-MODE                   = PARTNER-QUALIFIER
TRANSACTION-ENV-REF-ID                 = ALL

  PROCESSING SUMMARY
  -----

INBOUND ARCHIVES READ                  :          1
OUTBOUND ARCHIVES READ                 :          0

INBOUND ARCHIVES SELECTED              :          8
  INTERCHANGES                        :          1
  GROUPS                               :          1
  TRANSACTIONS                         :          6

OUTBOUND ARCHIVES SELECTED             :          0
  INTERCHANGES                        :          0
  GROUPS                               :          0
  TRANSACTIONS                         :          0

REPORT RECORDS WRITTEN                 :          8

NUMBER OF ERRORS THIS RUN              :          0
HIGHEST RETURN CODE THIS RUN          :          0
    
```

Figure 4.5 EDI Archive Inquiry Report

## *Field Definitions*

### **Summary Report**

#### **Options Used This Run:**

This section of the report displays all parameter options in effect for the EDI Archive Databank Inquiry. Only those parameter options used are displayed.

#### **Processing Summary:**

##### **Inbound Archive Read**

This field displays the total number of inbound archive records read.

##### **Outbound Archive Read**

This field displays the total number of outbound archive records read.

##### **Inbound Archives Selected**

This field displays the total number of inbound records selected.

##### **Interchanges**

This field displays the total number inbound interchanges selected.

##### **Groups**

This field displays the total number of inbound groups selected.

##### **Transactions**

This field displays the total number of inbound transactions selected.

##### **Outbound Archives Selected**

This field displays the total number of outbound records selected.

##### **Interchanges**

This field displays the total number of outbound interchanges selected.

##### **Groups**

This field displays the total number of outbound groups selected.

##### **Transactions**

This field displays the total number of outbound transactions selected.

##### **Report Records Written**

This field displays the total number of sequential archive report records written.

##### **Number of Errors This Run**

This field displays the total number of errors that occurred during processing the report.

##### **Highest Return Code This Run**

This field displays the highest return code occurring on this run of the report.

Archive Report Records File

Figure 4.6 illustrates the EDI Archive Report Records file for inbound.

```

REPORT DATE: 12/01/2005                GENTRAN: BASIC                PAGE : 00001
REPORT TIME: 09:32:39                  EDI ARCHIVE INQUIRY
REPORT ID : EDI153                      **** INBOUND ****                VERSION: 6.4
DATABANK RUN NUMBER: 00000004

                                     **** COUNTS ****
TYPE PARTNER          ENVELOPE     GROUP/   LOADED   OUTPUT   GROUP/
                     QUAL REFERENCE ID  TRAN ID  DATE     DATE     TRANS  CHARACTERS
I  VENDOR-1           000000005  IN      12/01/2005 12/01/2005 000001 0000006330
G  VENDOR-1           000000007  IN      12/01/2005 12/01/2005 000006 0000006208
T  VENDOR-1           000070001  810     12/01/2005 12/01/2005 000000 0000001011
T  VENDOR-1           000070002  810     12/01/2005 12/01/2005 000000 0000002035
T  VENDOR-1           000070003  810     12/01/2005 12/01/2005 000000 0000003057
T  VENDOR-1           000070004  810     12/01/2005 12/01/2005 000000 0000004083
T  VENDOR-1           000070005  810     12/01/2005 12/01/2005 000000 0000005111
T  VENDOR-1           000070006  810     12/01/2005 12/01/2005 000000 0000006135
*** MESSAGE STORE AVAILABLE ***

REPORT DATE: 12/01/2005                GENTRAN: BASIC                PAGE : 00001
REPORT TIME: 09:32:39                  EDI ARCHIVE INQUIRY
REPORT ID : EDI152                      PROCESSING LOG                   VERSION: 6.4
                                     COMPILE DATE: 12/01/05

MESSAGES
-----

EDI-042101-I 00 EDI DATABANK INQUIRY PROCESSING BEGINS . . . DATE: 12/01/2005, TIME: 09:32:40
EDI-042102-I 00 EDI DATABANK INQUIRY PROCESSING ENDS . . . DATE: 12/01/2005, TIME: 09:32:40
REPORT DATE: 12/01/2005                GENTRAN: BASIC                PAGE : 00001
REPORT TIME: 09:32:39                  EDI ARCHIVE INQUIRY
REPORT ID : EDI154                      SUMMARY REPORT                   VERSION: 6.4

OPTIONS USED THIS RUN
-----

DATABANK                      = INBOUND
DATABANK-RUN-NUMBER           = ALL
FROM-DATE                     = 01/02/1900
FROM-TIME                     = 00:00
GROUP-ENV-REF-ID              = ALL
INTERCHANGE-ENV-REF-ID        = ALL
PARTNER                       = ALL
PARTNER-QUALIFIER             = ALL
TO-DATE                       = 12/31/2099
TO-TIME                       = 23:59
TRADING-PROFILE-MODE          = PARTNER-QUALIFIER
TRANSACTION-ENV-REF-ID        = ALL

PROCESSING SUMMARY
-----

INBOUND ARCHIVES READ         :      12
OUTBOUND ARCHIVES READ        :         0

INBOUND ARCHIVES SELECTED     :         8
  INTERCHANGES                :         1
  GROUPS                       :         1
  TRANSACTIONS                 :         6

OUTBOUND ARCHIVES SELECTED    :         0
  INTERCHANGES                :         0
  GROUPS                       :         0
  TRANSACTIONS                 :         0

REPORT RECORDS WRITTEN        :         8

NUMBER OF ERRORS THIS RUN     :         0
HIGHEST RETURN CODE THIS RUN :         0
    
```

Figure 4.6 EDI Archive Inquiry Records Report – Inbound

*File Contents*

The following table provides information about the fields in the Archive Report Records file.

<b>Position</b>	<b>Length</b>	<b>Format</b>	<b>Field Name</b>
001 – 008	8	N	Databank Run Number
009 – 009	1	AN	Databank Type
<b>The following fields apply when in Relationship mode on the databank:</b>			
010 – 024	15	AN	User ID
025 – 039	15	AN	Partner ID
040 – 048	9	--	Unused
<b>The following fields apply when in Partner/Qualifier mode on the databanks:</b>			
10 – 44	35	AN	Partner
45 – 48	4	AN	Partner/Qualifier
<b>The following fields apply to all modes:</b>			
49 – 62	14	AN	Envelope Reference ID
63 – 68	6	AN	Group/Transaction ID
69 – 76	8	AN	Loaded Date
77 – 84	8	AN	Output Date
85 – 90	6	AN	Group/Transaction Count
91 – 100	10	AN	Character Count
101 – 108	8	AN	Network ID (Outbound only)

## Databank Change Audit Inquiry Program

**EDID500**

The Databank Change Audit Inquiry program (EDID500) enables you to produce reports for databank maintenance updates that have been applied to any of the four databanks. This program produces either a summary level report of activity or a detail report that shows all update activity, including before and after images of any “edit” updates.

### Operation

The Databank Change Audit Inquiry program runs as a batch job step.

### Files Used

#### *Input*

Input for the Databank Change Audit Inquiry program varies based on the databank control card selection criteria value specified. The following table is a list of possible input:

<b>DDname</b>	<b>Description</b>
EDIACA	Inbound Application Databank Change Audit
EDIIECA	Inbound EDI Databank Change Audit
EDIOACA	Outbound Application Databank Change Audit
EDIOECA	Outbound EDI Databank Change Audit
EDICNTL	Control Card File
ERRCTL	Error Message and Control File
EDICFG	System Configuration File

#### *Output*

None.

#### *Reports*

Program EDID500 produces the following reports:

<b>DDname</b>	<b>Description</b>
EDIRPT	Databank Change Audit Inquiry Report
EDILOG	Databank Change Audit Processing Log
EDISUM	Databank Change Audit Summary Report

## Control Card Options

The following Databank Change Audit Inquiry operations and selection criteria are specified by control cards:

### OPERATIONS

SELECT (Default Operation)

### SELECTION CRITERIA

#### Defaults

BYPASS	ALL
DATABANK	ALL
DATABANK-RUN-NUMBER	ALL
FROM-UPDATE-APPLIED-DATE	01/01/1900
FROM-UPDATE-APPLIED-TIME	00:00
FROM-UPDATE-ONLINE-DATE	01/01/1900
FROM-UPDATE-ONLINE-TIME	00:00
PARTNER	ALL
PARTNER-QUALIFIER	ALL
REFERENCE-TAG	ALL
REPORT-TYPE	SUMMARY
TO-UPDATE-APPLIED-DATE	12/31/2099
TO-UPDATE-APPLIED-TIME	23:59
TO-UPDATE-ONLINE-DATE	12/31/2099
TO-UPDATE-ONLINE-TIME	23:59
UPDATE	ALL
USER	ALL
USER-REFERENCE	ALL

### VALID CRITERIA VALUES

#### Bypass

Indicates online updates that were bypassed due to errors.

ALL	Select all updates including bypassed updates.
NO	Select just updates that were not bypassed.
YES	Select just updates that were bypassed.

#### Databank

Identifies which Databank Change Audit files should be searched. Up to four databank control cards can be specified.

ALL
INBOUND-EDI
INBOUND-APPLICATION
OUTBOUND-APPLICATION
OUTBOUND-EDI



**Databank-Run-Number**

An 8-digit number that identifies the run that last processed the document.

ALL

nnnnnnnn (must include leading zeros)

**From-Update-Applied-Date**

Specifies the date that the update was first applied by batch to the databank. Starts the range. Used with To-Update-Applied-Date.

MM/DD/YYYY

**From-Update-Applied-Time**

Specifies the time that the update was first applied by batch to the databank. Starts the range. Used with To-Update-Applied-Time.

HH:MM

**From-Update-Online-Date**

Specifies the date that a user last updated the record through the online Databank Maintenance facility. Starts the range. Used with To-Update-Online-Date.

MM/DD/YYYY

**From-Update-Online-Time**

Specifies the time that a user last updated the record through the online Databank Maintenance facility. Starts the range. Used with To-Update-Online-Time.

HH:MM

**Partner**

A 15- to 35-character partner identification. The length of the field depends on the trading mode. For the Relationship mode, the length is 15 characters. If processing using the Partner/Qualifier mode, the length is 35 characters.

ALL

a . . . (35) . . . a (Partner/Qualifier mode)

a . . . (15) . . . a (Relationship mode)

**Partner-Qualifier**

The 4-character Qualifier for the Partner ID. You can specify a Partner/Qualifier or ALL for all Partner/Qualifiers. This is valid only if the trading mode is Partner/Qualifier.

ALL

aaaa

**Reference-Tag**

The 10-character Reference Tag that has been assigned to the document/transaction by Gentran:Basic.

ALL

OAnnnnnnnn (for Outbound Application)

OEnnnnnnnn (for Outbound EDI)

IEnnnnnnnn (for Inbound EDI)

IAnnnnnnnn (for Inbound Application)

**Report-Type**

Identifies the level at which the Inquiry should report.

FULL	Report includes detailed information for each update.
SUMMARY	Summary list of updates.

**To-Update-Applied-Date**

Specifies the date that the update was first applied by batch to the databank. Ends the range. Used with From-Update-Applied-Date.

MM/DD/YYYY

**To-Update-Applied-Time**

Specifies the time that the update was first applied by batch to the databank. Ends the range. Used with From-Update-Applied-Time.

HH:MM

**To-Update-Online-Date**

Specifies the date that a user last performed the update via the online Databank Maintenance facility. Ends the range. Used with From-Update-Online-Date.

MM/DD/YYYY

**To-Update-Online-Time**

Specifies the time that a user last performed the update via the online Databank Maintenance facility. Ends the range. Used with From-Update-Online-Time.

HH:MM

**Update**

Specifies the type of update that should be searched for.

ALL  
ACKNOWLEDGE  
DELETE  
EDIT  
NETWORK  
RESET

**User**

ALL or a 15-character user identification. Only valid if trading mode is specified as Relationship in the system configuration file.

**User-Reference**

Specifies the 1- to 45-character user reference value. You can enter either a user reference or ALL for all user references.

## Databank Change Audit Maintenance Program

EDID502

The Databank Change Audit Maintenance program (EDID502) maintains the Databank Change Audit files. The program functions include deleting, archiving, and restoring change audit information.

The EDI databank change audit files contain records of these functions: manual acknowledgments, network acknowledgments, resets/reprocesses, and deletions. The application databank change audit files contain records of edits, resets/reprocesses, and deletions.

### Operation

The Databank Change Audit Maintenance program operates in the following general modes:

- **Periodic Housekeeping**  
On a periodic basis (anywhere from daily to monthly depending on volume and retention requirements) the Housekeeping operation deletes and optionally archives data based on specified retention periods.
- **Ad Hoc Maintenance (Purge and Restore)**  
In addition to the regularly scheduled housekeeping runs, the Databank Change Audit Maintenance's Purge and Restore functions deletes and optionally archives selected data and restores data from archives.

### Processing

The Databank Change Audit Maintenance program processes documents using the following three operations:

- Housekeeping
- Purge
- Restore

#### *Housekeeping Operation*

If the Housekeeping operation is requested, Databank Change Audit Maintenance deletes and optionally archives the change audit records that exceed the retention periods specified in the control cards. An option of the Housekeeping operation is to archive all selected change audit records to a sequential output file.

#### *Purge Operation*

If the Purge operation is requested, change records are removed if they match ALL criteria specified. An option of the Purge operation is to archive selected change audit records to a sequential output file.

#### *Restore Operation*

If the Restore operation is requested, change audit records are restored from the Archive file to the Change Audit if the records match ALL selection criteria specified.

## Files Used

### Input

Control cards and the four Databank Change Audit files are the principal input to Databank Change Audit Maintenance.

DDname	Description
EDIARCH	Change Audit Archive File
EDIACA	Inbound Application Databank Change Audit
EDIIECA	Inbound EDI Databank Change Audit
EDIOACA	Outbound Application Databank Change Audit
EDIOECA	Outbound EDI Databank Change Audit
EDICNTL	Control Card File
ERRCTL	Error Message and Control File
EDICFG	System Configuration File

**Note:** The databank files used for the run are determined by Control Card Selection Criteria. See the control card DATABANK for available options.

### Output

The following files are used as output:

DDname	Description
EDIARCH	Change Audit Archive File
EDIACA	Inbound Application Databank Change Audit
EDIIECA	Inbound EDI Databank Change Audit
EDIOACA	Outbound Application Databank Change Audit
EDIOECA	Outbound EDI Databank Change Audit

### Reports

Program EDID502 produces the following reports:

DDname	Description
EDIAUDT	Change Audit Maintenance Report
EDILOG	Processing Log

DDname	Description
EDISUM	Processing Summary Report

**Control Card Options**

*OPERATIONS*

HOUSEKEEPING (Default Operation)  
 PURGE  
 RESTORE

*SELECTION CRITERIA*

**Defaults**

For all operations:

DATABANK ALL

For Housekeeping operation:

RETENTION-DAYS-UPDATE-APPLIED 030  
 RETENTION-DAYS-UPDATE-ONLINE 030

For Purge and Restore operations:

BYPASS ALL  
 DATABANK-RUN-NUMBER ALL  
 FROM-UPDATE-APPLIED-DATE 01/01/1900  
 FROM-UPDATE-APPLIED-TIME 00:00  
 FROM-UPDATE-ONLINE-DATE 01/01/1900  
 FROM-UPDATE-ONLINE-TIME 00:00  
 PARTNER ALL  
 PARTNER-QUALIFIER ALL  
 REFERENCE-TAG ALL  
 TO-UPDATE-APPLIED-DATE 12/31/2099  
 TO-UPDATE-APPLIED-TIME 23:59  
 TO-UPDATE-ONLINE-DATE 12/31/2099  
 TO-UPDATE-ONLINE-TIME 23:59  
 UPDATE ALL  
 USER ALL  
 USER-REFERENCE ALL

For Purge and Housekeeping operations:

ARCHIVE NO

*VALID CRITERIA VALUES*

**Archive**

This controls whether an archive is to be created for deleted change records.

NO

YES

**Bypass**

Controls removal of updates that were bypassed due to errors.

ALL

Select all updates including bypassed updates.

NO

Select just updates that were not bypassed.

YES

Select just updates that were bypassed.

**Databank**

Identifies which Databank Change Audit files should be processed. Up to four databank control cards can be specified.

ALL

INBOUND-EDI

INBOUND-APPLICATION

OUTBOUND-APPLICATION

OUTBOUND-EDI

**Databank-Run-Number**

An 8-digit number that identifies the run that last processed the document.

ALL

nnnnnnnn (must include leading zeros)

**From-Update-Applied-Date**

Specifies the date that the update was first applied by batch to the databank. Starts the range. Used with To-Update-Applied-Date.

MM/DD/YYYY

**From-Update-Applied-Time**

Specifies the time that the update was first applied by batch to the databank. Starts the range. Used with To-Update-Applied-Time.

HH:MM

**From-Update-Online-Date**

Specifies the date that a user last performed the update via the online Databank Maintenance facility. Starts the range. Used with To-Update-Online-Date.

MM/DD/YYYY

**From-Update-Online-Time**

Specifies the time that a user last performed the update via the online Databank Maintenance facility. Starts the range. Used with To-Update-Online-Time.

HH:MM

**Partner**

A 15- to 35-character partner identification. The length of the field depends on the trading mode. For the Relationship mode, the length is 15 characters. If processing using the Partner/Qualifier mode, the length is 35 characters.

ALL  
a . . . . (35) . . . . a (Partner/Qualifier mode)  
a . . . (15) . . . a (Relationship mode)

**Partner-Qualifier**

The 4-character Qualifier for the Partner ID. You can specify a Partner/Qualifier or ALL for all Partner/Qualifiers. This is valid only if the trading mode is Partner/Qualifier.

ALL  
aaaa

**Reference-Tag**

The 10-character Reference Tag that has been assigned to the document/transaction by Gentran:Basic.

ALL  
OAnnnnnnnn (for Outbound Application)  
OEnnnnnnnn (for Outbound EDI)  
IEnnnnnnnn (for Inbound EDI)  
IANnnnnnnn (for Inbound Application)

**Retention-Days-Update-Applied**

Specifies the number of days that have elapsed since the update was applied via batch.

000 – 999

**Retention-Days-Update-Online**

Specifies the number of days that have elapsed since the update was applied via online.

000 – 999

**To-Update-Applied-Date:**

Specifies the date when the update was first applied by batch to the databank. Ends the range. Used with From-Update-Applied-Date.

MM/DD/YYYY

**To-Update-Applied-Time**

Specifies the time when the update was first applied by the batch to the databank. Ends the range. Used with From-Update-Applied-Time.

HH:MM

**To-Update-Online-Date:**

Specifies the date when a user last performed the update via the online Databank Maintenance facility. Ends the range. Used with From-Update-Online-Date.

MM/DD/YYYY

**To-Update-Online-Time:**

Specifies the time when a user last performed the update by means of the online Databank Maintenance facility. Ends the range. Used with From-Update-Online-Time.

HH:MM

**Update**

Specifies the type of update that should be searched for.

ALL

RESET

EDIT

ACKNOWLEDGE

DELETE

NETWORK

**User**

ALL or a 15-character user identification. Valid only if the trading mode is specified as Relationship.

**User-Reference**

Specifies the 1- to 45-character user reference value. You can enter either a user reference or ALL for all user references.



## EDI Databank Inquiry Program

**EDID550**

The EDI Databank Inquiry report is generated when the batch program EDID550 is executed. The EDI Databank Inquiry report contains a detailed inquiry, summary level inquiry, or processing log, based on the report-type control card value specified. This report displays information from the Inbound and Outbound EDI Databanks. A wide range of selection criteria enables you to produce reports that meet your specific requirements. These requirements may be based on the needs of the user viewing the reports, or they may be based on what information is stored on the directories by the Inbound and Outbound Editors.

An option of the EDI Databank Inquiry program is the Report-Structure control card. This control card allows reporting based on:

- EDI Interchange level Directory information with all related group and transaction level information.
- EDI Interchange level Directory information only.
- EDI Group level Directory information with all related transaction level information.
- EDI Group level Directory information only.
- EDI Transaction level Directory information.

The EDI Databank Inquiry program also produces a Statistics file if requested. The EDI Databank Statistics file is a data file containing statistical information on inbound and outbound EDI data.

For Gentran:Plus users, the INS-TRADANET network status information is printed at the interchange level on the Summary report.

### Operation

The EDI Databank Inquiry report runs as a batch job step.

### Files Used

#### *Input*

Input varies based on the databank control card criteria used. The following is a list of possible input:

DDname	File Name
EDIIEA	Inbound EDI Databank Directory
EDIOEA	Outbound EDI Databank Directory
SYS090	Partner Profile
EDICNTL	Control Card File
ERRCTL	Error Message and Control File
EDICFG	System Control File

*Output*

The following databank files may be updated if the Reported control card selection criteria is used with a value of NO or ALL:

DDname	File Name
EDIIEA	Inbound EDI Databank Directory
EDIOEA	Outbound EDI Databank Directory
EDISTX	EDI Databank Statistics File

*Reports*

Program EDID550 produces the following reports:

DDname	Description
EDIRPT	Inquiry Report
EDILOG	Processing Log
EDISUM	Summary Report

**Control Card Options**

The following Databank Inquiry operation and selection criteria are specified by control cards.

**OPERATIONS**

SELECT (Default Operation)

**SELECTION CRITERIA**

**Defaults**

ACKNOWLEDGEMENT-STATUS	ALL
AGE-DAYS	ALL
AGE-HOURS	ALL
APPLICATION-DATA-ID	No default, must specify
DATABANK	ALL
DATABANK-RUN-NUMBER	ALL
DIVISION	ALL
ENVELOPE-VALIDATION-STATUS	ALL
FROM-DATE	01/01/1900
FROM-TIME	00:00
FUNCTIONAL-GROUP-ID	ALL
FUNCTIONAL-GROUP-ID-2	No default, must specify
FUNCTIONAL-GROUP-ID-3	No default, must specify
FUNCTIONAL-GROUP-ID-4	No default, must specify

GROUP-ENV-REF-ID	ALL
INTERCHANGE-ENV-REF-ID	ALL
NETWORK	ALL
NETWORK-STATUS	ALL
PARTNER	ALL
PARTNER-QUALIFIER	ALL
REFERENCE-TAG	ALL
REPORTED	ALL
REPORT-DEFERRED	NO
REPORT-STRUCTURE	INTERCHANGE
REPORT-TYPE	SUMMARY
STATISTICS-FILE	NO
STATUS	ALL
TEST-PRODUCTION-STATUS	ALL
TO-DATE	12/31/2099
TO-TIME	23:59
TRANSACTION-ENV-REF-ID	ALL
TRANSACTION-SET-ID	ALL
TRANSACTION-SET-ID-2	No default, must specify
TRANSACTION-SET-ID-3	No default, must specify
TRANSACTION-SET-ID-4	No default, must specify
USER	ALL
USER-DUPLICATE-IND	ALL
USER-REFERENCE	ALL

**Note:** Date and time selection criteria are evaluated independently. Time-based selection criteria are used to limit the data selected using the specified time range for each day within the date range. For example, selecting data with time selection criteria of 9 a.m. to 5 p.m. selects data processed between 9 and 5 on any day. Adding a date range further reduces the data selected to that which was processed between 9 and 5 on any day in the date range.

**VALID CRITERIA VALUES**

**Acknowledgement-Status**

- ALL
- ACCEPTED
- ACCEPTED-WITH-ERRORS
- INDETERMINATE
- NOT-REQUIRED
- OVERDUE
- PARTIAL-ACCEPTANCE
- REJECTED

MANUALLY-ACKNOWLEDGED

WAITING

RECEIVED

**Note:** The Report Structure parameter affects what information is reported when the Acknowledgment Status parameter is used. EDID550 evaluates only the acknowledgment status values on databank directory records that are at the same level (interchange, group, transaction) as the Report Structure parameter.

For example, if you are using a Report Structure of Interchange and the Acknowledgment Status parameter is set with a value of Overdue, only those databank records that have an acknowledgment status of overdue at the interchange level will be reported. If you wish to see databank records that are overdue at the group level, you must use a Report Structure of Group.

### Age-Days

The age of the data in number of days. Enter a three-digit value to request documents that are the specified age or newer. Specify 000 to get only documents added on the current date. To request documents older than a specified age, enter a greater than sign and the age-days (e.g., enter >002 to select documents more than two days old).

ALL

nnn (000-999)

>nnn (where nnn is 000-999)

### Age-Hours

The age of the data in number of hours. Enter a three-digit value to request documents that are a specified age or newer. To request documents older than the specified age, enter a greater than sign and the age-hours (e.g., enter >005 to select documents more than five hours old).

ALL

nnn (000-999)

>nnn (where nnn is 000-999)

### Application-Data-ID

Specifies the 1- to 12-character application identification. You can specify either an Application Data ID or ALL for all application data. This criteria is only valid if the Report-Structure control card has a value of TRANSACTION. It is also only used for selection of outbound data.

### Databank

Identifies which EDI Databanks should be used for the inquiry.

ALL	Select data from both inbound and outbound EDI databanks.
INBOUND-EDI	Select data from the inbound EDI databank.
OUTBOUND-EDI	Select data from the outbound EDI databank.

**Databank-Run-Number**

An 8-digit number that identifies the run that last processed the document. You can specify either a databank run number (which must include leading zeros) or ALL for all databank runs.

**Division**

Specifies the 3-character division code. You can specify either a division code or ALL for all division codes.

**Envelope-Validation-Status**

Specifies the envelope validation status as assigned by the Inbound Editor. Used only for Inbound EDI data.

ALL

<08, <12, 00, 04, 08, 12, >00, >04, >08

**From-Date**

For inbound EDI data, this value specifies the inbound Edited Date. For outbound EDI data, this value specifies the date when the data was last output to communications. This could have been performed by either the Outbound Editor or the Outbound EDI Extract program. Starts the range. Used with To-Date.

MM/DD/YYYY

**From-Time**

For inbound EDI data, this value specifies the inbound Edited Time. For outbound EDI data, this value specifies the time when the data was last output to communications. This could have been performed by either the Outbound Editor or the Outbound EDI Extract program. Starts the range. Used with To-Time.

HH:MM

**Functional-Group-ID**

Specifies the 1- to 6-character functional group identification. You can specify either a Functional Group ID or ALL for all functional groups. This selection criteria is valid only if the Report-Structure control card selection criteria has a value of GROUP.

**Functional-Group-ID-2**

Specifies the 1- to 6-character functional group identification. This parameter is used in conjunction with the Functional-Group-ID parameter if you want to select a second functional group for your report. A valid functional group ID must be specified. A value of ALL is invalid. This selection criteria is valid only if the Report-Structure control card selection criteria has a value of GROUP.

**Functional-Group-ID-3**

Specifies the 1- to 6-character functional group identification. This parameter is used in conjunction with the Functional-Group-ID parameter if you want to select a third functional group for your report. A valid functional group ID must be specified. A value of ALL is invalid. This selection criteria is valid only if the Report-Structure control card selection criteria has a value of GROUP.

**Functional-Group-ID-4**

Specifies the 1- to 6-character functional group identification. This parameter is used in conjunction with the Functional-Group-ID parameter if you want to select a fourth functional group for your report. A valid functional group ID must be specified. A value of ALL is invalid. This selection criteria is valid only if the Report-Structure control card selection criteria has a value of GROUP.

**Group-Env-Ref-ID**

This selection criteria is only valid if the Report-Structure control card selection criteria has a value of GROUP.

aaaaaaaaaaaa	EDIFACT
nnnnnnnnnn	X12 and TDCC – leading zeros required
ALL	All group envelope reference IDs

**Interchange-Env-Ref-ID**

This selection criteria is only valid if the Report-Structure control card selection criteria has a value of INTERCHANGE.

aaaaaaaaaaaa	EDIFACT
nnnnnnnnnn	X12 and TDCC – leading zeros required
ALL	All interchange envelope reference IDs

**Network**

Specifies the 1- to 15-character network identifier. You can specify either a network identifier or ALL for all networks. This selection criteria is only valid for Outbound EDI data.

**Network-Status**

This selection criteria is only valid if the Report-Structure control card selection criteria has a value of INTERCHANGE. This selection criteria is only valid for Outbound EDI Data.

ALL  
 ERRORS-ENCOUNTERED  
 MANUALLY-COMPLETED  
 NOT-REQUIRED  
 PARTNER-PICKED-UP  
 RECEIVED-NETWORK  
 SORTED-NETWORK  
 WAITING

**Partner**

ALL or 15- or 35-character partner identification. The length of the field depends on the trading mode. For the Relationship mode, the length is 15 characters. If processing using the Partner/Qualifier mode, the length is 35 characters.

**Partner-Qualifier**

The 4-character Qualifier for the Partner ID. You can specify a Partner/Qualifier or ALL for all Partner/Qualifiers. This is valid only if the trading mode is Partner/Qualifier.

**Reference-Tag**

This selection criteria is only valid if the Report-Structure control card selection criteria has a value of TRANSACTION. A 10-character Reference Tag assigned to the document/transaction by Gentran:Basic.

ALL	All reference tags
aannnnnnn	A reference tag, where <i>aa</i> =IE, IA, OE, or OA. Leading zeros must be specified for the 8-digit number.

**Reported**

This selection criteria is used to limit selection based on what has been reported previously by an Inquiry report.

ALL	Selects everything regardless of previous reports and updates the reported status on the databank.
YES	Selects only what has been reported previously.
NO	Selects only what has not been reported previously and updates the reported status on the databank.
IGNORE	Selects everything regardless of previous reports and does not perform the databank update.

**Report-Deferred**

Controls whether the Inquiry Program reports outbound EDI data which was deferred and unprocessed.

YES	Yes, print deferred and nondeferred transactions.
NO	No, do not print deferred transactions, but print nondeferred transactions.
ONLY	Print deferred transactions only.

**Report-Structure**

Specifies how the EDI Inquiry Report produced should be structured.

INTERCHANGE	All Groups and Transactions within Interchanges
INTERCHANGE ONLY	Interchanges only
GROUP	All Transaction within Groups
GROUP ONLY	Groups only
TRANSACTION	All Transactions only

**Report-Type**

Identifies the level at which the Inquiry should report.

FULL	Detailed information for each interchange, group and transaction that is selected is reported.
SUMMARY	Summary list of selected interchanges, groups and transactions.

**Statistics-File**

Specifies whether the EDI Databank Statistics file is to be created.

YES	Yes, create the Statistics file, in addition to the inquiry reports.
NO	No, do not create the Statistics file.
ONLY	Create the Statistics file only, and do not print the inquiry reports.

**Status**

Specifies whether the data has been output to communications. Can be output by either the Outbound Editor or the Outbound EDI Extract program.

PROCESSED	Has been output to communications.
UNPROCESSED	Has never been output to communications.

**Test-Production-Status**

Specifies the test/production status as assigned by Gentran:Basic that should be used for selecting documents.

ALL	Selects all data, whether test or production.
PRODUCTION	Selects the production data.
TEST	Selects the test data.

**To-Date**

For inbound EDI data this value specifies the Inbound Edited Date. For outbound EDI data this value specifies the date when the data was last output to communications. This could have been performed by either the Outbound Editor or the Outbound EDI Extract program. Ends the range. Used with From-Date.

MM/DD/YYYY

**To-Time**

For inbound EDI data this value specifies the Inbound Edited Time. For Outbound EDI data this value specifies the time when the data was last output to communications. This could have been performed by either the Outbound Editor or the Outbound EDI Extract program. Ends the range. Used with From-Time.

HH:MM

**Transaction-Env-Ref-ID**

This selection criteria is only valid if the Report-Structure control card selection criteria has a value of TRANSACTION.

aaaaaaaaaaaa	EDIFACT
nnnnnnnn	X12 and TDCC – leading zeros required
ALL	All transaction envelope reference IDs

**Transaction-Set-ID**

Specifies the 1- to 6-character transaction set identifier. You can specify either a Transaction Set ID or ALL for all transaction sets. This selection criteria is only valid if the Report-Structure control card has a value of TRANSACTION.

**Transaction-Set-ID-2**

Specifies the 1- to 6-character transaction set identifier. This parameter is used in conjunction with the Transaction-Set-ID parameter if you want to select a second transaction set for your report. A valid transaction set ID must be specified. A value of ALL is invalid. This selection criteria is valid only if the Report-Structure control card selection criteria has a value of TRANSACTION.



**Transaction-Set-ID-3**

Specifies the 1- to 6-character transaction set identifier. This parameter is used in conjunction with the Transaction-Set-ID parameter if you want to select a third transaction set for your report. A valid transaction set ID must be specified. A value of ALL is invalid. This selection criteria is valid only if the Report-Structure control card selection criteria has a value of TRANSACTION.

**Transaction-Set-ID-4**

Specifies the 1- to 6-character transaction set identifier. This parameter is used in conjunction with the Transaction-Set-ID parameter if you want to select a fourth transaction set for your report. A valid transaction set ID must be specified. A value of ALL is invalid. This selection criteria is valid only if the Report-Structure control card selection criteria has a value of TRANSACTION.

**User**

Specifies the 1- to 15-character user identification. You can specify either a User ID or ALL for all users. This selection criteria is valid only if processing in the Relationship trading mode.

**User-Duplicate-Ind**

Specifies whether a transaction has been marked as a user duplicate. This selection criteria is valid only if the Report-Structure control card has a value of TRANSACTION. It is also only used for selection of outbound EDI data.

YES	Selects duplicate transactions.
NO	Selects non-duplicate transactions.
ALL	Selects all transactions, duplicate or non-duplicate.

**User-Reference**

Specifies the 1- to 45-character user reference value. You can specify either a user reference or ALL for all user references. This selection criteria is only valid if the Report-Structure control card selection criteria has a value of TRANSACTION.

Example Reports

Figure 4.7 is an example of the EDI Databank Inquiry Summary report.

```

PAGE : 00001
VERSION: 6.3

GENTRAN: BASIC
EDI DATABANK INQUIRY
SUMMARY REPORT

REPORT DATE: 12/01/2005
REPORT TIME: 12:00:00
REPORT ID : EDI144

OPTIONS USED THIS RUN
-----
REQUESTED-OPERATION = SELECT
ACKNOWLEDGEMENT-STATUS = ALL
AGE-DAYS = ALL
AGE-HOURS = ALL
DATABANK = ALL
DATABANK-RUN-NUMBER = ALL
DIVISION = ALL
ENVELOPE-VALIDATION-STATUS = ALL
FROM-DATE = 01/01/1900
FROM-TIME = 00:00
NETWORK = ALL
PARTNER = ALL
PARTNER-QUALIFIER = ALL
REALTIME-DATABANKS = NO
REPORTED = YES
REPORT-DEFERRED = INTERCHANGE
REPORT-STRUCTURE = SUMMARY
REPORT-TYPE = YES
STATISTICS-FILE = ALL
STATUS = ALL
TEST-PRODUCTION-STATUS = ALL
TO-DATE = 12/31/2099
TO-TIME = 23:59
TRADING-PROFILE-MODE = PARTNER-QUALIFIER
INTERCHANGE-ENV-REF-ID = ALL
NETWORK-STATUS = ALL
USER-DUPLICATE-IND = ALL
CONCURRENCY-ENABLED = NO

PROCESSING SUMMARY
-----
INBOUND INTERCHANGES READ : 1
OUTBOUND INTERCHANGES READ : 4

INBOUND DATA SELECTED
INTERCHANGES : 1
GROUPS : 1
TRANSACTIONS : 6

OUTBOUND DATA SELECTED
INTERCHANGES : 4
GROUPS : 4
TRANSACTIONS : 4

STATISTICS FILE RECORDS WRITTEN : 23
NUMBER OF ERRORS THIS RUN : 0
HIGHEST RETURN CODE THIS RUN : 0
    
```

Figure 4.7 EDI Databank Inquiry (Summary Report)

Figure 4.8 is an example of the EDI Databank Inquiry Processing Log.

```

REPORT DATE: 12/01/2005
REPORT TIME: 12:00:00
REPORT ID : EDI142

GENTRAN: BASIC
EDI DATABANK INQUIRY
PROCESSING LOG

PAGE . . . 0001
VERSION . 6.4
(12/01/2005)

MESSAGES
-----
EDI-042101-I 00 EDI DATABANK INQUIRY PROCESSING BEGINS . . . DATE: 12/01/2005, TIME: 12:00:00
EDI-042102-I 00 EDI DATABANK INQUIRY PROCESSING ENDS . . . DATE: 12/01/2005, TIME: 12:00:00
    
```

**Figure 4.8 EDI Databank Inquiry (Processing Log)**

Figure 4.9 is an example of the EDI Databank Inquiry Detail Inquiry report.

REPORT DATE: 12/01/2005	GENTRAN: BASIC	PAGE : 00001
REPORT TIME: 12:00:00	EDI DATABANK INQUIRY	VERSION: 6.4
REPORT ID : EDI143	DETAIL INQUIRY REPORT	
OUTBOUND EDI DATABANK		
INTERCHANGE ID : : VENDOR-1	QUAL :	ACKNOWLEDGE STATUS :
NAME . . . . . :		NETWORK STATUS :
ENV REF ID . . . . . : 000000000000004	EDITED : 12/01/2005 12:00	GROUPS : 000001
DIVISION . . . . . : 000	OUTPUT : 12/01/2005 12:00	TRANSACTIONS : 000001
NETWORK . . . . . :	ACKNOWLEDGE :	CHARACTERS : 0000001744
DATABANK RUN NR: 000000001		
GROUP ID . . . . . : VENDOR-1	QUAL :	ACKNOWLEDGE STATUS :
NAME . . . . . :		TRANS COUNT : 000001
ENV REF ID . . . . . : 000000000000004	ACKNOWLEDGE :	TRANS ACKNOWLEDGED : 000000
DIVISION . . . . . : 000		TRANS ACCEPTED : 000000
FUNC GROUP ID : PO		CHARACTERS : 0000001622
TRANS ID . . . . . : VENDOR-1	QUAL :	ACKNOWLEDGE STATUS :
SET ID ENV REF ID USER REFERENCE		TRANS COUNT : 000001
850 000000000000004 FONUMBER-001		TRANS ACKNOWLEDGED : 000000
		TRANS ACCEPTED : 000000
		CHARACTERS : 0000001622
INTERCHANGE ID : : VENDOR-2	QUAL :	ACKNOWLEDGE STATUS :
NAME . . . . . :		NETWORK STATUS :
ENV REF ID . . . . . : 000000000000004	EDITED : 12/01/2005 12:00	GROUPS : 000001
DIVISION . . . . . : 000	OUTPUT : 12/01/2005 12:00	TRANSACTIONS : 000001
NETWORK . . . . . :	ACKNOWLEDGE :	CHARACTERS : 0000001441
DATABANK RUN NR: 000000001		
GROUP ID . . . . . : VENDOR-2	QUAL :	ACKNOWLEDGE STATUS :
NAME . . . . . :		TRANS COUNT : 000001
ENV REF ID . . . . . : 000000000000004	ACKNOWLEDGE :	TRANS ACKNOWLEDGED : 000000
DIVISION . . . . . : 000		TRANS ACCEPTED : 000000
FUNC GROUP ID : PO		CHARACTERS : 0000001319
TRANS ID . . . . . : VENDOR-2	QUAL :	ACKNOWLEDGE STATUS :
SET ID ENV REF ID USER REFERENCE		TRANS COUNT : 000001
850 000000000000004 FONUMBER-002		TRANS ACKNOWLEDGED : 000000
		TRANS ACCEPTED : 000000
		CHARACTERS : 0000001319
INTERCHANGE ID : : VENDOR-3	QUAL :	ACKNOWLEDGE STATUS :
NAME . . . . . :		NETWORK STATUS :
ENV REF ID . . . . . : 000000000000004	EDITED : 12/01/05 12:00	GROUPS : 000001
DIVISION . . . . . : 000	OUTPUT : 12/01/05 12:00	TRANSACTIONS : 000001
NETWORK . . . . . :	ACKNOWLEDGE :	CHARACTERS : 0000001644
DATABANK RUN NR: 000000001		
GROUP ID . . . . . : VENDOR-3	QUAL :	ACKNOWLEDGE STATUS :
NAME . . . . . :		TRANS COUNT : 000001
ENV REF ID . . . . . : 000000000000004	ACKNOWLEDGE :	TRANS ACKNOWLEDGED : 000000
DIVISION . . . . . : 000		TRANS ACCEPTED : 000000
FUNC GROUP ID : PO		CHARACTERS : 0000001522
TRANS ID . . . . . : VENDOR-3	QUAL :	ACKNOWLEDGE STATUS :
SET ID ENV REF ID USER REFERENCE		TRANS COUNT : 000001
850 000000000000004 FONUMBER-3		TRANS ACKNOWLEDGED : 000000
		TRANS ACCEPTED : 000000
		CHARACTERS : 0000001522

Figure 4.9 EDI Databank Inquiry (Detail Inquiry Report)

Figure 4.10 is an example of the EDI Databank Inquiry Summary Inquiry report.

REPORT DATE: 12/01/2005		GENTRAN: BASIC		PAGE : 00001		
REPORT TIME: 12:00:00		EDI DATABANK INQUIRY		VERSION: 6.4		
REPORT ID : EDI143		SUMMARY INQUIRY REPORT				
		OUTBOUND EDI DATABANK				
DIR	ENVELOPE	EDITED	OUTPUT	STATUS	GROUP/	USER
TYPE PARTNER	REFERENCE ID	DATE/TIME	DATE/TIME	NET ACK COMM	TRANS	CHARACTER DUF
	QUAL					CHARACTER DUF
I	VENDOR-1	000000004	12/01/2005 12:00	12/01/2005 12:00	000000	0000001744 N
G	VENDOR-1	000000004			000001	0000001622 N
T	VENDOR-1	000000004				0000001573 N
I	VENDOR-2	000000004	12/01/2005 12:00	12/01/2005 12:00	000000	0000001441 N
G	VENDOR-2	000000004			000001	0000001319 N
T	VENDOR-2	000000004				0000001265 N
I	VENDOR-3	000000004	12/01/2005 12:00	12/01/2005 12:00	000000	0000001644 N
G	VENDOR-3	000000004			000001	0000001522 N
T	VENDOR-3	000000004				0000001470 N
I	VENDOR-4	000000004	12/01/2005 12:00	12/01/2005 12:00	000000	0000001427 N
G	VENDOR-4	000000004			000001	0000001305 N
T	VENDOR-4	000000004				0000001253 N

Figure 4.10 EDI Databank Inquiry (Summary Inquiry Report)

## Field Definitions

### Summary Report

**Options Used this Run:**

This section of the report displays all the parameter options in effect for EDI Databank Inquiry. Only those parameter options used are displayed.

**Processing Summary:****Inbound Interchanges Read:**

This field displays the total number of inbound interchanges read.

**Outbound Interchanges Read:**

This field displays the total number of outbound interchanges read.

**Inbound Data Selected Interchanges:**

This field displays the total number of inbound interchanges selected.

**Inbound Data Selected Groups:**

This field displays the total number of inbound groups selected.

**Inbound Data Selected Transactions:**

This field displays the total number of inbound transactions selected.

**Outbound Data Selected Interchanges:**

This field displays the total number of outbound interchanges selected.

**Outbound Data Selected Groups:**

This field displays the total number of outbound groups selected.

**Outbound Data Selected Transactions:**

This field displays the total number of outbound transactions selected.

**Number of Errors this Run:**

This field displays the total number of errors that occurred on this run of the report.

**Highest Return Code this Run:**

This field displays the highest return code occurring on this run of the report.

### Processing Log

**Messages:**

Displays error and system messages pertaining to EDI Databank Inquiry processing.

### Detail Inquiry Report

The Detail Inquiry report is produced when option FULL is specified for parameter REPORT-TYPE.

**Interchange ID:**

Displays the interchange identification. The Interchange ID is either a telephone number identifying a trading partner (if using the Commerce:Network), or the Partner ID for the trading partner (if using Gentran:Basic). The system displays up to 10 Interchange IDs starting with the Interchange ID specified.

**Partner Qual:**

Displays the Partner/Qualifier.

**Acknowledge Status:**

This field is used to restrict the list to functional groups with a specific acknowledgment Status.

A	Accepted
E	Accepted with errors
I	Indeterminate
N	Not Required
O	Overdue acknowledgment
P	Partial acceptance
R	Rejected
U	Manually Acknowledged
W	Waiting acknowledgment
4	Rejected (Edifact only)
7	Acknowledged (Edifact only)
8	Received (Edifact only)

**Name:**

This field displays a text description of the interchange ID.

**Network Status:**

This field is used to display which Network Status type was used to build the list. If this field is left blank all Network Status types are displayed.

E	Errors were encountered during network processing
N	As defined by the partner profile that was used to process this interchange, network tracking is not performed for this interchange.
P	Partner has picked up the interchange
R	Network has received the interchange
S	Network has sorted (processed) the interchange
U	User has manually set to a network completed status
W	Interchange is waiting for a response from the Network

**Env Ref ID:**

Displays the envelope reference identification for the interchange.

**Edited:**

Displays the last date (in MM/DD/YYYY format) and time (in HH:MM format) that the interchange was last edited.

**Groups:**

Displays the total number of groups processed.

**Division:**

Displays the Division defined to the partner profile which was used to process the Groups being listed.

**Output:**

For outbound interchanges, the date and time the interchange was output from the translation system is displayed. For inbound interchanges, the date and time the interchange was loaded to the translation system is displayed.

**Transactions:**

Displays the total number of transaction processed.

**Network:**

The name of the network that processed this interchange (if applicable), as specified in the trading partner profile.

**Acknowledge:**

Displays the date and time acknowledgment reconciliation was performed against the transaction.

**Characters:**

Displays the total number of characters processed.

**Databank Run Nr:**

Displays the Databank Run number which was assigned to the interchange by Gentran:Basic.

**Proc:**

Displays if the transaction has been processed or has been output to communications.

**User Dup:**

Displays if the Outbound EDI transaction was marked as a user duplicate.

**Summary Inquiry Report**

The Summary Inquiry report is produced when option SUMMARY is specified for parameter REPORT-TYPE.

**Dir Type:**

Indicates the type of record being displayed from the EDI Databank Directory. Valid values are: I (Interchange), G (Group), T (Transaction).

**Partner:**

Displays the partner identification you have entered or selected.

**Partner Qual:**

Displays the Partner/Qualifier.

**Envelope Reference ID:**

Displays the envelope reference identification for the interchange.

**Edited Date/Time:**

Displays the last date (in MM/DD/YYYY format) and time (in HH:MM format) that the interchange was last edited.

**Output Date/Time:**

For outbound interchanges, the date and time the interchange was output from the translation system is displayed. For inbound interchanges, the date and time the interchange was loaded to the translation system is displayed.



**Status Net:**

Displays the current network status for the interchange.

E	Errors were encountered during network processing
N	As defined by the partner profile that was used to process this interchange, network tracking is not perform for this interchange.
P	Partner has received the interchange
R	Network has received the interchange
S	Network has sorted (processed) the interchange
U	User has manually set to a network completed status
W	Waiting acknowledgment
4	Rejected (Edifact only)
7	Acknowledged (Edifact only)
8	Received (Edifact only)

**Status Ack:**

Displays the current acknowledgment status for the interchange.

A	Accepted
E	Accepted with errors
I	Indeterminate
N	Not Required
O	Overdue acknowledgment
P	Partial acceptance
R	Rejected
U	Manually Acknowledged
W	Waiting acknowledgment

**Status Comm:**

If Gentrans:Plus is installed, this field displays the first 5 VSAM Batch Status flags for the interchange.

See your Gentrans:Plus documentation for further information regarding status flags.

**Counts Group/Trans:**

Displays the total number of groups/transactions processed.

**Counts Character:**

Displays the total number of characters processed.

**User Dup:**

Displays if the outbound EDI transaction was marked as a user duplicate.

**EDI Databank Statistics File (EDISTX)**

The EDI Databank Statistics file is an optional data file that contains statistical information on inbound and outbound EDI data. Its purpose is to provide statistical and performance information in a machine-readable format. The EDI Databank Statistics file is produced by the EDI Databank Inquiry program (EDID550).

The EDI Databank Statistics file contains character and segment counts for each transaction, functional group, interchange, and summary record for the inquiry run. Statistical information cannot be generated for unprocessed deferred transactions.

The level of detail report in the EDI Databank Statistics file can be tailored to meet your requirements by setting the EDI Databank Inquiry parameters.

### *Logical Structure*

The EDI Databank Statistics file contains the following types of logical data records:

- Transaction level
- Functional group level
- Interchange level
- Run level

**Transaction level records** consist of data that identifies the *class* of the transaction (e.g., user, partner, network, application), as well as character and segment counts. There is one Transaction Level record per EDI Databank Directory Transaction record processed.

**Functional group level records** consist of data that identifies the *class* of the functional group, as well as character and segment counts for the entire functional group, including character and segment counts attributable to the functional group envelopes. There is one Functional Group Level record per EDI Databank Directory Group record processed.

**Interchange level records** consist of data that identifies the *class* of the interchange, as well as character and segment counts for the entire interchange, including character and segment counts attributable to interchange envelopes. There is one Interchange Level record per EDI Databank Directory Interchange record processed.

**Run level records** consist of character, segment, and record counts for the entire inquiry run. There is one run level record for each Databank Directory file (e.g., inbound and outbound) processed and one run level record to summarize the entire inquiry run.

### *Physical Structure*

The EDI Databank Statistics file is a physical, sequential file that contains records created in one run of the EDI Databank Inquiry program. The file consists of fixed-length records and is 512 bytes in length.

The EDI Databank Statistics file contains one record for each *item* that occurs in the run. *Items* include: transactions, functional groups, interchanges, and the inquiry run itself.

The EDI Databank Statistics file consists of one physical record layout that is used for all item types. The contents of the record are produced for the indicated item (e.g., for inbound/outbound, I = interchange, G = group, T = transaction, and R = run).

*Record Sequence*

The file is sequenced as the inbound or outbound EDI data is processed. For example:

- inbound transaction #1
- inbound transaction #2
- inbound functional group #1
- inbound transaction #3
- inbound transaction #4
- inbound functional group #2
- inbound interchange #1
- inbound run totals
- outbound transaction #1
- outbound transaction #2
- outbound functional group #1
- outbound interchange #1
- outbound run totals
- inquiry run totals

*Record Contents*

The following record description lists the fields on the Statistics file and the item for which they apply. The item types are denoted by the following abbreviations: I = interchange, G = group, T = transaction, R = run, \*\* = not available.

Position	Length	Format	Field Name	Inbound	Outbound
001 – 001	1	AN	Direction Indicator	All	All
002 – 002	1	AN	Statistics Level	All	All
003 – 042	40	AN	Profile ID	I,G,T	I,G,T
<b>The following fields apply when in Relationship mode on the databank:</b>					
003 – 003	1	--	Unused		
004 – 018	15	AN	User ID	I,G,T	I,G,T
019 – 033	15	AN	Partner ID	I,G,T	I,G,T
034 – 042	9	--	Unused		
<b>The following fields apply when in Partner/Qualifier mode on the databanks:</b>					
003 – 003	1	--	Unused		
004 – 038	35	AN	Partner	I,G,T	I,G,T
039 – 042	4	AN	Partner Qualifier	I,G,T	I,G,T
<b>The following fields apply to all modes:</b>					
043 – 056	14	AN	Control Number	I,G,T	I,G,T

Position	Length	Format	Field Name	Inbound	Outbound
057 – 071	15	AN	Network ID	**	I,G,T
072 – 081	10	AN	Application Data ID	T	T
082 – 082	1	AN	Test Production Status	I,G,T	I,G,T
083 – 127	45	AN	User Reference	T	T
128 – 135	8	N	Envelope Date (YYYYMMDD)	**	I,G
136 – 143	8	N	Envelope Time (HHMMSSHS)	**	I,G
144 – 151	8	N	Processed Date (YYYYMMDD)	I,G,T	I,G,T
152 – 159	8	N	Processed Time (HHMMSSHS)	I,G,T	I,G,T
160 – 167	8	AN	Databank Run Number	I,G,T	I,G,T
168 – 175	8	BIN	Number of Characters	All	All
176 – 183	8	BIN	Number of Segments	All	All
184 – 191	8	BIN	Total Sets	G,I,R	G,I,R
192 – 199	8	BIN	Total Groups	I,R	I,R
200 – 207	8	BIN	Total Interchanges	R	R
208 – 213	6	AN	Transaction Set ID	T	T
214 – 225	12	AN	Compliance Version	I,G,T	I,G,T
226 – 231	6	AN	Functional Group ID	G,T	G,T
232 – 311	80	AN	User Defined	**	T
312 – 323	12	AN	Envelope Version	**	G
324 – 358	35	AN	Envelope Sender ID	**	G
359 – 393	35	AN	Envelope Receiver ID	**	G
394 – 396	3	AN	Envelope Agency	**	G
397 – 512	116	AN	Future Use		

### Field Descriptions

#### Direction Indicator

The direction of the item. Valid values are:

- I Inbound
- O Outbound
- blank Both inbound and outbound (Run Totals)

**Statistics Level**

The level of the statistics record. Valid values are:

1	Inquiry Run level total
2	Databank Run level total
3	Interchange level
4	Functional Group level
5	Transaction level

**Profile ID**

The User ID for the item, or the Partner ID and Partner/Qualifier for the item.

**User ID**

The user identification for the item.

**Partner ID**

The partner identification for the item.

**Partner (Relationship Mode)**

The partner identification for Partner/Qualifier trading mode. It is used with User/Partner for relationship mode.

**Partner Qualifier**

The Partner/Qualifier for the item.

**Control Number**

A 1- to 14-character code containing the control number sent or received in the associated envelope segment.

**Network ID**

The Network ID that identifies the source or destination of the item.

**Application Data ID**

A 1- to 10-character identifier for the application that is the originating or receiving destination for the transaction.

**Test Production Status**

The test production status obtained from the trading profiles used to process the item.

T	Test
P	Production

**User Reference**

The user reference of the item.

**Envelope Date**

The date (in the format YYYYMMDD) on which the item was enveloped.

**Envelope Time**

The time (in the format HHMMSS) when the item was enveloped.

**Processed Date**

The date (in the format YYYYMMDD) on which the item was processed.

**Processed Time**

The time (in the format HHMMSS) when the item was processed.

**Databank Run Number**

A number assigned by the system that identifies when the item was added to the databank.

**Number of Characters**

The number of characters in the associated item.

**Number of Segments**

The number of segments in the associated item.

**Total Sets**

The total number of transaction sets in the item.

**Total Groups**

The total number of functional groups in the item.

**Total Interchanges**

The total number of interchanges selected in the run.

**Transaction Set ID**

The 6-character identifier of the EDI transaction set.

**Compliance Version**

A 1- to 12-character code associated with the compliance version.

**Compliance Agency**

A 3-character agency code containing the version used to perform compliance checking against the EDI document.

**Functional Group ID**

The 6-character identifier of the functional group.

**User Defined**

The 80-character application data link information extracted from the User Field (UF) application field type. This field contains customer-specific application information.

**Envelope Version**

The 12-character envelope version identification for the GS08, UNG07, UNH02, and STX01 elements. This field contains the *actual* outbound Version ID, not the compliance version. This field is used on outbound envelopes.

**Envelope Agency**

The 3-character agency code associated with the envelope version.

**Envelope Sender ID**

The 35-character outbound group envelope EDI identification for the following elements:

- GS02
- UNG0201

This field contains the *actual* group EDI identification. This field is used on outbound envelopes.

**Envelope Receiver ID**

The 35-character outbound group envelope EDI identification for the following elements:

- GS03
- UNG0203

This field contains the *actual* group EDI identification. This field is used on outbound envelopes.

## Inbound Application Databank Extract Program

**EDID405**

The Inbound Application Databank Extract program (EDID405) extracts documents from the Inbound Application Databank for processing by user applications.

The Inbound Application Databank Extract also applies outstanding online databank maintenance updates to the Inbound Application Databank. The first function the extract program always performs is to apply any outstanding online databank maintenance transactions.

### Operation

The Inbound Application Databank Extract program runs as a batch job step.

### Files Used

#### *Input*

The following files are used as input:

<b>DDname</b>	<b>File Name</b>
EDICFG	Configuration File
EDIACA	Inbound Application Databank Change Audit
EDIIDAA	Inbound Application Databank Directory
EDIIDAS	Inbound Application Databank Message Store
EDIQ095	Inbound Application Databank Transaction Queue
EDICNTL	Control Card File
ERRCTL	Error Message and Control File

#### *Output*

The following are used as output:

<b>DDname</b>	<b>File Name</b>
EDIAPP	Sequential or Gentran:Control Queue File containing extracted data
EDIACA	Inbound Application Databank Change Audit File
EDIIDAA	Inbound Application Databank Directory
EDIIDAS	Inbound Application Databank Message Store
EDIQ095	Inbound Application Databank Transaction Queue
EDIQ###	Gentran:Control Queue File, where ### identifies the queue file (Gentran:Control users only)

**Gentran:Viewpoint**

The following files are used only in Gentran:Viewpoint:

DDname	File Name
EDIVP	Profile File
EDIVSF	Staging File
EDIVT	Trace ID File (Tracking Management users only)

**Reports**

The Extract Audit Trail lists the documents that have been extracted. The Update Audit Trail reports databank maintenance updates that were applied during the run.

DDname	Description
EDIAUDT	Inbound Application Databank Extract Audit Trail
EDIAUD3	Inbound Application Databank Update Audit Trail
EDILOG	Processing Log
EDISUM	Processing Summary Report

**Control Card Options**

**OPERATIONS**

SELECT (Default operation)

**SELECTION CRITERIA**

**Defaults**

APPLICATION-DATA-ID	No default, must specify.
APPLY-UPDATES-ONLY	NO
DATABANK-CONFIGURATION	FULL
DATABANK-RUN-NUMBER	ALL
FROM-MAPPED-DATE	01/01/1900
FROM-MAPPED-TIME	00:00
FROM-OUTPUT-DATE	01/01/1900
FROM-OUTPUT-TIME	00:00
FUNCTIONAL-GROUP-ID	ALL
FUNCTIONAL-GROUP-ENV-REF-ID	ALL
INTERCHANGE-ENV-REF-ID	ALL
MAP-VALIDATION-STATUS	ALL
ONLINE-UPDATE-REPORT	YES
OUTPUT-FILE	EDIAPP



PARTNER	ALL
PARTNER-QUALIFIER	ALL
QUEUE-FILE-NUMBER	NONE
REFERENCE-TAG	ALL
STATUS	UNPROCESSED
TEST-PRODUCTION-STATUS	ALL
TO-MAPPED-DATE	12/31/2099
TO-MAPPED-TIME	23:59
TO-OUTPUT-DATE	12/31/2099
TO-OUTPUT-TIME	23:59
TRANSACTION-SET-ID	ALL
TRANSACTION-SET-ENV-REF-ID	ALL
USER-REFERENCE	ALL
USER-DUPLICATE-INDICATOR	ALL
USER	ALL
VIEWPOINT-TRACKING	NO

**VALID CRITERIA VALUES**

**Application-Data-ID**

A value must be specified in the following format:

aaaaaaaaaa

**Apply-Updates-Only**

Specifies whether to perform extract processing and create an Output file when applying updates to the databank.

NO	Apply updates, perform extract processing and create an Output file.
YES	Apply updates and do not perform extract or create an Output file.

**Databank-Configuration**

Identifies which databank files are to be used for the run. This control card should reflect how the Inbound Application Databank has been implemented.

FULL	Both the Directory and Message Store files should be used.
DIRECTORY	Directory file only.

**Databank-Run-Number**

An 8-digit number that identifies the run that last processed the document.

ALL  
nnnnnnnn (must include leading zeros)

**From-Mapped-Date**

Specifies the date when the data was last processed by the Inbound Mapper. Starts the range. Used with To-Mapped-Date.

MM/DD/YYYY

**From-Mapped-Time**

Specifies the time when the data was last processed by the Inbound Mapper. Starts the range. Used with To-Mapped-Time.

HH:MM

**From-Output-Date**

Specifies the date when the data was last output to an application. This could have been performed by either the Inbound Mapper or the Inbound Application Extract program. Starts the range. Used with To-Output-Date.

MM/DD/YYYY

**From-Output-Time**

Specifies the last time the data was output to an application. This could have been performed by either the Inbound Mapper or the Inbound Application Extract program. Starts the range. Used with To-Output-Time.

HH:MM

**Functional-Group-ID**

Specifies the 1- to 6-character functional group identification. You can specify either a Functional Group ID or ALL for all functional groups. This selection criteria is valid only if the Report-Structure control card selection criteria has a value of GROUP.

ALL  
aaaaaa

**Functional-Group-Env-Ref**

Specifies the 1- to 14-character functional group envelope reference identification.

aaaaaaaaaaaa EDIFACT  
nnnnnnnnnn X12 and TDCC – Leading zeros required  
ALL All Functional Group Envelope Reference IDs

**Interchange-Env-Ref-ID**

Specifies the 1- to 14-character interchange envelope reference identification.

aaaaaaaaaaaa EDIFACT  
nnnnnnnnnn X12 and TDCC – Leading zeros required  
ALL All Interchange Envelope Reference IDs

**Map-Validation-Status**

Used to select documents based on the validation status assigned to the document by the Inbound Mapper.

ALL  
<08, <12, 00, 04, 08, 12, >00, >04, >08

**Online-Update-Report**

Used to request an Audit Trail report for online databank maintenance updates that are applied.

NO  
YES

**Output-File**

Specifies the file the extracted documents should be written to.

EDIAPP	Application Data file.
QUEUE	Gentran:Control users only

**Partner**

A 15- to 35-character partner identification. The length of the field depends on the trading mode. For the Relationship mode, the length is 15 characters. If processing using the Partner/Qualifier mode, the length is 35 characters.

ALL	
a . . . . (35) . . . . a	(Partner/Qualifier mode)
a . . . . (15) . . . . a	(Relationship mode)

**Partner-Qualifier**

The 4-character Qualifier for the Partner ID. You can specify a Partner/Qualifier or ALL for all Partner/Qualifiers. This is valid only if the trading mode is Partner/Qualifier.

ALL
aaaa

**Queue-File-Number**

Specifies the 3-digit queue file number.

NONE	
nnn	(Only valid if OUTPUT-FILE specifies QUEUE)

**Reference-Tag**

A 10-character field assigned by Gentran:Basic that uniquely identifies a document anywhere in the system.

ALL
IAnnnnnnnn

**Status**

Specifies whether the data has been output to a user application. Can be output by either the Inbound Mapper or the Inbound Application Extract program.

PROCESSED	Has been output to an application.
UNPROCESSED	Has never been output to an application.

**Test-Production-Status**

Specifies the test/production status as assigned by Gentran:Basic that should be used for selecting documents.

ALL
PRODUCTION
TEST

**To-Mapped-Date**

Specifies the date when the data was last processed by the Inbound Mapper. Ends the range. Used with From-Mapped-Date.

MM/DD/YYYY
------------

**To-Mapped-Time**

Specifies the time when the data was last processed by the Inbound Mapper. Ends the range. Used with From-Mapped-Time.

HH:MM

**To-Output-Date**

Specifies the date when the data was last output to an application. This could have been performed by either the Inbound Mapper or the Inbound Application Extract program. Ends the range. Used with From-Output-Date.

MM/DD/YYYY

**To-Output-Time**

Specifies the last time the data was output to an application. This could have been performed by either the Inbound Mapper or the Inbound Application Extract program. Ends the range. Used with From-Output-Time.

HH:MM

**Transaction-Set-ID**

A 6-character identification used to specify the EDI transaction set.

ALL  
aaaaaa

**Transaction-Env-Ref-ID**

This selection criteria is only valid if the Report-Structure control card selection criteria has a value of TRANSACTION.

aaaaaaaaaaaaa      EDIFACT  
nnnnnnnnnn      X12 and TDCC – Leading zeros required.  
ALL

**User**

ALL or a 15-character user identification. Only valid if trading profile mode is specified as relationship.

ALL  
aaaaaaaaaaaaaaa      15-character User ID, only valid if trading mode is specified as RELATIONSHIP.

**User-Duplicate-Indicator**

Used to specify whether duplicate documents, based on the User Reference field, should be selected.

ALL      Select documents whether they are duplicates or not  
NO      Select only documents that are not duplicates  
YES      Select documents which are duplicates

**User-Reference**

Specifies the 1- to 45-character user reference value. You can enter either a user reference or ALL for all user references.

**Viewpoint-Tracking**

Indicates whether Gentran:Viewpoint information should be included in the Extract Output file to be used by a user written application tracking program. The information is passed on the fixed format Viewpoint User Tracking record.

- |     |   |
|-----|---|
| NO  | No Viewpoint information is included in the Extract Output file.  |
| YES | Viewpoint information is included in the Extract Output file.<br>Only valid if Gentran:Viewpoint Tracking Management is installed). |

## Inbound Application Databank Maintenance Program

**EDID401**

The Inbound Application Databank Maintenance program (EDID401) maintains the Inbound Application Databank Directory file, the Inbound Application Databank Message Store file, and the Inbound EDI Databank Link file. Its functions include deleting, archiving, and restoring data for the Inbound Application Databank.

### Operation

The Inbound Application Databank Maintenance program operates in the following two general modes:

- **Periodic Housekeeping**  
On a periodic basis (anywhere from daily to monthly depending on volume and retention requirements) the Housekeeping operation can be used to delete and optionally archive data based on a specified control card Retention period.
- **Ad Hoc Maintenance (Purge and Restore)**  
In addition to the regularly scheduled housekeeping runs the Inbound Application Databank Maintenance's Purge and Restore operations can be used to delete and optionally archive selected data and restore data from archives.

In either of these modes, maintenance allows the removal of Directory information and all associated Message Store data, or Message Store data only. Removing only Message Store data frees valuable disk space storage, while allowing the user to continue status inquiries.

### Processing

The Inbound Application Databank Maintenance program processes documents with the following three operations:

- Housekeeping
- Purge
- Restore

#### *Housekeeping Operation*

If the Housekeeping operation is requested the Inbound Application Databank Maintenance deletes and optionally archives data that exceeds the retention periods specified in the control cards. An option of the Housekeeping function is to archive all selected data to a sequential output file. Another option of the Housekeeping operation is to delete the data portion of documents only. This option allows the user to access status information while removing unneeded document data.

#### *Purge Operation*

If the Purge operation is requested, data is removed if it matches ALL criteria specified. An option of the Purge operation is to archive the data to a sequential output file. Another option is to delete the Message Store data portion of transactions only.

### Restore Operation

If the Restore operation is requested, data is restored from the Archive file to the Inbound Application Databank if it matches ALL selection criteria specified.

### Files Used

#### Input

The following files are used as input:

DDname	Description
EDIARCH	Databank Archive File
EDIIAA	Inbound Application Databank Directory
EDIAS	Inbound Application Databank Message Store
EDIIEL	Inbound EDI Databank Link
EDICNTL	Control Card File
ERRCTL	Error Message and Control File
EDICFG	System Control File

#### Output

The following files are used as output:

DDname	Description
EDIARCH	Databank Archive File
EDIIAA	Inbound Application Databank Directory
EDIAS	Inbound Application Databank Message Store
EDIIEL	Inbound EDI Databank Link

#### Reports

Program EDID401 produces the following reports:

DDname	Description
EDIAUDT	<p>Inbound Application Databank Maintenance Audit Trail</p> <p><b>Note:</b> This report is not always produced when the program is run. The EDIAUDT report is produced <i>only</i> if documents have been dropped from the databank or restored to the databank by meeting the selection criteria.</p>

DDname	Description
EDILOG	Processing Log
EDISUM	Processing Summary Report

**Control Card Options**

*OPERATIONS*

HOUSEKEEPING (Default operation)  
 PURGE  
 RESTORE

*SELECTION CRITERIA*

**Defaults**

For All Operations:

CONCURRENCY-ENABLED NO  
 DATABANK-CONFIGURATION FULL  
 DATABANK-MODE APPLICATION

For Housekeeping and Purge Operations:

ARCHIVE NO  
 DATABANK-DELETE-LEVEL ALL

For Housekeeping Operation:

RETENTION-DAYS-PROCESSED 030  
 RETENTION-DAYS-UNPROCESSED 030

For Purge and Restore Operations:

APPLICATION-DATA-ID ALL  
 DATABANK-RUN-NUMBER ALL  
 FROM-MAPPED-DATE 01/01/1900  
 FROM-MAPPED-TIME 00:00  
 FROM-OUTPUT-DATE 01/01/1900  
 FROM-OUTPUT-TIME 00:00  
 PARTNER ALL  
 PARTNER-QUALIFIER ALL  
 REFERENCE-TAG ALL  
 REPORTED ALL  
 TO-MAPPED-DATE 12/31/2099  
 TO-MAPPED-TIME 23:59  
 TO-OUTPUT-DATE 12/31/2099  
 TO-OUTPUT-TIME 23:59



USER ALL  
USER-REFERENCE ALL

For Purge Operation:

DATABANK-LOGICAL-DELETES NO

**Note:** Date and time selection criteria are evaluated independently. Time-based selection criteria are used to limit the data selected using the specified time range for each day within the date range. For example, selecting data with time selection criteria of 9 a.m. to 5 p.m. selects data processed between 9 and 5 on any day. Adding a date range further reduces the data selected to that which was processed between 9 and 5 on any day in the date range.

### **VALID CRITERIA VALUES**

#### **Application-Data-ID**

A 10-character value that identifies the Gentran:Basic Application file that was used to map the document.

aaaaaaaaaa

#### **Archive**

Specifies whether documents that are removed should be archived.

NO

YES

#### **Concurrency-Enabled**

Specifies whether or not concurrent processing is enabled.

NO

Disables concurrent processing. This value must be specified in a non-concurrent processing environment. In a current processing environment, specifying this value permits Databank Maintenance to be performed on concurrent Databank files when the on-line CICS system is not available.

YES

Enables concurrent processing so Databank Maintenance can be performed simultaneously with batch processing. Specifying this value requires that the on-line CICS system that owns the Databank files be available.

#### **Databank-Configuration**

Identifies which databank files are to be used for the run. This control card should reflect how the Inbound Application Databank has been implemented.

FULL

Both the Directory and Message Store files.

DIRECTORY

Directory file only.

**Databank-Delete-Level**

Controls the level of data that is removed from the databank.

ALL	Delete both Directory and Message Store data.
MESSAGE-STORE	Only delete Message Store Data.

**Databank-Logical-Deletes**

Used to specify how documents that have been marked for deletion by the Online Databank Maintenance facility should be handled.

NO	Do not delete documents that have been marked for deletion by Online Databank Maintenance.
YES	Delete only documents that have been marked for deletion.
ALL	Delete documents whether they have been marked for deletion or not.

**Databank-Run-Number**

An 8-digit number that identifies the run that last processed the document.

ALL  
nnnnnnnn (must include leading zeros)

**From-Mapped-Date**

Specifies the date when the data was first processed by the Inbound Mapper. Starts the range. Used with To-Mapped-Date.

MM/DD/YYYY

**From-Mapped-Time**

Specifies the time when the data was first processed by the Inbound Mapper. Starts the range. Used with To-Mapped-Time.

HH:MM

**From-Output-Date**

Specifies the date when the data was last output to an application. This could have been performed by either the Inbound Mapper or the Inbound Application Extract program. Starts the range. Used with To-Output-Date.

MM/DD/YYYY

**From-Output-Time**

Specifies the time when the data was last output to an application. This could have been performed by either the Inbound Mapper or the Inbound Application Extract program. Starts the range. Used with To-Output-Time.

HH:MM

**Partner**

A 15- to 35-character partner identification. The length of the field depends on the trading mode. For the Relationship mode, the length is 15 characters. If processing using the Partner/Qualifier mode, the length is 35 characters.

ALL  
a . . . . (35) . . . .a (Partner Qualifier mode)  
a . . . (15) . . .a (Relationship mode)

**Partner-Qualifier**

The 4-character Qualifier for the Partner ID. You can specify a Partner/Qualifier or ALL for all Partner/Qualifiers. This is valid only if the trading mode is Partner/Qualifier.

ALL  
aaaa

**Retention-Days-Processed**

Specifies the number of days that have elapsed since the data was output to communication by either the Inbound Editor or the Inbound Application Extract program. Used with the Housekeeping Operation.

000-999

**Retention-Days-Unprocessed**

Specifies the number of days the data has been on the databank without having been output to communications.

000-999

**Reference-Tag**

A 10-character field assigned by Gentran:Basic that uniquely identifies a document anywhere in the system.

ALL  
IAAnnnnnnn

**Reported**

This selection criteria is used to limit selection based on what has been reported previously by an Inquiry report.

ALL	Select everything regardless of previous reports.
YES	Select only what has been reported previously.
NO	Select only what has not been reported previously.

**To-Mapped-Date**

Specifies the date when the data was first processed by the Inbound Mapper. Ends the range. Used with From-Mapped-Date.

MM/DD/YYYY

**To-Mapped-Time**

Specifies the time when the data was first processed by the Inbound Mapper. Ends the range. Used with From-Mapped Time.

HH:MM

**To-Output-Date**

Specifies the date when the data was last output to an application. This could have been performed by either the Inbound Mapper or the Inbound Application Extract program. Ends the range. Used with From-Output-Date.

MM/DD/YYYY

**To-Output-Time**

Specifies the time when the data was last output to an application. This could have been performed by either the Inbound Mapper or the Inbound Application Extract program. Ends the range. Used with From-Output-Time.

HH:MM

**User**

ALL or a 15-character user identification. Only valid if the trading mode is specified as Relationship.

**User-Reference**

Specifies the 1- to 45-character user reference value. You can enter either a user reference or ALL for all user references.

**Databank-Mode**

Specifies the action the databank should perform.

APPLICATION	Application records only
STRUCTURE	Structure records only
BOTH	Application and structure records

## Inbound EDI Databank Maintenance Program

EDID301

The Inbound EDI Databank Maintenance program (EDID301) maintains the Inbound EDI Directory file and any related documents on the Inbound EDI Message Store file. Its functions include deleting and optionally archiving EDI data and restoring EDI data from archives.

**Note:** Provisions have been made to handle existing archives from Gentran:Basic Releases 5.0x, 5.5, and 5.6. The system looks at existing data and adjusts it accordingly. No data is lost in the process.

### Operation

The Inbound EDI Databank Maintenance program operates in the following two general modes:

- **Periodic Housekeeping**  
On a periodic basis (anywhere from daily to monthly depending on volume and retention requirements) the Housekeeping operation can be used to delete and optionally archive data based on a specified control card Retention period.
- **Ad Hoc Maintenance (Purge and Restore)**  
In addition to the regularly scheduled housekeeping runs the Inbound EDI Databank Maintenance's Purge and Restore operations can be used to delete and optionally archive selected data and restore data from archives.

In either of these modes, maintenance allows the removal of Directory information and all associated Message Store data, or Message Store data only. Removing only Message Store data frees valuable storage, while continuing to allow status inquiries.

### Processing

The Inbound EDI Databank Maintenance program processes documents with the following three operations:

- Housekeeping
- Purge
- Restore

#### *Housekeeping Operation*

If the Housekeeping operation is requested, the Inbound EDI Databank Maintenance deletes and optionally archives data that exceeds the retention periods specified in the control cards. An option of the Housekeeping operation is to archive all selected data to a sequential output file. Another option of the Housekeeping operation is to delete the Message Store data portion of transactions only. This option allows the user to continue to make status inquiries for the removed transactions while removing unneeded transaction data.

### *Purge Operation*

If the Purge operation is requested, data is removed if it matches ALL criteria specified. An option of the Purge operation is to archive the data to a sequential output file. Another option of the Purge operation is to delete the Message Store data portion of transactions only.

**Note:** The Databank-Run-Number selection criteria must be used to Purge from the message store. All other selection criteria are ignored when purging interchanges for which message store data was saved.

### *Restore Operation*

If the Restore operation is requested, data is restored from the Archive file to the Inbound EDI Databank if it matches ALL selection criteria specified.

**Note:** The Databank-Run-Number selection criteria must be used to Restore from the archive file to the message store file. All other selection criteria are ignored when restoring interchanges for which message store data was saved.

### **Files Used**

#### *Input*

Control cards and the Inbound EDI Databank are its principal inputs. The Archive file is used as input when the Restore operation is requested.

<b>DDname</b>	<b>File Name</b>
EDIARCH	Databank Archive File
EDIIEA	Inbound EDI Databank Directory
EDIIES	Inbound EDI Databank Message Store
EDICNTL	Control Card File
ERRCTL	Error Message and Control File
EDICFG	System Control File
EDISASC	Gentran:Basic Standards Association File (optional, used during restore)
EDISVER	Gentran:Basic Version File (optional, used during restore)

### Output

The following files are used as output:

DDname	File Name
EDIARCH	Databank Archive File
EDIIEA	Inbound EDI Databank Directory
EDIIES	Inbound EDI Databank Message Store

### Reports

Program EDID301 produces the following reports:

DDname	File Name
EDIAUDT	Inbound EDI Databank Maintenance Audit Report <b>Note:</b> This report is not always produced when the program is run. The EDIAUDT report is produced <i>only</i> if documents are dropped from the databank or restored to the databank by meeting the selection criteria.
EDILOG	Processing Log
EDISUM	Processing Summary Report

### Control Card Options

#### OPERATIONS

HOUSEKEEPING (Default operation)  
 PURGE  
 RESTORE

#### SELECTION CRITERIA

##### Defaults

For All Operations:

CONCURRENCY-ENABLED NO  
 DATABANK-CONFIGURATION FULL  
 DATABANK-DIRECTORY-LEVEL INTERCHANGE  
 REPORT-TYPE FULL

For Housekeeping and Purge Operations:

ARCHIVE NO  
 DATABANK-DELETE-LEVEL ALL

For Housekeeping Operation:

RETENTION-DAYS-LOADED 030

For Purge and Restore Operations:

DATABANK-RUN-NUMBER	ALL
FROM-LOADED-DATE	01/01/1900
FROM-LOADED-TIME	00:00
FROM-OUTPUT-DATE	01/01/1900
FROM-OUTPUT-TIME	00:00
INTERCHANGE-ENV-REF-ID	ALL
PARTNER	ALL
PARTNER-QUALIFIER	ALL
REPORTED	ALL
TO-LOADED-DATE	12/31/2099
TO-LOADED-TIME	23:59
TO-OUTPUT-DATE	12/31/2099
TO-OUTPUT-TIME	23:59
USER	ALL

For Purge Operation:

DATABANK-LOGICAL-DELETES	NO
--------------------------	----

**Note:** Date and time selection criteria are evaluated independently. Time-based selection criteria are used to limit the data selected using the specified time range for each day within the date range. For example, selecting data with time criteria of 9 a.m. to 5 p.m. equates to data processed between 9 and 5 on any day. Adding a date range further reduces the data selected to that which was processed between 9 and 5 on any day in the date range.

**VALID CRITERIA VALUES**

**Archive**

Specifies whether or not data that is removed should be archived.

- NO
- YES

**Concurrency-Enabled**

Specifies whether or not concurrent processing is enabled.

- NO Disables concurrent processing. This value must be specified in a non-concurrent processing environment. In a current processing environment, specifying this value permits Databank Maintenance to be performed on concurrent Databank files when the on-line CICS system is not available.
- YES Enables concurrent processing so Databank Maintenance can be performed simultaneously with batch processing. Specifying this value requires that the on-line CICS system that owns the Databank files be available.



**Databank-Configuration**

Identifies which databank files are to be used for the run. This control card should reflect how the Inbound EDI Databank has been implemented.

FULL	Both the Directory and Message Store files.
DIRECTORY	Directory file only.

**Databank-Delete-Level**

Controls the level of data that is removed from the databank.

ALL	Delete both Directory and Message Store data.
MESSAGE-STORE	Only delete Message Store Data.

**Databank-Directory-Level**

Identifies how data is maintained on the Directory file.

INTERCHANGE	Interchange must be specified if EDI Data is stored on the message store.
GROUP	Group should be specified if Interchange level directory records are not maintained.
TRANSACTION	Transaction should be specified if only Transaction level directory records are maintained.

**Databank-Logical-Deletes**

Used to specify how documents that have been marked for deletion by the online Databank Maintenance facility should be handled.

NO	Do not delete documents that have been marked for deletion by online Databank Maintenance.
YES	Delete only documents that have been marked for deletion.
ALL	Delete documents whether they have been marked for deletion or not.

**Databank-Run-Number**

An 8-digit number that identifies the run that last processed the data. A Databank Run Number other than ALL must be specified to Purge or Restore message store related information.

ALL  
nnnnnnnn (must include leading zeros)

**Note:** The Databank-Run-Number selection criteria must be used to Purge from the message store. All other selection criteria are ignored when purging interchanges for which message store data was saved. (All interchanges are in one record; Gentran:Basic does not distinguish each interchange individually.)

**From-Loaded-Date**

Specifies the date when the data was first processed by the Inbound Editor. Starts the range. Used with the To-Loaded-Date.

MM/DD/YYYY

**From-Loaded-Time**

Specifies the time when the data was first processed by the Inbound Editor. Starts the range. Used with the To-Loaded-Time.

HH:MM

**From-Output-Date**

Specifies the date when the data was loaded or last updated by the Inbound Editor. This could have been performed by online Databank Maintenance. Ends the range. Used with the To-Output-Date.

MM/DD/YYYY

**From-Output-Time**

Specifies the time when the data was loaded or last updated by the Inbound Editor. This could have been performed by online Databank Maintenance. Ends the range. Used with the To-Output-Time.

HH:MM

**Interchange-Env-Ref-ID**

Used to specify the interchange control number for purging and restoring individual interchanges.

ALL

aaaaaaaaaaaaaa

**Partner**

A 15- to 35-character partner identification. The length of the field depends on the trading mode. For the Relationship mode, the length is 15 characters. If processing using the Partner/Qualifier mode, the length is 35 characters.

ALL

a . . . . (35) . . . .a            (Partner/Qualifier mode)

a . . . (15) . . .a                (Relationship mode)

**Partner-Qualifier**

The 4-character Qualifier for the Partner ID. You can specify a Partner/Qualifier or ALL for all Partner/Qualifiers. This is valid only if the trading mode is Partner/Qualifier.

ALL

aaaa

**Retention-Days-Loaded**

Specifies the number of days that have elapsed since the data was loaded onto the Databank by the Inbound Editor. Used with Housekeeping:

000-999

**Reported**

This selection criteria is used to limit selection based on what has been reported previously by an Inquiry report.

ALL                    Select everything regardless of previous reports.

YES                    Select only what has been reported previously.

NO                     Select only what has not been reported previously.

**Report-Type**

Identifies the level at which the Inbound EDI Databank Maintenance Audit Trail should report.

FULL	All Groups and Transactions within Interchanges.
SUMMARY	Interchanges only.

**To-Loaded-Date**

Specifies the date when the data was first processed by the Inbound Editor. Ends the range. Ends the range. Used with the From-Loaded-Date.

MM/DD/YYYY

**From-Loaded-Time**

Specifies the time when the data was first processed by the Inbound Editor. Ends the range. Used with the From-Loaded-Time.

HH:MM

**To-Output-Date**

Specifies the date when the data was loaded or last updated by the Inbound Editor. This could have been performed by online Databank Maintenance. Ends the range. Used with the From-Output-Date.

MM/DD/YYYY

**To-Output-Time**

Specifies the time when the data was loaded or last updated by the Inbound Editor. This could have been performed by online Databank Maintenance. Ends the range. Used with the From-Output-Time.

HH:MM

**User**

ALL or a 15 character user identification. Only valid if trading mode is specified as Relationship.

## Outbound Application Databank Maintenance Program

EDID101

The Outbound Application Databank Maintenance (EDID101) program maintains the Outbound Application Directory file and the Outbound Application Message Store file. Its functions include deleting, archiving, and restoring databank data.

### Operation

The Outbound Application Databank Maintenance program operates in the following two general modes:

- **Periodic Housekeeping**  
On a periodic basis (anywhere from daily to monthly depending on volume and retention requirements) the Housekeeping operation can be used to delete and optionally archive data based on a specified control card Retention period.
- **Ad Hoc Maintenance (Purge and Restore)**  
In addition to the regularly scheduled housekeeping runs the Outbound Application Databank Maintenance's Purge and Restore operations can be used to delete and optionally archive selected data and restore data from archives.

In either of these modes, maintenance allows the removal of Directory information and all associated Message Store data, or Message Store data only. Removing only Message Store data frees valuable storage, while continuing to allow status inquiries.

### Processing

The Outbound Application Databank Maintenance program processes documents with the following three operations:

- Housekeeping
- Purge
- Restore

#### *Housekeeping Operation*

If the housekeeping operation is requested the Outbound Application Databank Maintenance deletes and optionally archives data that exceeds the retention periods specified in the control cards. An option of the Housekeeping function is to archive all selected data to a sequential output file. Another option of the Housekeeping operation is to delete the data portion of documents only. This option allows the user to access status information while removing unneeded document data.

#### *Purge Operation*

If the Purge operation is requested, data is removed if it matches ALL criteria specified. An option of the Purge operation is to archive the data to a sequential output file. Another option of the Purge operation is to delete the data portion of documents.

### Restore Operation

If the Restore operation is requested, data is restored from the Archive file to the Outbound Application Databank if it matches ALL selection criteria specified.

### Files Used

#### Input

The following files are used as input:

DDname	Description
EDIARCH	Databank Archive File
EDIOAA	Outbound Application Databank Directory
EDIOAS	Outbound Application Databank Message Store
EDICNTL	Control Card File
ERRCTL	Error Message and Control File
EDICFG	System Configuration File

#### Output

The following files are used as output:

DDname	Description
EDIARCH	Databank Archive File
EDIOAA	Outbound Application Databank Directory
EDIOAS	Outbound Application Databank Message Store

### Reports

Program EDID101 produces the following reports:

DDname	Description
EDIAUDT	Outbound Application Databank Maintenance Report <b>Note:</b> This report is not always produced when the program is run. The EDIAUDT report is produced <i>only</i> if documents have been dropped from the databank or restored to the databank by meeting the selection criteria.
EDILOG	Processing Log
EDISUM	Processing Summary Report

## Control Card Options

### OPERATIONS

HOUSEKEEPING	(Default operation)
PURGE	
RESTORE	

### SELECTION CRITERIA

#### Defaults

For All Operations:

CONCURRENCY-ENABLED	NO
DATABANK-MODE	APPLICATION
DATABANK-CONFIGURATION	FULL

For Housekeeping and Purge Operations:

ARCHIVE	NO
DATABANK-DELETE-LEVEL	ALL

For Housekeeping Operation:

RETENTION-DAYS-LOADED	030
RETENTION-DAYS-MAPPED	030

For Purge and Restore Operations:

APPLICATION-DATA-ID	ALL
DATABANK-RUN-NUMBER	ALL
FROM-LOAD-DATE	01/01/1900
FROM-LOAD-TIME	00:00
FROM-MAPPED-DATE	01/01/1900
FROM-MAPPED-TIME	00:00
PARTNER	ALL
PARTNER-QUALIFIER	ALL
REPORTED	ALL
TO-LOAD-DATE	12/31/2099
TO-LOAD-TIME	23:59
TO-MAPPED-DATE	12/31/2099
TO-MAPPED-TIME	23:59
REFERENCE-TAG	ALL
USER	ALL
USER-REFERENCE	ALL

For Purge Operations:

DATABANK-LOGICAL-DELETES	NO
--------------------------	----

**Note:** Date and time selection criteria are evaluated independently. Time-based selection criteria are used to limit the data selected using the specified time range for each day within the date range. For example, selecting data with time criteria of 9 a.m. to 5 p.m. equates to data

processed between 9 and 5 on any day. Adding a date range further reduces the data selected to that which was processed between 9 and 5 on any day in the date range.

**VALID CRITERIA VALUES**

**Application-Data-ID**

A 10-character value that identifies the Gentran:Basic Application file that was used to map the document.

aaaaaaaaaa

**Archive**

Specifies whether or not documents that are removed should be archived.

NO

YES

**Concurrency-Enabled**

Specifies whether or not concurrent processing is enabled.

NO

Disables concurrent processing. This value must be specified in a non-concurrent processing environment. In a current processing environment, specifying this value permits Databank Maintenance to be performed on concurrent Databank files when the on-line CICS system is not available.

YES

Enables concurrent processing so Databank Maintenance can be performed simultaneously with batch processing. Specifying this value requires that the on-line CICS system that owns the Databank files be available.

**Databank-Configuration**

Identifies which databank files are to be used for the run. This control card should reflect how the Outbound Application Databank has been implemented.

DIRECTORY

Directory file only.

FULL

Both the Directory and Message Store files.

**Databank-Delete-Level**

Controls the level of data that is removed from the Databank.

ALL

Both Directory and Message Store data is deleted.

MESSAGE-STORE

Only Message Store Data is deleted. If this value is used, Databank-Configuration control card must specify a value of FULL.

**Databank-Logical-Deletes**

Used to specify how documents that have been marked for deletion by the Online Databank Maintenance facility should be handled.

NO

Do not delete documents that have been marked for deletion by Online Databank Maintenance.

YES

Delete only documents that have been marked for deletion.

ALL

Delete documents whether they have been marked for deletion or not.

**Databank-Mode**

Specifies the databank on which the selected operation is performed.

Application	Application records only
Structure	Structure records only
Both	Application and structure records

**Databank-Run-Number**

An 8-digit number that identifies the run that last processed the document.

ALL  
nnnnnnnn (must include leading zeros)

**From-Load-Date**

Specifies the date when the data was first processed by the Outbound Mapper. Starts the range. Used with the To-Load-Date.

MM/DD/YYYY

**From-Load-Time**

Specifies the time when the data was first processed by the Outbound Mapper. Starts the range. Used with the To-Load-Time.

HH:MM

**From-Mapped-Date**

Specifies the date when the data was loaded or last updated by the Outbound Mapper. This could have been performed by online Databank Maintenance. Starts the range. Used with the T0-Mapped-Date.

MM/DD/YYYY

**From-Mapped-Time**

Specifies the time when the data was loaded or last updated by the Outbound Mapper. This could have been performed by online Databank Maintenance. Starts the range. Used with the T0-Mapped-Time.

HH:MM

**Partner**

A 15- to 35-character partner identification. The length of the field depends on the trading mode. For the Relationship mode, the length is 15 characters. If processing using the Partner/Qualifier mode, the length is 35 characters.

ALL  
a . . . (35) . . . a            (Partner/Qualifier mode)  
a . . . (15) . . . a            (Relationship mode)

**Partner-Qualifier**

The 4-character Qualifier for the Partner ID. You can specify a Partner/Qualifier or ALL for all Partner/Qualifiers. This is valid only if the trading mode is Partner/Qualifier.

ALL  
aaaa



**Reference-Tag**

A 10-character field assigned by Gentran:Basic that uniquely identifies a document anywhere in the system.

ALL  
OAnnnnnnnn

**Reported**

This selection criteria is used to limit selection based on what has been reported previously by an Inquiry report.

ALL            Select everything regardless of previous reports.  
YES            Select only what has been reported previously.  
NO             Select only what has not been reported previously.

**Retention-Days-Loaded**

Number of days elapsed since the data was first loaded onto the Databank. Data with a Loaded Date that is equal to or greater than the retention specified is deleted. Used with the Housekeeping Operation.

000-999

**Retention-Days-Mapped**

Number of days elapsed since the data was last Mapped. Data with a Map Date that is equal to or greater than the retention specified is deleted. Used with the Housekeeping Operation.

000-999

**To-Load-Date**

Specifies the date when the data was first processed by the Outbound Mapper. Ends the range. Used with the From-Load-Date.

MM/DD/YYYY

**To-Load-Time**

Specifies the time when the data was first processed by the Outbound Mapper. Ends the range. Used with the From-Load-Time.

HH:MM

**To-Mapped-Date**

Specifies the date when the data was loaded or last updated by the Outbound Mapper. This could have been performed by online Databank Maintenance. Ends the range. Used with the From-Mapped-Date.

MM/DD/YYYY

**To-Mapped-Time**

Specifies the time when the data was loaded or last updated by the Outbound Mapper. This could have been performed by online Databank Maintenance. Ends the range. Used with the From-Mapped-Time.

HH:MM

**User**

ALL or a 15-character user identification. Only valid if trading mode is specified as Relationship.

**User-Reference**

Specifies the 1- to 45-character user reference value. You can enter either a user reference or ALL for all user references.

---

## Outbound EDI Databank Extract Program

**EDID205**

The Outbound EDI Databank Extract program (EDID205) extracts EDI data from the outbound EDI databank. A sequential file is created for processing by your communications application. You can tailor the selection criteria to meet your communication interface needs.

You can use the Outbound EDI Databank Extract program either for nondeferred or deferred processing. For nondeferred processing, the SELECT operation extracts enveloped EDI interchanges. For deferred processing, the ENVELOPE and RE-ENVELOPE operations select and envelope EDI transactions.

The Outbound EDI Databank Extract program also applies online databank maintenance updates to the outbound EDI databank. The first function the Outbound EDI Databank Extract program always performs is to apply any outstanding online databank maintenance transactions.

### *Gentran:Control only*

Used with Gentran:Control, the Outbound EDI Databank Extract program also writes data to Queue files.

### **Operation**

The Outbound EDI Databank Extract program runs as a batch step. This program operates in Deferred Envelope mode or Nondeferred Envelope mode. These modes are described below.

### *Deferred Envelope Mode (Envelope or Re-Envelope Operation)*

In Deferred Envelope mode, the Outbound Mapper and Outbound Editor translate and compliance check EDI data that has not yet been enveloped. This data is stored in the Outbound EDI Databank Message Store file without any envelopes. You can run the Outbound Mapper and Outbound Editor as many times as necessary before you run the Outbound EDI Databank Extract program to extract and envelope the selected EDI transactions. After deferred data has been extracted and enveloped, the envelope audit information is stored on the Outbound EDI Databank Directory only. The Message Store file is never updated with this information.

If the deferred data that has already been enveloped needs to be transmitted again, the Outbound EDI Databank Extract program must be run again to extract and re-envelope the data. The envelopes are rebuilt from the information stored on the Outbound EDI Databank Directory and updated with the new dates and times for this extraction. The control numbers for interchange, group, and transaction remain the same.

The envelope and re-envelope processes compliance check the generated envelopes if parameter COMPLIANCE-CHECK-ENVELOPES is set to Y (Yes). If generated envelopes fail compliance, a message is written to the Processing Log and that document is bypassed. Once the cause of the error has been corrected, the document is extracted.

**Note:** Code checking is not performed on envelopes.

*Nondeferred Envelope Mode (Select Operation)*

In Nondeferred Envelope mode, the Outbound Mapper or Envelope Generation program has already created fully-enveloped EDI interchanges that are ready for outbound transmission. The EDI data has been stored on the Outbound EDI Databank Message Store file with envelopes. The Outbound EDI Databank Extract program is run to select the requested interchanges and extract them to the specified output file. This output now is ready for outbound transmission.

**Files Used***Input*

Control cards and the outbound EDI databank are the principal inputs to the Outbound EDI Databank Extract program. The program also inputs online update transactions from the outbound EDI databank transaction queue.

<b>DDname</b>	<b>File Name</b>
EDIOEA	Outbound EDI Databank Directory
EDIOES	Outbound EDI Databank Message Store
EDIOAL	Outbound Application Link
EDIQ091	Outbound EDI Databank Transaction Queue
EDICNTL	Control Card File
EDICFG	Configuration File
ERRCTL	Error Message and Control File
SYS090	Partner Profile (Deferred Envelope Mode only)
SYS091	Outbound Partner Control File (Deferred Envelope)
SYS001	Optimized Table File

*Output*

The principal output of the Outbound EDI Databank Extract program is a sequential extract file for processing by user applications.

The Outbound EDI Databank Change Audit file is updated to reflect any databank maintenance updates that are applied.

<b>DDname</b>	<b>File Name</b>
EDIOEA	Outbound EDI Databank Directory
EDIOES	Outbound EDI Databank Message Store
EDIOAL	Outbound Application Link

DDname	File Name
EDIOECA	Outbound EDI Change Audit
EDIQ###	Gentran:Control Queue File, where ### identifies the queue file number (Gentran:Control Users Only)
EDINRC	Network Reconciliation File
EDI800	Users Sequential Extract File
EDI5120	Users Sequential Extract File (Deferred Envelope Mode only)
EDIQ091	Outbound EDI Databank Transaction Queue
SYS091	Outbound Partner Control File (Deferred Envelope Mode only)

### *Gentran:Viewpoint*

The following files are used by Gentran:Viewpoint:

DDname	File Name
EDIVP	Profile File
EDIVSF	Staging File
EDIVT	Trace ID File (Tracking Management users only)

### *Reports*

The Outbound EDI Databank Extract Audit Trail shows those documents that were extracted. A Processing Log report, a Summary report, and an Online Update Audit Trail are also produced.

DDname	Description
EDIAUDT	Outbound EDI Databank Extract Audit Trail
EDIAUD3	Outbound EDI Databank Update Audit Trail
EDILOG	Outbound EDI Databank Processing Log
EDISUM	Outbound EDI Databank Summary Report

## Control Card Layout

### Extract Parm Record (EDICNTL)

The following is the layout for the Outbound Databank Extract program:

Function Name	Begins in Column	Example
Operation	1	1 2 3 4 5 6 7 ... 35 S E L E C T
Selection Criteria Parns	3	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 ... 35 APPLY – UPDATES – ONLY
Selected Criteria Values	35	1 2 3 4 5 ... 35 NO

## Control Card Options

### OPERATIONS

SELECT (Default operation)  
 ENVELOPE  
 RE-ENVELOPE

### SELECTION CRITERIA

#### Defaults

For all operations:

APPLY-UPDATES-ONLY NO  
 DATABANK-CONFIGURATION FULL  
 FROM-EDITED-DATE 01/01/1900  
 FROM-EDITED-TIME 00:00  
 NETWORK ALL  
 NETWORK-CONFIGURATION NO  
 NETWORK-TRACKING NO  
 ONLINE-UPDATE-REPORT YES  
 OUTPUT-FILE EDI800  
 PARTNER ALL  
 PARTNER-QUALIFIER ALL  
 QUEUE-FILE-NUMBER NONE  
 TEST-PRODUCTION-STATUS ALL  
 TO-EDITED-DATE 12/31/2099  
 TO-EDITED-TIME 23:59  
 USER ALL

VALIDATION-STATUS	COMPLIANT
VIEWPOINT-TRACKING	NO

For Select operations:

DATABANK-RUN-NUMBER	ALL
FROM-OUTPUT-DATE	01/01/1900
FROM-OUTPUT-TIME	00:00
INTERCHANGE-ENV-REF-ID	ALL
STATUS	UNPROCESSED
TO-OUTPUT-DATE	12/31/2099
TO-OUTPUT-TIME	23:59

For Re-Envelope operations:

FROM-OUTPUT-DATE	01/01/1900
FROM-OUTPUT-TIME	00:00
INTERCHANGE-ENV-REF-ID	ALL
STATUS	UNPROCESSED
TO-OUTPUT-DATE	12/31/2099
TO-OUTPUT-TIME	23:59
TCR	NO
PLUS	NO
GENERATE-BATCH-NUMBERS	YES

For Envelope operation:

FUNCTIONAL-GROUP-ID	ALL
INTERCHANGE-CHARACTER-LIMIT	9999999999
INTERCHANGE-GROUP-LIMIT	999999
INTERCHANGE-TRANSACTION-LIMIT	999999
TRANSACTION-SET-ID	ALL
USER-DUPLICATE-IND	ALL
USER-REFERENCE	ALL
UCS-NUMBERS	
6/3-NUMBERS	
5/3-NUMBERS	
5/4-NUMBERS	
COMPLIANCE-CHECK-ENVELOPES	NO
GENERATE MAX LENGTH GS TIME	NO
TCR	NO
PLUS	NO
GENERATE-BATCH-NUMBERS	YES

**VALID CRITERIA VALUES****Apply-Updates-Only**

Specifies whether or not you want only online updates applied to the databank. Valid values are:

NO	Apply updates and perform extract processing to create an Output file.
YES	Apply updates only; do <i>not</i> perform extract to create an Output file.

**Compliance-Check-Envelopes**

Specifies whether or not envelopes for deferred documents that are generated during the envelope or re-envelope process are to be compliance checked against the standards definitions . Valid values are:

NO	Do not compliance check envelopes.
YES	Compliance check envelopes.

**Databank-Configuration**

Identifies which databank files are to be used for the run. This control card should reflect how the outbound EDI databank has been implemented. Valid values are:

FULL	Both the Directory and Message Store files are to be used for the run.
DIRECTORY	The Directory file only is to be used for the run.

**Databank-Run-Number**

Identifies the run that last processed the data. Valid values are:

ALL	Selects data from all previous runs.
nnnnnnnn	Specifies the databank run number (an 8-digit number that must include leading zeros).

**From-Edited-Date**

Specifies the date on which the data was last processed by the Outbound Editor. This date is the beginning of the edited data range. This field is used with the To-Edited-Date field. Use the following format: MM/DD/YYYY.

**From-Edited-Time**

Specifies the time when the data was last processed by the Outbound Editor. This time is the beginning of the edited time range. This field is used with the To-Edited-Time field. Use the following format: HH:MM

**From-Output-Date**

Specifies the date on which the data was last output to communications, which could have been performed by either the Outbound Editor or the Outbound EDI Databank Extract program. Use the following format: MM/DD/YYYY.

**From-Output-Time**

Specifies the last time the data was output to communications, which could have been performed by either the Outbound Editor or the Outbound EDI Databank Extract program. Use the following format: HH:MM.



**Functional-Group-ID**

Specifies the 1- to 6-character functional group identification. You can specify either a Functional Group ID or ALL for all functional groups. This selection criteria is valid only if the Report-Structure control card selection criteria has a value of GROUP.

**Generate-Batch-Numbers**

Indicates whether or not the batch number are assigned by the Outbound EDI Databank Extract program for Gentran:Plus interchanges.

YES	Generate batch numbers for Gentran:Plus interchanges.
NO	Do not generate batch numbers.

**Generate-Max-Length-GS-Time**

Indicates what length to use when generating the time element in the GS Segment (GS05).

YES	The GS envelope will be generated with the GS05 time field having a length equal to the maximum allowed by the standard. ANSI versions 003020 through 003030 will have a six-digit GS time field; versions after 003030 will have an eight-digit GS time field. Versions prior to 003020 will have a four-digit GS time field.
NO	The GS envelope record will be generated with a four-digit GS time field.

**Interchange-Env-Ref-ID**

Specifies the envelope reference identification of the interchange in the following format:

aaaaaaaaaaaa	Selects EDIFACT or TRADACOMS reference IDs.
nnnnnnnn	Selects X12 or TDCC reference IDs (leading zeros required).
ALL	Selects all envelope reference IDs.

**Interchange-Character-Limit**

Limits the number of characters to be included within an EDI interchange. Use the following format: nnnnnnnnnn (10-digit number).

**Interchange-Group-Limit**

Limits the number of groups to be included within an EDI interchange. Use the following format: nnnnnn (6-digit number).

**Interchange-Transaction-Limit**

Limits the number of transactions to be included within an EDI interchange. Use the following format: nnnnnn (6-digit number).

**Network**

Specifies the 1- to 15-character network identifier for extracts. You can specify either a network identifier or ALL for all networks.

**Network-Configuration**

Indicates whether or not the Network Tracking feature is available for this run. Valid values are:

YES	Network Tracking is available and the Network Reconciliation file must be included in the job step.
NO	Network Tracking is not available.

**Network-Tracking**

Specifies whether or not network tracking should be performed against all EDI interchanges extracted. Valid values are:

YES	Perform network tracking for all EDI interchanges
NO	Perform network tracking if requested on the partner profile

**Online-Update-Report**

Used to request an Audit Trail report of online databank updates that are applied. Valid values are:

YES	Generate an Audit Trail report of online databank updates
NO	Do not generate an Audit Trail report

**Output-File**

Specifies the file to which the extracted data should be written. Valid values are:

EDI800	80-byte record output file
QUEUE	80-byte queue file records (Gentran:Control only)

The following values are available only for Deferred Envelope mode.

EDI512O	512-byte record output file
QUEUE-512	512-byte queue file records (Gentran:Control only)

**Partner**

A 15- or 35-character partner identification. The length depends on the trading mode. If the mode is Relationship, the Partner ID length is a maximum of 15; if the mode is Partner/Qualifier, the partner ID length is a maximum of 35. Valid values are:

ALL	All partners
aaaaaaaaaaaaaaa	Partner ID (only for Relationship mode)
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa	Partner ID (only for Partner/Qualifier mode)

**Partner-Qualifier**

The 4-character Qualifier for the Partner ID. You can specify a Partner/Qualifier or ALL for all Partner/Qualifiers. This is valid only if the trading mode is Partner/Qualifier.

**Plus**

Specifies whether the CONNECT \$\$ADD card is generated for Gentran:Plus data. Valid values are:

YES	Generate \$\$ADD records
NO	Do not generate \$\$ADD records

**Queue-File-Number**

Specifies the number of the Gentran:Control queue file to which the extracted data is to be written. Valid values are:

NONE	No queue files are used.
nnn	Specify the queue file number. (Valid only if OUTPUT-FILE specifies QUEUE or QUEUE-512. For Gentran:Control only.)

**Status**

Specifies whether or not the data has been output to communications. The data can be output by either the Outbound Editor or the Outbound EDI Databank Extract program. Valid values are:

ALL	Select all data, processed or unprocessed
PROCESSED	Has been output to communications
UNPROCESSED	Has not been output to communications

**Test-Production-Status**

Specifies whether or not data marked as test or production should be selected. Valid values are:

ALL	Select all test or production data
PRODUCTION	Select the production data
TEST	Select the test data

**To-Edited-Date**

Specifies the date on which the data was last processed by the Outbound Editor. This date is the ending of the edited date range. This field is used with the From-Edited-Date field. Use the following format: MM/DD/YYYY.

**To-Edited-Time**

Specifies the time when the data was last processed by the Outbound Editor. This time is the ending of the edited time range. This field is used with the From-Edited-Time field. Use the following format: HH:MM.

**To-Output-Date**

Specifies the date on which the data was last output to communications, which could have been performed by either the Outbound Editor or the Outbound EDI Databank Extract program. This date is the ending of the output date range. This field is used with the From-Output-Date field. Use the following format: MM/DD/YYYY.

**To-Output-Time**

Specifies the time when the data was last output to communications, which could have been performed by either the Outbound Editor or the Outbound EDI Databank Extract program. This time is the ending of a range of times. This field is used with the From-Output-Time field. Use the following format: HH:MM.

**Transaction-Set-ID**

Specifies the 1- to 6-character transaction set identification. You can enter either a Transaction Set ID or ALL for all transaction sets.

**User**

Specifies the 15-character user identification. You can enter either a user ID or ALL for all users. This is valid only if trading mode is Relationship.

**User-Duplicate-Ind**

Specifies whether or not transactions marked as user duplicates should be selected and enveloped.

YES	Select and envelope duplicate transactions
NO	Select and envelope non-duplicate transactions
ALL	Select and envelope all transactions

**User-Reference**

Specifies the 1- to 45-character user reference value. You can enter either a user reference or ALL for all user references.

**Validation-Status**

Specifies whether or not data marked as compliant or rejected by the Outbound Editor should be selected. Valid values are:

- COMPLIANT            Select data that meets all compliance checking in the Outbound Editor
- REJECTED            Select data that has been rejected by the Outbound Editor

**Viewpoint-Tracking**

Indicates whether or not Gentran:Viewpoint information should be included in the Extract Output file used by a user-written communications tracking program. The information is passed on the fixed-format Viewpoint User Tracking record. Valid values are:

- NO                    Gentran:Viewpoint information is *not* included in the Extract Output file.
- YES                   Gentran:Viewpoint information is included in the Extract Output file (valid only if Gentran:Viewpoint Tracking Management is installed).

**UCS-Numbers**

Specifies the 1- to 12-character standards version. You can enter either the standards version followed by agency or spaces for no standards version. This is valid only for Deferred Envelope mode, and indicates which standards versions (e.g., 003040UCS) requires special transaction control numbers to be generated. This parameter can be specified up to 100 times.

**Format:**

<b>Start Position:</b>		<u>Version</u>	<u>Agency</u>
3		35	48
	UCS-NUMBERS		
	UCS-NUMBERS	003040UCS	T

**TCR**

Indicates whether or not passthru records should be generated for deferred documents during the envelope and re-envelope process. Valid values are:

- NO                    Do not attempt to generate TCR records
- YES                   Check partner profile for TCR user records. If found, generate TCR records

**5/3-Numbers**

Specifies a 1- to 6-character Transaction Set ID. You can enter either a Transaction Set ID or spaces for no transaction set. This is valid only for Deferred Envelope mode, and indicates which transaction sets require special transaction control numbers to be generated. You can also define a version and agency to control the special transaction control numbers.

For this parameter, the transaction set control number is an 8-digit number comprised of the last five digits of the group control number and a three-digit incremental value. This parameter can be specified up to 100 times in combination with the parameters 5/4-NUMBERS and 6/3-NUMBERS.

**Format:**

		<u>Transaction</u>	<u>Version</u>	<u>Agency</u>
<b>Start Position:</b>	3	35	42	55
	5/3-NUMBERS			
	5/3-NUMBERS (all 875 transactions)	875		
	5/3-NUMBERS (only 875 for 003040UCS)	875	003040UCS	T

**5/4-Numbers**

Specifies a 1- to 6-character Transaction Set ID. You can enter either a Transaction Set ID or spaces for no transaction set. This is valid only for Deferred Envelope mode, and indicates which transaction sets require special transaction control numbers to be generated. You can also define a version and agency to control the special transaction control numbers.

For this parameter, the transaction set control number is a 9-digit number comprised of the last five digits of the group control number and a four-digit incremental value. This parameter can be specified up to 100 times in combination with the parameters 5/3-NUMBERS and 6/3-NUMBERS.

**Format:**

		<u>Transaction</u>	<u>Version</u>	<u>Agency</u>
<b>Start Position:</b>	3	35	42	55
	5/4-NUMBERS			
	5/4-NUMBERS (all 875 transactions)	875		
	5/4-NUMBERS (only 875 for 003040UCS)	875	003040UCS	T

**6/3-Numbers**

Specifies a 1- to 6-character Transaction Set ID. You can enter either a Transaction Set ID or spaces for no transaction set. This is valid only for Deferred Envelope mode, and indicates which transaction sets require special transaction control numbers to be generated. You can also define a version and agency to control the special transaction control numbers.

For this parameter, the transaction set control number is a 9-digit number comprised of the last six digits of the group control number and a three-digit incremental value. This

parameter can be specified up to 100 times in combination with the parameters 5/3-NUMBERS and 5/4-NUMBERS.

**Format:**

		<b>Transaction</b>	<b>Version</b>	<b>Agency</b>
<b>Start Position:</b>	3	35	42	55
	6/3-NUMBERS			
	6/3-NUMBERS (all 875 transactions)	875		
	6/3-NUMBERS (only 875 for 003040UCS)	875	003040UCS	T

## Outbound EDI Databank Maintenance Program

EDID201

The Outbound EDI Databank Maintenance program (EDID201) maintains the Outbound EDI Directory file and any related data on the Outbound EDI Message Store file. Its functions include deleting, optionally archiving EDI data, and restoring data from archives.

**Note:** Provisions have been made to handle existing archives from Gentran:Basic Releases 5.0x, 5.5, and 5.6. The system looks at existing data and adjusts it accordingly. No data is lost in the process.

### Operation

The Outbound EDI Databank Maintenance program operates in the following two general modes:

- **Periodic Housekeeping**  
On a periodic basis (anywhere from daily to monthly depending on volume and retention requirements) the Housekeeping operation can be used to delete and optionally archive data based on a specified control card Retention period.
- **Ad Hoc Maintenance (Purge and Restore)**  
In addition to the regularly scheduled housekeeping runs the Outbound EDI Databank Maintenance's Purge and Restore functions can be used to delete and optionally archive selected data and restore data from archives.

In either of these modes, maintenance allows the removal of Directory information and all associated Message Store, or Message Store data only. Removing only Message Store data frees valuable disk storage space, while continuing to allow status inquiries.

### Processing

The Outbound EDI Databank Maintenance program processes documents with the following four operations:

- Acknowledge
- Housekeeping
- Purge
- Restore

#### *Acknowledge Operation*

The Acknowledge operation marks selected documents on the Outbound EDI Databank file as manually acknowledged.

### *Housekeeping Operation*

If the Housekeeping operation is requested the Outbound EDI Databank Maintenance program deletes and optionally archives data that exceeds the retention periods specified in the control cards.

One option of the Housekeeping function is to archive all selected data to a sequential output file. Another option of the Housekeeping operation is to delete the Message Store data portion of documents only. This option allows the user to continue to access status information while removing unneeded document data.

### *Purge Operation*

If the Purge operation is requested, data is removed if it matches ALL criteria specified. An option of the Purge operation is to archive the data to a sequential output file. Another option of the Purge operation is to delete the Message Store data portion of documents only. This option allows the user to continue to access status information while removing unneeded document data.

### *Restore Operation*

If the Restore operation is requested, data is restored from the Archive file to the Outbound EDI Databank if it matches ALL selection criteria specified.

## **Files Used**

### *Input*

Control cards and the Outbound EDI Databank are its principal inputs. The Archive file is used as an input when the Restore operation is requested.

The Outbound Application Databank Link file is updated concurrently with the Outbound EDI Databank.

<b>DDname</b>	<b>File Name</b>
EDIARCH	Databank Archive File
EDIOEA	Outbound EDI Databank Directory
EDIOES	Outbound EDI Databank Message Store
EDIOAL	Outbound Application Databank Link File
EDICNTL	Control Card File
ERRCTL	Error Message and Control File
EDICFG	System Configuration File
EDISASC	Gentran:Basic Standard Association File (optional, used during restore)
EDISVER	Gentran:Basic Version File (optional, used during restore)



*Output*

The following files are used as output:

<b>DDname</b>	<b>File Name</b>
EDIARCH	Users Archive File
EDIOEA	Outbound EDI Databank Directory
EDIOES	Outbound EDI Databank Message Store
EDIOAL	Outbound Application Databank Link File

*Reports*

Program EDID201 produces the following reports:

<b>DDname</b>	<b>File Name</b>
EDIAUDT	Outbound EDI Databank Maintenance Audit Report <b>Note:</b> This report is not always produced when the program is run. The EDIAUDT report is produced <i>only</i> if documents are processed by meeting the selection criteria.
EDILOG	Processing Log
EDISUM	Processing Summary Report

**Control Card Options**

*OPERATIONS*

HOUSEKEEPING (Default operation)  
 PURGE  
 RESTORE  
 ACKNOWLEDGE

*SELECTION CRITERIA*

**Defaults**

For All Operations:

CONCURRENCY-ENABLED NO  
 DATABANK-CONFIGURATION FULL  
 DATABANK-DIRECTORY-LEVEL INTERCHANGE  
 REPORT-TYPE FULL

For Housekeeping and Purge Operations:

ARCHIVE	NO
DATABANK-DELETE-LEVEL	ALL

For Housekeeping Operation:

CLEAN	YES
RETENTION-DAYS-PROCESSED	030
RETENTION-DAYS-UNPROCESSED	030

For Acknowledge, Purge, and Restore Operations:

DATABANK-RUN-NUMBER	ALL
FROM-EDITED-DATE	01/01/1900
FROM-EDITED-TIME	00:00
FROM-OUTPUT-DATE	01/01/1900
FROM-OUTPUT-TIME	00:00
INTERCHANGE-ENV-REF-ID	ALL
NETWORK	ALL
NETWORK-STATUS	ALL
PARTNER	ALL
PARTNER-QUALIFIER	ALL
REPORTED	ALL
STATUS	PROCESSED
TO-EDITED-DATE	12/31/2099
TO-EDITED-TIME	23:59
TO-OUTPUT-DATE	12/31/2099
TO-OUTPUT-TIME	23:59
USER	ALL

For Purge Operation:

DATABANK-LOGICAL-DELETES	NO
--------------------------	----

For Acknowledge Operation:

ACKNOWLEDGE-ENV-REF-ID	ALL
------------------------	-----

**Note:** Date and time selection criteria are evaluated independently. Time-based selection criteria are used to limit the data selected using the specified time range for each day within the date range. For example, selecting data with time selection criteria of 9 a.m. to 5 p.m. selects data processed between 9 and 5 on any day. Adding a date range further reduces the data selected to that which was processed between 9 and 5 on any day in the date range.

**VALID CRITERIA VALUES**

**Acknowledge-Env-Ref-ID**

A 14-character envelope reference identification for the databank directory level specified.  
Valid values are:

ALL	All envelope reference IDs
aaaaaaaaaaaa	An envelope ID (14 characters)

**Archive**

Specifies whether or not data that is removed should be archived.

NO	No, do not archive.
YES	Yes, archive.

**Clean**

Specifies whether obsolete deferred enveloping directory records should be deleted.

YES	Yes, perform deferred cleanup
NO	No, do not perform deferred cleanup
ONLY	Only perform deferred cleanup

**Concurrency-Enabled**

Specifies whether or not concurrent processing is enabled.

NO	Disables concurrent processing. This value must be specified in a non-concurrent processing environment. In a current processing environment, specifying this value permits Databank Maintenance to be performed on concurrent Databank files when the on-line CICS system is not available.
YES	Enables concurrent processing so Databank Maintenance can be performed simultaneously with batch processing. Specifying this value requires that the on-line CICS system that owns the Databank files be available.

**Databank-Configuration**

Identifies which databank files are to be used for the run. This control card should reflect how the Outbound EDI Databank has been implemented.

FULL	Both the Directory and Message Store files.
DIRECTORY	Directory file only.

**Databank-Delete-Level**

Controls the level of data that is removed from the Databank.

ALL	Both Directory and Message Store data are deleted.
MESSAGE-STORE	Only Message Store Data are deleted.

**Databank-Directory-Level**

Identifies how data is maintained on the Directory file.

INTERCHANGE	Interchanges which match all specified selection criteria are processed, along with all associated groups and transactions. Interchange must be specified when processing documents with related message store data.
GROUP	Group should be specified if Interchange level directory records are not maintained. Group Directory entries which were saved with associated Message Store Data are not selected when the Databank-Directory-Level is set to Group.
TRANSACTION	Transaction should be specified if only Transaction level directory records are maintained. Transaction Directory entries which were saved with associated Message Store Data are not selected when the Databank-Directory-Level is set to Transaction.

**Databank-Logical-Deletes**

Used to specify how documents that have been marked for deletion by the Online Databank Maintenance facility should be handled.

NO	Do not delete documents that have been marked for deletion by Online Databank Maintenance.
YES	Delete only documents that have been marked for deletion.
ALL	Delete documents whether they have been marked for deletion.

**Databank-Run-Number**

An 8-digit number that identifies the run that last processed the data.

ALL  
nnnnnnnn (must include leading zeros)

**From-Edited-Date**

Specifies the date when the data was first processed by the Outbound Editor. Starts the range. Used with the To-Edited-Date.

MM/DD/YYYY

**From-Edited-Time**

Specifies the time when the data was first processed by the Outbound Editor. Starts the range. Used with the To-Edited-Time.

HH:MM

**From-Output-Date**

Specifies the date when the data was last output to communications. This could have been performed by either the Outbound Editor or the Outbound EDI Extract program. Starts the range. Used with the To-Output-Date.

MM/DD/YYYY

**From-Output-Time**

Specifies the time the data was output to communications. This could have been performed by either the Outbound Editor or the Outbound EDI Extract program. Starts the range. Used with the To-Output-Time.

HH:MM

**Interchange-Env-Ref-ID**

Used to specify the interchange control number for purging and restoring individual interchanges.

ALL  
aaaaaaaaaaaaa

**Network**

Specifies the Network identifier for purge and restore operations.

ALL  
aaaaaaaaaaaaa

**Network-Status**

Used to specify the network status of the data. Valid for interchange level data only.

ALL  
ERROR-ENCOUNTERED  
MANUALLY-COMPLETED  
NOT-REQUIRED  
PARTNER-PICKED-UP  
RECEIVED-BY-NETWORK  
SORTED-BY-NETWORK  
WAITING

**Partner**

A 15- to 35-character partner identification. The length of the field depends on the trading mode. For the Relationship mode, the length is 15 characters. If processing using the Partner/Qualifier mode, the length is 35 characters.

ALL  
a . . . . (35) . . . . a            (Partner/Qualifier mode)  
a . . . (15) . . . a                (Relationship mode)

**Partner-Qualifier**

The 4-character Qualifier for the Partner ID. You can specify a Partner/Qualifier or ALL for all Partner/Qualifiers. This is valid only if the trading mode is Partner/Qualifier.

ALL  
aaaa

**Retention-Days-Processed**

Specifies the number of days that have elapsed since the data was output to communication by either the Outbound Editor or the Outbound EDI Extract program. Used with the Housekeeping Operation.

000-999

**Retention-Days-Unprocessed**

Specifies the number of days the data has been on the Databank without having been output to communications.

000-999

**Reported**

This selection criteria is used to limit selection based on what has been reported previously by an Inquiry report.

ALL	Select everything regardless of previous reports.
YES	Select only what has been reported previously.
NO	Select only what has not been reported previously.

**Report-Type**

Identifies the level at which the Outbound EDI Databank Maintenance Audit Trail should report.

FULL	All Groups and Transaction within Interchanges.
SUMMARY	Interchanges only.

**Status**

Specifies whether the data has been output to communications. Can be output by either the Outbound Editor or the Outbound EDI Extract program.

PROCESSED	Has been output to communication.
UNPROCESSED	Has never been output to communications or has been reset to unprocessed by the Online Databank Maintenance Facility.

**To-Edited-Date**

Specifies the date when the data was first processed by the Outbound Editor. Ends the range. Used with the From-Edited-Date.

MM/DD/YYYY

**To-Edited-Time**

Specifies the time when the data was first processed by the Outbound Editor. Ends the range. Used with the From-Edited-Time.

HH:MM

**To-Output-Date**

Specifies the date when the data was last output to communications. This could have been performed by either the Outbound Editor or the Outbound EDI Extract program. Ends the range. Used with the From-Output-Date.

MM/DD/YYYY

**To-Output-Time**

Specifies the time when the data was last output to communications. This could have been performed by either the Outbound Editor or the Outbound EDI Extract program. Ends the range. Used with the From-Output-Time.

HH:MM

**User**

ALL or a 15-character user identification. Only valid if trading mode is specified as Relationship.

## Transaction Queue Initialize Program

**EDID520**

The Transaction Queue Initialize program (EDID520) initializes Transaction Queues. Transaction Queues collect online updates performed by users by means of the Databank Maintenance facility.

### Operation

Databank Transaction Queue Initialize runs as a batch job.

**Note:** This program is only used if online updates are going to be made to one or more databanks. A transaction queue must be initialized for each databank that is being used.

### Files Used

#### Input

The following files are used as input:

DDname	Description
EDICNTL	Control Card File
ERRCTL	Error Message and Control File
EDICFG	System Configuration File

#### Output

The following file is used as output:

DDname	Description
EDIQF	Transaction Queue File <b>Note:</b> The Transaction-Queue control card identifies that queue is to be initialized.

#### Reports

Program EDID520 produces the following reports:

DDname	Description
EDILOG	Processing Log
EDISUM	Processing Summary

## Control Card Options

Transaction Queue Initialize allows the following operation and selection criteria to be specified by control cards:

### *OPERATION*

FORMAT (Default operation)

### *SELECTION CRITERIA*

#### **Defaults**

TRANSACTION-QUEUE	No Default value, must specify
COUNT	000000030

### *VALID CRITERIA VALUES*

#### **Transaction-Queue**

INBOUND-EDI  
INBOUND-APPLICATION  
OUTBOUND-APPLICATION  
OUTBOUND-EDI

#### **Count**

Specifies the number of records to initialize in the queue file. The parameter is defined as:

columns 3 – 7 value COUNT  
columns 35 – 43 byte numeric field, zero filled



## EDI Databank Orphan Record Cleanup Program

**EDIG300**

This utility program reads the inbound and outbound EDI databank directory files and deletes “orphan” G, D, and T databank records. Orphan databank records are caused when a housekeeping job (EDID201 or EDID301) is cancelled. These records were extremely difficult to delete when using Interchange Level databanking and the Message Store feature. Program EIDG300 is only available in a batch environment.

This program should run immediately after a housekeeping function has completed in its entirety. The initial housekeeping cleans off all records older than a specified number of days. After housekeeping is performed, you can run the Orphan Cleanup program to delete all records older than a specified number of days (this should be the same number of days used in the housekeeping job).

**Note:** The retention days parameters used by housekeeping must be identical to the retention days parameters used by the Orphan Record Cleanup program. EDIG300 sequentially reads the EDIOEA and EDIIEA VSAM files to clean up all records older than the specified number of days. We strongly recommend that you back up the files prior to running this job.

### Files Used

#### *Input*

The following files are used as input:

DDname	Description
ERRCTL	Error Message and Control File
EDICFG	System Configuration File
EDIIEA	Inbound EDI Databank Directory File (Valid when Report Only parameter is on.)
EDIOEA	Outbound EDI Databank Directory File (Valid when Report Only parameter is on.)
EDIOAL	Outbound EDI Databank Link File (Valid when the Report Only parameter is on.)
EDICNTL	Control Card File

#### *Output*

The following files are used as output:

DDname	Description
EDIIEA	Inbound EDI Databank Directory File

DDname	Description
EDIOEA	Outbound EDI Databank Directory File
EDIOAL	Outbound EDI Databank Link File
EDICNTL	Control Card File

### Reports

Program EIDG300 produces the following reports:

DDname	Description
EDIAUDT	EDI Databank Maintenance Report
EDISUM	Processing Summary Report
SYS005	Error List Report

### Control Card Options

EDI Databank Orphan Record Cleanup allows the following criteria to be specified by control cards:

#### VALID CRITERIA VALUES

##### Trading Profile-Mode

Determines how the Partner, Partner/Qualifier and User control cards should be used in the search. The mode specified here should reflect how trading partner profiles have been configured. The default value is Partner/Qualifier. Valid values are:

PARTNER-QUALIFIER	Partner/Partner Qualifier
RELATIONSHIP	User/Partner

##### Databank

You can run this program against both EDI databanks or you can select either option. The default value is BOTH. Valid values are:

INBOUND-EDI  
 OUTBOUND-EDI  
 BOTH

##### Report-Only

You have the option to run this program against the files and only report on the records that you want to delete. If the parameter is set to **YES**, the records you want to delete are listed on a report, but the records are not be deleted from the file. If the parameter is set to **NO**, the records are listed on a report and also deleted from the file. The default value is **YES**. Valid values are:

YES  
 NO

**Retention-Days-Loaded**

This parameter is used for the inbound EDI databank. This parameter specifies the number of days that have elapsed since the data was loaded onto the databank by the Inbound Editor. You **MUST** use the same number of days as the Inbound EDI Housekeeping job (EDID301). Valid values are:

000-999

**Retention-Days-Processed**

This parameter is used for the outbound EDI databank. This parameter specifies the number of days that have elapsed since the data was output to communications by either the Outbound Editor or the Outbound EDI Extract program. You **MUST** use the same number of days as the Outbound EDI Housekeeping job (EDID201).

Valid values are:

000-999

**Retention-Days-Unprocessed**

This parameter is used for the outbound EDI databank. This parameter specifies the number of days the data has been on the databank without having been output to communications. You **MUST** use the same number of days as the Outbound EDI Housekeeping job (EDID201). Valid values are:

000-999

## Program Report Names and Descriptions

The following table describes each program report and its associated program:

Report Number	Name	Report Description	Program Name
EDI094	EDIAUDT	Outbound Application Databank Maintenance Audit Report	EDID101
EDI095	EDILOG	Outbound Application Processing Log	EDID101
EDI096	EDISUM	Outbound Application Processing Summary Report	EDID101
EDI097	EDIAUDT	Inbound Application Databank Maintenance Audit Report	EDID401
EDI098	EDILOG	Inbound Application Processing Log	EDID401
EDI099	EDIAUDT	Outbound EDI Databank Maintenance Audit Report	EDID201
EDI100	EDILOG	Outbound EDI Processing Log	EDID201
EDI101	EDISUM	Outbound EDI Processing Summary Report	EDID201
EDI102	EDIAUDT	Inbound EDI Databank Maintenance Audit Report	EDID301
EDI103	EDILOG	Inbound EDI Processing Log	EDID301
EDI104	EDISUM	Inbound EDI Processing Summary Report	EDID301
EDI106	EDIAUD3	Inbound EDI Databank Update Audit Trail	EDID405
EDI107	EDIAUD3	Outbound EDI Databank Update Audit Trail	EDID205
EDI112	EDIRPT	Databank Change Audit Inquiry Report	EDID500
EDI113	EDILOG	Databank Change Audit Processing Log	EDID500
EDI114	EDISUM	Databank Change Audit Summary Report	EDID500
EDI115	EDILOG	Acknowledgment Reconciliation/ Monitor Processing Log	EDID510
EDI116	EDIRPT1	Acknowledgment Reconciliation/ Monitor Reconciliation Report	EDID510

Report Number	Name	Report Description	Program Name
EDI117	EDIPRT2	Acknowledgment Reconciliation/ Monitor Expected acknowledgment Report	EDID510
EDI118	EDISUM	Acknowledgment Reconciliation/ Monitor Summary Report	EDID510
EDI119	EDILOG	Transaction Queue Initialization Processing Log	EDID520
EDI120	EDISUM	Transaction Queue Initialization Summary Log	EDID520
EDI121	EDIAUDT	Inbound EDI Databank Extract Audit Trail	EDID405
EDI122	EDILOG	Inbound EDI Processing Log	EDID405
EDI123	EDISUM	Inbound EDI Summary Report	EDID405
EDI124	EDILOG	Commerce:Network Response Processing Log	EDID800
EDI125	EDISUM	Commerce:Network Response Processing Summary	EDID800
EDI126	EDILOG	GEIS Response Processing Log	EDID805
EDI127	EDISUM	GEIS Response Processing Summary	EDID805
EDI128	EDILOG	IBM Information Network Response Processing Log	EDID810
EDI129	EDISUM	IBM Information Network Response Processing Summary	EDID810
EDI130	EDIAUDT	Network Reconciliation Audit	EDID850
EDI131	EDILOG	Network Reconciliation Processing Log	EDID850
EDI132	EDISUM	Network Reconciliation Summary Log	EDID850
EDI133	EDILOG	Network Reconciliation Maintenance Log	EDID860
EDI134	EDISUM	Network Reconciliation Maintenance Summary Report	EDID860
EDI139	EDIAUDT	Outbound EDI Databank Extract Audit Trail	EDID205
EDI140	EDILOG	Outbound EDI Processing Log	EDID205

Report Number	Name	Report Description	Program Name
EDI141	EDISUM	Outbound EDI Summary Report	EDID205
EDI142	EDILOG	EDI Databank Inquiry Processing Log	EDID550
EDI143	EDIRPT	EDI Databank Inquiry Summary Inquiry Report/Detail Inquiry	EDID550
EDI144	EDISUM	EDI Databank Inquiry Summary Report	EDID550
EDI145	EDISUM	Inbound Application Processing Summary Report	EDID401
EDI146	EDIAUDT	Change Audit Maintenance Report	EDID502
EDI147	EDILOG	Change Audit Maintenance Processing Log	EDID502
EDI148	EDISUM	Change Audit Maintenance Summary Log	EDID502
EDI149	EDILOG	Application Databank Inquiry Processing Log	EDID551
EDI150	EDIRPT	Application Databank Inquiry Report	EDID551
EDI151	EDISUM	Application Databank Inquiry Summary Report	EDID551
EDI152	EDILOG	EDI Archive Inquiry Processing Log	EDID555
EDI153	EDIRPT	EDI Archive Inquiry Report	EDID555
EDI154	EDISUM	EDI Archive inquiry Summary Report	EDID555

# Network Utility Programs

## Overview

This chapter contains the following topics related to the network utility programs:

<b>Topic</b>	<b>Page</b>
Commerce:Network Response Processor.....EDID800 .....	6-2
Commerce:Network Response Processor (International).....EDID800I .....	6-4
GEIS Response Processor .....	6-6
EDID805 .....	6-6
IBM IN Response Processor.....EDID810 .....	6-7
Network Reconciliation.....EDID850 .....	6-9
Network Reconciliation Maintenance .....	6-11
EDID860 .....	6-11
Universal Response Processor.....EDID825 .....	6-14

**Commerce:Network Response Processor****EDID800**

The Commerce:Network Response Processor (EDID800) program extracts network status information for outbound interchanges from selected machine-readable Commerce:Network network reports.

**Note:** The machine-readable reports processed from Commerce:Network Network Services are at **Revision Level 07/91**. The input record length is 80.

**Operation**

The Commerce:Network Response Processor program runs as a batch job step.

You can execute the Commerce:Network response processor after receiving either a Transmission Confirmation report (020), an On-Demand Mailslot Status report (060), or a Daily Processing Summary report (070) in machine-readable format. The network status information is formatted into a temporary sequential file (EDIFNF) that should then be processed by the Network Reconciliation program.

**Files Used***Input*

The Network Response file from the Commerce:Network network reports is the principal input for this program.

<b>DDname</b>	<b>Description</b>
EDINR	Network Response File
ERRCTL	Error Message and Control File
EDICFG	System Configuration File

*Output*

The Formatted Network file is the principal output for this program.

<b>DDname</b>	<b>Description</b>
EDIFNF	Formatted Network File



## Reports

The Commerce:Network Response Processor produces the following reports:

DDname	Description
EDILOG	Processing Log Report
EDISUM	Processing Summary Report

## Control Card Options

None

**Commerce:Network Response Processor (International)****EDID800I**

The Commerce:Network Response Processor (EDID800I) program (international) extracts network status information for outbound interchanges from selected machine-readable Commerce:Network network reports.

**Note:** The machine-readable reports processed from Commerce:Network Network Services are at **Revision Level 05/96**. The input record length is 133.

**Operation**

The Commerce:Network Response Processor program runs as a batch job step.

You can execute the Commerce:Network Response Processor after receiving either a Transmission Confirmation report (020), an On-Demand Mailslot Status report (060), or a Daily Processing Summary report (070), in machine-readable format. The network status information is formatted into a temporary sequential file (EDIFNF) that should then be processed by the Network Reconciliation program.

**Files Used***Input*

The Network Response file from the Commerce:Network network reports is the principal input for this program.

DDname	Description
EDINR	Network Response File (for international use only)
ERRCTL	Error Message and Control File
EDICFG	System Configuration File

*Output*

The Formatted Network file is the principal output for this program.

DDname	Description
EDIFNF	Formatted Network File

## Reports

The Commerce:Network Response Processor generates the following reports:

DDname	Description
EDILOG	Processing Log Report
EDISUM	Processing Summary Report

## Control Card Options

None

## GEIS Response Processor

**EDID805**

The GEIS Response Processor (EDID805) program is used to extract network status information for outbound interchanges from the electronic format reports.

**Note:** The machine-readable reports processed from GEIS Network Services are **Electronic Format Reports version 2**. The input record length is 80.

### Operation

The GEIS Response Processor program runs as a batch job step.

You can execute the GEIS Response Processor after receiving either a Sender Status report, or an Unretrieved Documents report. The network status information is formatted into a temporary sequential file (EDIFNF) that should then be processed by the Network Reconciliation program.

### Files Used

#### *Input*

The Network Response file from GEIS network services is the principal input for this program.

DDname	Description
EDINR	Network Response File
ERRCTL	Error Message and Control File (for international use only)
EDICFG	System Configuration File

#### *Output*

The Formatted Network file is the principal output for this program.

DDname	Description
EDIFNF	Formatted Network File

#### *Reports*

The GEIS response processor produces the following reports:

DDname	Description
EDILOG	Processing Log Report
EDISUM	Processing Summary Report

### Control Card Options

None

**IBM IN Response Processor****EDID810**

The IBM IN Response Processor (EDID810) program is used to extract network status information for outbound interchanges from selected IBM IN audit reports.

**Note:** The machine-readable reports processed from IBM IN are **IBM Audit Reports Information Exchange version 3.0**. The input record length is 254.

**Operation**

The IBM IN Response Processor program runs as a batch job step.

You can execute the IBM IN Response Processor after receiving an audit report. The audit status information is formatted into a temporary sequential file (EDIFNF) that should then be processed by the Network Reconciliation program.

**Files Used***Input*

The Network Response file from IBM IN audit reports is the principal input for this program.

DDname	Description
EDINR	Network Response File
ERRCTL	Error Message and Control File
EDICNTL	Control Card File (for international use only)
EDICFG	System Configuration File

*Output*

The Formatted Network file is the principal output for this program.

DDname	Description
EDIFNF	Formatted Network Response File

*Reports*

The IBM IN Response Processor produces the following reports:

DDname	Description
EDILOG	Processing Log Report
EDISUM	Processing Summary Report

**Control Card Options (For International Customers Only)**

For international conventions, IBM sometimes attaches banner messages to their IBM IN audit reports. You must activate certain parameters for the EDID810 program to bypass these banner messages. The SELECT parameter, for example, instructs EDID810 to process only those report records that contain the specified IBM account name in the first eight positions of the record. All other records in the audit report are bypassed.

**Note:** If the IBM IN audit reports do not contain these banner messages, you do not need to active the parameters.

**OPERATIONS**

SELECT

**SELECTION CRITERIA****Defaults**

ACCOUNT-ID	No, default, must specify
DISPLAYS	NO

**VALID CRITERIA VALUES****Account-ID**

An alphanumeric value up to eight characters in length that indicates the IBM network account name that appears in the first eight positions of the audit report.

**Displays**

Indicates whether or not to display the bypassed report records.

NO	Do not display the bypassed records.
YES	Display all bypassed records.

## Network Reconciliation

## EDID850

The Network Reconciliation program (EDID850) reconciles network status information received from your network(s), back to interchanges that have been sent.

### Operation

The Network Reconciliation program runs as a batch job step.

You should execute the Network Reconciliation program after running one of the network response processors. The Network Reconciliation program posts the formatted network information created by the network response processor back to the EDI interchanges which were sent.

Multiple formatted network files from the network response processors can be processed by the Network Reconciliation program in a single run.

### Files Used

#### *Input*

The Formatted Network file and the Network Reconciliation file are the principal input for this program.

DDname	Description
EDIFNF	Formatted Network Response File
EDINRC	Network Reconciliation File
ERRCTL	Error Message and Control File
EDICFG	System Configuration File

#### *Output*

The Outbound EDI Databank Directory is updated by this program.

DDname	Description
EDIOEA	Outbound EDI Databank Directory

*Reports*

The Network Reconciliation program produces the following reports:

<b>DDname</b>	<b>Description</b>
EDIAUDT	Network Reconciliation Audit Trail
EDILOG	Processing Log
EDISUM	Summary Report

*Gentran:Viewpoint*

The following files are only used in Gentran:Viewpoint.

<b>DDname</b>	<b>Description</b>
EDIVT	Trace ID File
EDIVP	Profile File
EDIVSF	Staging Files

**Control Card Options***OPERATIONS*

PROCESS (Default operation)

*SELECTION CRITERIA***Defaults**

AUDIT-LEVEL FULL

*VALID CRITERIA VALUES***Audit-Level**

EXCEPTION	Only report exceptions.
FULL	Report all activity.
NONE	No Audit level report produced.



## Network Reconciliation Maintenance

EDID860

The Network Reconciliation Maintenance (EDID860) program maintains the Network Reconciliation file.

### Operation

The Network Reconciliation Maintenance program processes Expected Network Status records on the Network Reconciliation file with the following two operations:

- Housekeeping
- Purge

#### *Housekeeping Operation*

Housekeeping is run periodically to delete Expected Network Status records based on the retention period specified.

If the housekeeping operation is requested, the Network Reconciliation File Maintenance deletes expected status records that exceed the retention period specified in the control cards.

#### *Purge Operation*

On an exception basis, the Purge operation is used to remove specific expected status records based on selection criteria specified.

If the Purge operation is requested, an expected status record is deleted if it matches ALL criteria specified.

### Files Used

#### *Input*

The Network Reconciliation file is the principal input for this program.

DDname	Description
EDINRC	Network Reconciliation File
ERRCTL	Error Message and Control File
EDICFG	System Configuration File

#### *Output*

The Network Reconciliation file is output for this program.

DDname	Description
EDINRC	Network Reconciliation File

## Reports

The Network Reconciliation Maintenance program produces the following reports:

DDname	Description
EDILOG	Processing Log
EDISUM	Summary Report

## Control Card Options

### OPERATIONS

HOUSEKEEPING (Default operation)  
PURGE

### SELECTION CRITERIA

#### Defaults

For Housekeeping operation:

RETENTION-DAYS-RECONCILED      030  
RETENTION-DAYS-UNRECONCILED    030

For Purge operation:

DATABANK-RUN-NUMBER            ALL  
DUPLICATE-INDICATOR            YES  
RECEIVER-EDI-ID                ALL  
RECONCILIATION-STATUS        ALL  
SENDER-EDI-ID                 ALL  
INTERCHANGE-ENV-REF-ID        ALL

### VALID CRITERIA VALUES

#### Databank-Run-Number

An 8-digit number that identifies the run that last processed the document.

ALL  
nnnnnnnn (must include leading zeros)

#### Duplicate-Indicator

Used to identify if the Expected Network Status record is for a duplicate on the Outbound EDI Databank.

ALL                    Select regardless of Duplicate indicator.  
NO                     Select only Expected Network Status records that are not  
                              duplicates.  
YES                    Select just Expected Network Status records that are  
                              duplicates.

**Interchange-Env-Ref-ID**

aaaaaaaaaaaaa	EDIFACT
nnnnnnnnn	X12 and TDCC – Leading zeros required.
ALL	

**Receiver-EDI-ID**

The Receiver ID from the transmission level EDI envelope.

**Reconciliation Status**

ALL	Includes all Reconciliation status values.
RECONCILED	Has been reconciled to the Outbound EDI data on the Databank.
EXPECTED	Awaiting status from Network.
UNMATCHED	No longer able to find Outbound EDI data on Databank for status received from Network.
UNRECONCILED	Includes both EXPECTED and UNMATCHED status values.

**Sender-EDI-ID**

The Sender ID from the transmission level EDI envelope.

## Universal Response Processor

EDID825

### Purpose

The Universal Response Processor program (EDID825) extracts network status information for outbound EDI interchanges from a machine-readable Network Response file. The Network Response file either can be received from a network, or it can be created by reformatting an existing network report. The file layout in this section describes the Network Response file.

### Operation

The Universal Response Processor program runs as a batch job step.

You can execute the Universal Response Processor after the system receives a Network Response file. The network status information is formatted into a temporary sequential file (EDIFNF), which is then processed by the Network Reconciliation program.

### Files Used

#### Input

The Network Response file is the principal input for this program.

DDname	Description
EDINR	Network Response File
ERRCTL	Error Message and Control File
EDICFG	System Configuration File

#### Output

The Formatted Network file is the principal output for this program.

DDname	Description
EDIFNF	Formatted Network File

#### Reports

The Network Response Processor produces the following reports:

DDname	Description
EDILOG	Processing Log Report
EDISUM	Processing Summary Report

### Control Card Options

None

*Universal Format Network Response File – File Layout*

The Universal Format Network Response file is a standard sequential file with a record length of 350 (DSORG=PS, LRECL=350).

A copybook is provided (NRURC) with the following record layout.

Position	Length	Format	Field Name
0001 – 0001	1	AN	Record Format Indicator
0002 – 0016	15	AN	Network Name
0017 – 0051	35	AN	EDI Sender ID
0052 – 0086	35	AN	EDI Receiver ID
0087 – 0100	14	AN	Interchange Envelope Reference ID
0101 – 0106	6	N	Received Date
0107 – 0112	6	AN	Received Time
0113 – 0115	3	AN	Received Time Qualifier
0116 – 0121	6	N	Received Error Count
0122 – 0127	6	N	Processed Date
0128 – 0133	6	N	Processed Time
0134 – 0136	3	AN	Processed Time Qualifier
0137 – 0142	6	N	Processed Error Count
0143 – 0148	6	N	Picked Up Date
0149 – 0154	6	N	Picked Up Time
0155 – 0157	3	AN	Picked Up Time Qualifier
0158 – 0163	6	N	Picked Up Error Count
0164 – 0203	40	AN	Received Error Detail
0204 – 0243	40	AN	Processed Error Detail
0244 – 0283	40	AN	Picked Up Error Detail
0284 – 0350	67	AN	Reserved For Future Use

*Field Descriptions***EDI Sender ID**

The EDI Sender ID on the interchange envelope.

**EDI Receiver ID**

The EDI Receiver ID on the interchange envelope.

**EDI Envelope Reference ID**

The interchange-level Envelope Reference ID (control number).

**Network Received Date**

The date on which the network received the interchange (format YYMMDD).

**Network Received Time**

The time at which the network received the interchange (format HHMMSS).

**Network Received Time Qualifier**

A 1- to 3-character Qualifier based on time zones. A blank value indicates local time.

Examples:

GMT = Greenwich Mean Time

EST = Eastern Standard Time

**Network Error Count**

The number of errors encountered by the network on receipt of this interchange.

**Network Processed Date**

The date on which the interchange was processed by the network (format YYMMDD).

**Network Processed Time**

The time at which the interchange was processed by the network (format HHMMSS).

**Network Processed Time Qualifier**

A 1- to 3-character Qualifier based on time zones. A blank value indicates local time.

Examples:

GMT = Greenwich Mean Time

EST = Eastern Standard Time

**Network Processed Error Count**

The number of errors encountered by the network on the process event of this interchange.

**Network Picked Up Date**

The date on which the interchange was picked up from the network (YYMMDD format).  
If the date is returned at the functional group or transaction level, this is the date on the last transaction or group in the interchange.

**Network Picked Up Time**

The time at which the interchange was picked up from the network (HHMMSS format).

**Network Picked Up Time Qualifier**

A 1- to 3-character Qualifier based on time zones. A blank value indicates the local time.

Examples:

GMT = Greenwich Mean Time

EST = Eastern Standard Time

**Network Picked Up Error Count**

The number of errors encountered by the network on the process event of this interchange.

**Network Receive Error Detail**

Description of any errors encountered on receipt of the interchange.

**Network Processed Error Detail**

Description of any errors encountered while processing the interchange.

**Network Picked Up Error Detail**

Description of any errors encountered when the interchange was picked up from the network.

**Record Format Indicator**

The format of the Universal Network Response file. Initially A.





# Reporting Programs

## Overview

This chapter contains the following topics related to the reporting programs:

<b>Topic</b>	<b>Page</b>
Acknowledgment Report.....	EBDI017, EBDI018..... 7-2
Application Change Audit Inquiry .....	EBDI062 ..... 7-10
Code Table Change Audit Inquiry.....	EBDI064 ..... 7-13
Configuration Change Audit Inquiry Report Program .....	EBDI070 ..... 7-16
EDI Data Print Report .....	EBDI019 ..... 7-19
Error Message Change Audit Inquiry.....	EBDI069 ..... 7-24
Global Parameter Change Audit Inquiry .....	EBDI071 ..... 7-27
JCL File Report .....	EDIRJCLR..... 7-29
Mapping Cross Reference Report.....	EBDI091 ..... 7-32
Mapping Report – Application Sequence.....	EBDI053 ..... 7-42
Mapping Report – Standard Sequence .....	EBDI052 ..... 7-50
Mapping Report Summary .....	EBDI061 ..... 7-59
Mapping Validation Report .....	EBDI082 ..... 7-66
Partner Change Audit Inquiry Report Program.....	EBDI092 ..... 7-72
Partner File Inquiry Report.....	EBDI008 ..... 7-76
Partner Profile Print Report .....	EBDI006 ..... 7-82
Security Change Audit Inquiry.....	EBDI073 ..... 7-93
Security File Display Report .....	EBDI009 ..... 7-95
Separator Change Audit Inquiry.....	EBDI072 ..... 7-97
Standard Change Audit Inquiry Report Program .....	EBDI037 ..... 7-100
Standard Optimized Table Report in X12 Format.....	EBDI035 ..... 7-105
Standard Report .....	EBDI036 ..... 7-116
Transaction Change Audit Inquiry Report Program.....	EBDI063 ..... 7-128

## Acknowledgment Report

## EBDI017, EBDI018

The Acknowledgment report lists the acknowledgments received from trading partners, detailing acceptance, error, and rejection messages. To create this report, the inbound acknowledgments must be directed to a user selected split file. Records on the selected split file are read by program EBDI017, which selects the acknowledgment records and writes them to a temporary keyed file. Acknowledgment interchanges are identified as follows: an interchange record followed by a 'TA1' record, or an interchange record followed by a 'GS' followed by a '997' or '999' ST segment record. Program EBDI017 then removes the sequence number used in the split file, and adds a 14 byte sequence number which is used as the key. This allows the records to be accessed in an appropriate sequence to create the report.

Program EBDI018 reads the temporary file created by program EBDI017 in key sequence. Program EBDI018 then uses the version/release number from the interchange or group record to access the online standard element and code files to interpret the error code, and create the acknowledgment report.

The type of interchange received (i.e., one format for ISA/ICS interchanges and another format for BG/GS interchanges) determines the format in which the Received Acknowledgment report prints.

### Files Used (Program EBDI017)

#### Input

The following file is used as input:

Filename	Description
SYS040	Split File This is a user-defined split file containing the inbound acknowledgments.

#### Output

The following file is used as output:

Filename	Description
SYS050	Work Acknowledgment File This temporary file contains the acknowledgment interchanges selected from the split file.

**Files Used (EBDI018 Program)***Input*

The following files are used as input:

<b>Filename</b>	<b>Description</b>
EDISSEG	Standard Segment File This file contains all the standard segments.
EDISELE	Standard Element File This file contains all the standard elements.
EDISCD1	Standard Code File 1 This file contains all the standard codes with a length between 1 and 5.
EDISASC	Standard Association File
EDICFG	System Configuration File
SYS050	Work Acknowledgment File This file is created by the EBDI017 program. It contains the acknowledgment interchanges selected from the split file.
SYS090	Partner Profile This file contains all the user-defined partner records.
SYS095	Partner X-Ref File This file contains the partner cross-reference records. The file is used only when processing in the Partner/Qualifier mode.
EDIPREL	Partner Relationship File This file is used when processing in the Relationship mode that cross-references EDI IDs to User/Partner IDs.

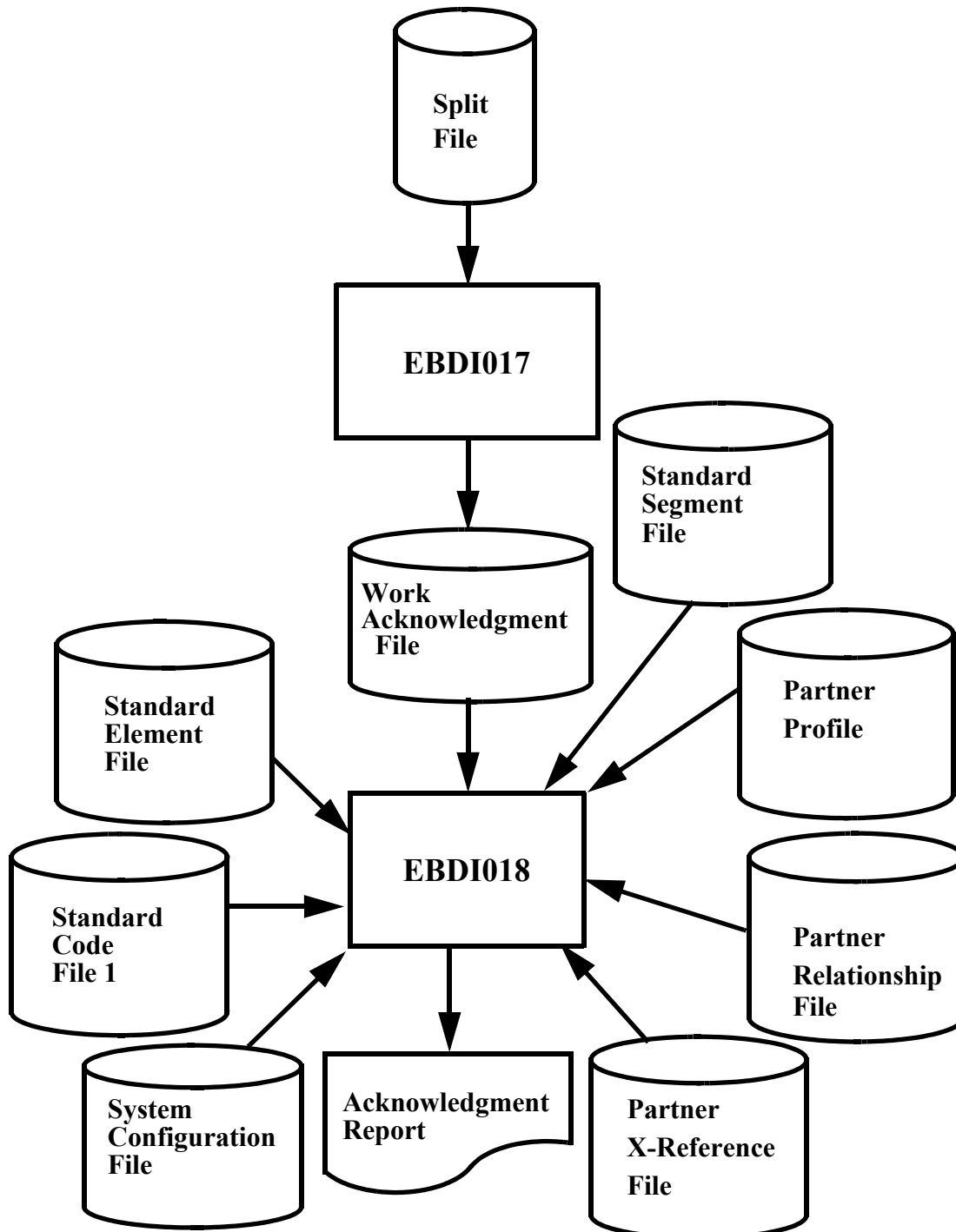
*Output*

The following files are used as output:

<b>Filename</b>	<b>Description</b>
SYS002	Acknowledgment Report This file contains the Acknowledgment report.

*Acknowledgment Report Process*

Figure 4.1 illustrates the Acknowledgment Report process.



**Figure 4.1 Programs EBDI017 and EBDI018 – Acknowledgment Report**

Example Reports

The following figures are examples of the Acknowledgment report.

```

EBDI018  DATE 12/01/2005          GENTRAN ACKNOWLEDGEMENT REPORT          TIME 12:00  PAGE 1
SENDER ID: VENDOR-1              QUAL:
INTERCHANGE CONTROL NO: 000000001  SENDER NAME: ABC COMPUTER COMPANY  RECEIVER ID: STERLING
GROUP CONTROL NO: 000000001      FUNCTIONAL ID: PO  INCLUDED SETS: 000004  RECEIVED SETS: 000004  ACCEPTED SETS: 000004
CODE MESSAGE
-----
A  ** GROUP ACCEPTED. **
TRANSACTION ID: 850  CONTROL NO: 000000073
CODE MESSAGE
-----
E  ** TRANSACTION ACCEPTED WITH ERRORS. **
SEG ID/VER: AMT 00 POS: 000001  LOOP ID: 0100
CODE MESSAGE
-----
1  UNRECOGNIZED SEGMENT ID
2  ENEXPECTED SEGMENT
3  MANDATORY SEGMENT MISSING
ELEMENTS:
POS  REF  CODE  DATA  MESSAGE
---  ---  ---  ---  ---
01  522  1      MANDATORY DATA ELEMENT MISSING
TRANSACTION ID: 850  CONTROL NO: 000000074
CODE MESSAGE
-----
A  ** TRANSACTION ACCEPTED WITH ERRORS **
    
```

Figure 4.2 Acknowledgment Report (ISA/ICS Interchanges)

```

EBDI018  DATE 12/01/2005          GENTRAN ACKNOWLEDGEMENT REPORT          TIME 12:03  PAGE  2
SENDER ID: VENDOR-1              QUAL:          RECEIVER ID: STERLING
INTERCHANGE CONTROL NO: 00000004  SENDER NAME: ABC COMPUTER COMPANY
GROUP CONTROL NO: 0000000032      RECEIVED SETS: 000003      ACCEPTED SETS: 000003
SEGMENT/ELEMENTS:
SEG//POS  REJECT SET IDENTIFIER  CODE  DATA  MESSAGE
-----
PO1/03    NUMERIC CODE                D    %$#@889940  INCORRECT DATA TYPE
PO1/04    AMOUNT CODE                  L    +0000000000  INCORRECT FIELD LENGTH
    
```

Figure 4.3 Acknowledgment Report (BG/GS Interchanges)

## *Field Definitions*

### **ISA/ICS Interchanges**

**Sender ID:**

This field displays the sender identifier from the interchange record.

**Qual:**

This field identifies the qualifier code associated with the Sender ID.

**Receiver ID:**

This field displays the receiver identifier from the interchange record.

**Qual:**

This field identifies the qualifier code associated with the Receiver ID.

### **Interchange Acknowledgment**

**Interchange Control No:**

This field designates the interchange control sequence number found on the interchange record.

**Sender Name:**

This field indicates the name of the sender from the partner profile.

**Code:**

This field identifies the error code of the interchange. This information is extracted from the interchange acknowledgment and its values are defined in the standards guide.

**Message:**

This field provides descriptive information about the error code from the standard element and code files.

### **Group Acknowledgment**

**Group Control No:**

This field designates the group control number from the group envelope.

**Functional ID:**

This field specifies the functional group identifier.

**Included Sets:**

This field identifies the total number of transaction sets included in the group.

**Received Sets:**

This field identifies the total number of transaction sets received.

**Accepted Sets:**

This field identifies the total number of transaction sets accepted.

**Code:**

This field identifies the error code of the group. This information is extracted from the interchange acknowledgment and its values are defined in the standards guide.

**Message:**

This field provides descriptive information about the error code from the standard element and code files.

**Transaction Acknowledgments****Transaction ID:**

This field specifies the transaction identifier.

**Control No:**

This field identifies the control number from the transaction envelope.

**Code:**

This field identifies the error code of the transaction. This information is extracted from the interchange acknowledgment and its values are defined in the standards guide.

**Message:**

This field provides descriptive information about the error code, from the standard element and code files.

**Segments****Segment ID:**

This field specifies the segment identifier.

**Pos:**

This field identifies the location of the segment in the transaction.

**Loop ID:**

This field displays the loop identifier.

**Code:**

This field identifies the error code of the segment. This information is extracted from the interchange acknowledgment and its values are defined in the standards guide.

**Message:**

This field provides descriptive information about the error code, from the standard element and code files.

**Elements****Pos:**

This field identifies the location of the element in the segment.

**Ref:**

This field identifies the dictionary reference number. This value is extracted from the interchange acknowledgment.

**Code:**

This field identifies the error code of the element. This information is extracted from the interchange acknowledgment and its values are defined in the standards guide.

**Data:**

This field displays the actual data in the field that caused the error.

**Message:**

This field provides descriptive information about the error code from the standard element and code files.



**BG/GS Interchanges****Sender ID:**

This field displays the sender identifier from the interchange record.

**Qual:**

This field identifies the qualifier code associated with the Sender ID.

**Receiver ID:**

This field displays the receiver identifier from the interchange record.

**Qual:**

This field identifies the qualifier code associated with the Receiver ID.

**Interchange Control No:**

This field designates the interchange control sequence number found on the interchange record.

**Sender Name:**

This field indicates the name of the sender from the partner profile.

**Group Control No:**

This field designates the group control number from the group envelope.

**Received Sets:**

This field identifies the total number of transaction sets received.

**Accepted Sets:**

This field identifies the total number of transaction sets accepted.

**Segment/Element****Seg/Pos:**

The Seg field displays the segment identifier. It is followed by a slash (/) and the Pos field, which identifies the position of the element in the segment.

**Reject Set Identifier:**

This field displays the reject set identifier, which is taken from the A1 segment record.

**Code:**

This field identifies the error code of the group. This code reads the dictionary file to extract the corresponding message.

**Data:**

This field displays the actual data in the field that caused the error.

**Message:**

This field provides descriptive information about the error code, from the dictionary file.

## Application Change Audit Inquiry

**EBDI062**

The Application Change Audit Inquiry Report program (EBDI062) enables you to produce reports for Application maintenance updates that have been applied to the four online application files. This program produces either a summary level report of activity or a detail report that shows all update activity, including before and after images of any "edit" updates.

The Application change audit file contains records that track changes made to the four application files (application header, record, field, and link). Add, delete, and update functions via the online Application subsystem result in records being written to the Application Change Audit file. The Batch copy utility (EBDI060) and Batch Application Re-sequence (EBDI090) will also write records to the Application change audit file indicating add, delete, replace and re-sequence functions. The online Upload process (EDIX235), using component EDIR060, will also write a change audit record to indicate an add or replace event.

### Operation

The Application Change Audit Inquiry program runs as a batch job step.

### Files Used

#### *Input*

The following files are used as input:

Filename	Description
EDIACHA	Application Change Audit File
EDICNTL	Control Card File
ERRCTL	Error Message and Control File
EDICFG	System Configuration File

#### *Output*

None.

#### *Reports*

Program EBDI062 produces the following reports:

Report Name	Description
EDIRPT	Application Change Audit Inquiry Report
EDILOG	Application Change Audit Processing Log
EDISUM	Application Change Audit Summary Report

## Control Card Options

The following Application Change Audit Inquiry operations and selection criteria are specified by control cards.

### Operations

SELECT (Default Operation)

### Selection Criteria

#### Defaults

FROM-UPDATE-DATE	01/01/1900
FROM-UPDATE-TIME	00:00
TO-UPDATE-DATE	12/31/2099
TO-UPDATE-TIME	23:59
USER	ALL
REPORT -TYPE	SUMMARY
APPLICATION-ID	ALL
SEND-RECEIVE	ALL

### Valid Criteria Values

#### From-Update--Date

Specifies the date that the update was applied to the Application. Starts the range. Used with To-Update-Date.

MM/DD/YYYY

#### From-Update-Time

Specifies the time that the update was applied to the Application. Starts the range. Used with To-Update-Time.

HH:MM

#### To-Update--Date

Specifies the date that the update was applied to the Application. Ends the range. Used with From-Update-Date.

MM/DD/YYYY

#### To-Update-Time

Specifies the time that the update was applied to the Application. Ends the range. Used with From-Update-Time.

HH:MM

#### User

Specifies the ID of online user who applied the change to the Application records.

ALL  
XXX

**Report-Type**

Identifies the level at which the Inquiry should report.

FULL	Includes detailed information for each update.
SUMMARY	Includes summary list of updates.

**Application-Id**

Identifies the application ID used to restrict which Application change audit records are used to create the inquiry report.

ALL	Indicates that all application records will be used to create the inquiry report.
XXXXXXXXXX	Includes only the change audit records found for the specific Application ID.

**Send-Receive**

Specifies the direction of the application. Used with Application-Id to restrict the records used to create the inquiry report.

S	–	Send
R	–	Receive
ALL	–	Send and Receive

**Code Table Change Audit Inquiry****EBDI064**

The Code Table Change Audit Inquiry Report program (EBDI064) enables you to produce reports for Code Table maintenance updates that have been applied to the four online Code Table files. This program produces either a summary level report of activity or a detail report that shows all update activity, including before and after images of any "edit" updates.

The Code Table change audit file contains records that track changes made to the four Code Table files (code define, code, data, and validation). Add, delete, and update functions via the online Code Table subsystem result in records being written to the Code Table Change Audit file. The Batch copy utility (EBDI076) will also write records to the Code Table change audit file indicating add, delete, and replace functions. The online Upload process (EDIX235), using component EDIR076, will also write a change audit record to indicate an add or replace event.

**Operation**

The Code Table Change Audit Inquiry program runs as a batch job step.

**Files Used***Input*

The following files are used as input:

Filename	Description
EDICCHA	Code Table Change Audit File
EDICNTL	Control Card File
ERRCTL	Error Message and Control File
EDICFG	System Configuration File

*Output*

None.

*Reports*

Program EBDI064 produces the following reports:

Report Name	Description
EDIRPT	Code Table Change Audit Inquiry Report
EDILOG	Code Table Change Audit Processing Log
EDISUM	Code Table Change Audit Summary Report

## Control Card Options

The following Code Table Change Audit Inquiry operations and selection criteria are specified by control cards.

### Operations

SELECT (Default Operation)

### Selection Criteria

#### Defaults

FROM-UPDATE-DATE	01/01/1900
FROM-UPDATE-TIME	00:00
TO-UPDATE-DATE	12/31/2099
TO-UPDATE-TIME	23:59
USER	ALL
REPORT -TYPE	SUMMARY
TABLE-ID	ALL

### Valid Criteria Values

#### From-Update--Date

Specifies the date that the update was applied to the Code Table. Starts the range. Used with To-Update-Date.

MM/DD/YYYY

#### From-Update-Time

Specifies the time that the update was applied to the Code Table. Starts the range. Used with To-Update-Time.

HH:MM

#### To-Update--Date

Specifies the date that the update was applied to the Code Table. Ends the range. Used with From-Update-Date.

MM/DD/YYYY

#### To-Update-Time

Specifies the time that the update was applied to the Code Table. Ends the range. Used with From-Update-Time.

HH:MM

#### User

Specifies the ID of online user who applied the change to the Code Table records.

ALL

XXX

**Report-Type**

Identifies the level at which the Inquiry should report.

FULL            Includes detailed information for each update.

SUMMARY       Includes summary list of updates.

**Table-Id**

Identifies the table ID used to restrict which Code Table change audit records are used to create the inquiry report.

ALL             Indicates that all application records will be used to create the inquiry report.

XXXXXXXXXX    Includes only the change audit records found for the specific Table ID.

## Configuration Change Audit Inquiry Report Program

**EBDI070**

The Configuration Change Audit Inquiry Report program (EBDI070) enables you to produce reports for Configuration maintenance updates that have been applied to the system application file. This program produces detail report that shows all update activity, including before and after images of any “edit” updates.

Update functions via the online System Configuration subsystem result in records being written to the Configuration Change Audit file.

### Operation

The Configuration Change Audit Inquiry program runs as a batch job step.

### Files Used

#### *Input*

The following files are used as input:

Filename	Description
EDIYCHA	Configuration Change Audit File
EDICNTL	Control Card File
ERRCTL	Error Message and Control File
EDICFG	System Configuration File

#### *Output*

None.

#### *Reports*

Program EBDI070 produces the following reports:

Report Name	Description
EDIRPT	Configuration Change Audit Inquiry Report
EDILOG	Configuration Change Audit Processing Log
EDISUM	Configuration Change Audit Summary Report



## Control Card Options

The following Configuration Change Audit Inquiry operations and selection criteria are specified by control cards.

### Operations

SELECT (Default Operation)

### Selection Criteria

#### Defaults

FROM-UPDATE-DATE	01/01/1900
FROM-UPDATE-TIME	00:00
TO-UPDATE-DATE	12/31/2099
TO-UPDATE-TIME	23:59
USER	ALL
RECORD -TYPE	ALL

### Valid Criteria Values

#### From-Update--Date

Specifies the date that the update was applied to the configuration record. Starts the range. Used with To-Update-Date.

MM/DD/YYYY

#### From-Update-Time

Specifies the time that the update was applied to the configuration record. Starts the range. Used with To-Update-Time.

HH:MM

#### To-Update--Date

Specifies the date that the update was applied to the configuration record. Ends the range. Used with From-Update-Date.

MM/DD/YYYY

#### To-Update-Time

Specifies the time that the update was applied to the configuration record. Ends the range. Used with From-Update-Time.

HH:MM

#### User

Specifies the ID of online user who applied the change to the configuration records.

ALL

#### Record-Type

Identifies the system configuration record type used to restrict which configuration change audit records are used to create the inquiry report.

ALL	Indicates that all application records will be used to create the inquiry report.
B	Gentran:Realtime processing option
E	Gentran:Control processing option

T	Clear Key processing option
V	Commerce:Advantage processing option
0	Online processing option
1	Additional online processing options
2	Databank processing options
3	Gentran:Plus processing options
4	Gentran:Viewpoint processing options
7	Gentran:Structure processing options

## EDI Data Print Report

## EBDI019

The EDI Data Print report is generated by the EBDI019 batch program. This report enables you to print EDI information contained in the transaction record for either inbound or outbound processing. This program reports the segments and the contents of all the elements, along with a description of each element.

**Note:** Object (Binary) data is not printed on this report.

This program can be run using EDI delimited data only (wrapped or unwrapped). EDI data can be unwrapped through the Unwrap program (EBDI010). This program accepts only 80-byte EDI data as input.

Three global parameter records used in the Editor (EBDI001) are available with program EBDI019:

- VERSION xxxxxxxxxxxx5xx5xxxxxxxxxxxx5xx (Version Conversion)
- ICS TABLES ON  
ICS TABLES OFF
- BG PARTNER YES

See Appendix A in this guide for further information on these input parameters.

### Files Used

#### Input

The following files are used as input:

Filename	Description
SYS001	Optimized Table File This file contains the Optimized Standards Table file records.
ERRCTL	Error Message and Control File This file contains Gentran:Basic error messages.
SYS030	Parameter File This file contains optional parameters that determine the way the program interprets the data.
SYS040	Input EDI Data File This file contains the 80-byte wrapped or unwrapped EDI data.
SYS090	Partner Profile This file contains user-defined partner records.
SYS095	Partner X-Ref File This file contains the partner Cross-Reference records. The file is used only when processing in the Partner/Qualifier mode.
EDICFG	System Configuration File

Filename	Description
EDISELD	Standard Element Description File This file contains the descriptions for all elements.
EDISCD1	Standard Code File 1 This file contains code records for elements with a maximum length between 1 and 5.
EDISCD2	Standard Code File 2 This file contains code records for elements with a maximum length between 6 and 10.
EDISCD3	Standard Code File 3 This file contains code records for elements with a maximum length between 11 and 20.
EDISCD4	Standard Code File 4 This file contains code records for elements with a maximum length between 21 and 40.
EDIPREL	Partner Relationship File This file is used when processing in the partner Relationship mode. It contains the EDI ID to partner relationship User/Partner ID conversion.

### Output

The following files are used as output:

Filename	Description
SYS010	EDI Data Print Report This file contains the EDI Data Print report.

See Appendix A in this guide for a complete description and layout of the parameters used in this program.

EDI Data Print Report Process

Figure 4.4 illustrates the EDI Data Print report process.

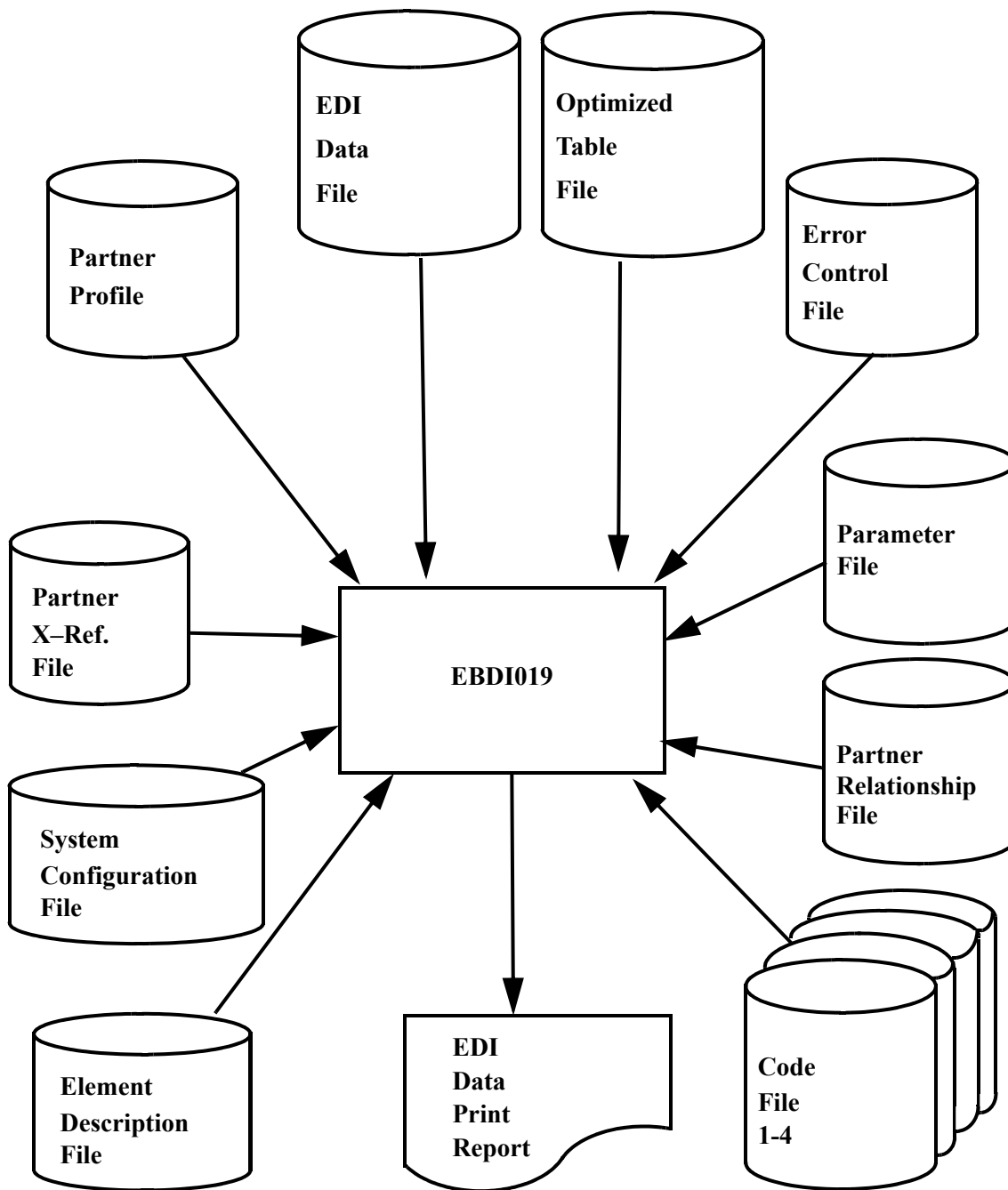


Figure 4.4 EDI Data Print Report – Program EBDI019

Example Report

Figure 4.5 illustrates the EDI Data Print report.

PAGE 1

EBDI019 RUN 12/01/2005 TIME 12:00 GENTRAN EDI DATA PRINT REPORT

VERSION.....: 00300 AGENCY.: X

TRANSACTION.....:

```

SEGMENT ID:  ISA  SEGMENT VER:  00
1  00  TYPE:ID MIN/MAX:  2  2 AUTHORIZATION INFORMATION QUALIFIER
2  00  TYPE:AN MIN/MAX: 10 10 AUTHORIZATION INFORMATION
3  00  TYPE:ID MIN/MAX:  2  2 SECURITY INFORMATION QUALIFIER
4  00  TYPE:AN MIN/MAX: 10 10 SECURITY INFORMATION
5  ZZ  TYPE:ID MIN/MAX:  2  2 INTERCHANGE ID QUALIFIER
6  51366666666  TYPE:ID MIN/MAX: 15 15 INTERCHANGE SENDER ID
7  ZZ  TYPE:ID MIN/MAX:  2  2 INTERCHANGE ID QUALIFIER
8  6147937000  TYPE:ID MIN/MAX: 15 15 INTERCHANGE RECEIVER ID
9  950105  TYPE:DT MIN/MAX:  6  6 DATE
10 1200  TYPE:TM MIN/MAX:  4  4 TIME
11 U  TYPE:ID MIN/MAX:  1  1 INTERCHANGE CONTROL STANDARDS ID CO
12 00300  TYPE:ID MIN/MAX:  5  5 INTERCHANGE VERSION
13 000000005  TYPE:NO MIN/MAX:  9  9 INTERCHANGE CONTROL NUMBER
14 0  TYPE:ID MIN/MAX:  1  1 ACKNOWLEDGE REQUESTED
15 P  TYPE:ID MIN/MAX:  1  1 TEST INDICATOR
16 *  TYPE:AN MIN/MAX:  1  1 SUBELEMENT SEPARATOR

SEGMENT ID:  GS  SEGMENT VER:  00
1  IN  TYPE:ID MIN/MAX:  2  2 FUNCTIONAL IDENTIFIER CODE
2  121212121  TYPE:AN MIN/MAX:  2  15 APPLICATION SENDER'S CODE
3  987654321  TYPE:AN MIN/MAX:  2  15 APPLICATION RECEIVER'S CODE
4  950105  TYPE:DT MIN/MAX:  6  6 DATA INTERCHANGE DATE
5  1200  TYPE:T6 MIN/MAX:  4  6 DATA INTERCHANGE TIME
6  7  TYPE:NO MIN/MAX:  1  9 GROUP CONTROL NUMBER
7  X  TYPE:ID MIN/MAX:  1  2 RESPONSIBLE AGENCY CODE
8  003030  TYPE:AN MIN/MAX:  1  12 VERSION / RELEASE / INDUSTRY IDENTIFI

SEGMENT ID:  ST  SEGMENT VER:  00
1  810  TYPE:ID MIN/MAX:  3  3 TRANSACTION SET IDENTIFIER CODE
2  000070001  TYPE:AN MIN/MAX:  4  9 TRANSACTION SET CONTROL NUMBER

SEGMENT ID:  BIG  SEGMENT VER:  00
1  950105  TYPE:DT MIN/MAX:  6  6 INVOICE DATE
2  INV01  TYPE:AN MIN/MAX:  1  22 INVOICE NUMBER
3  950103  TYPE:DT MIN/MAX:  6  6 PURCHASE ORDER DATE
4  PONUMBER-001  TYPE:AN MIN/MAX:  1  22 PURCHASE ORDER NUMBER

SEGMENT ID:  NTE  SEGMENT VER:  00
2  -----
2  TERMS ARE SPECIFIED BELOW
2  DISCOUNT WILL NOT BE APPLIED UNLESS INVOICE IS P
2  IN FULL BY THE DISCOUNT DUE DATE.
2  -----

SEGMENT ID:  REF  SEGMENT VER:  00
1  SL  TYPE:ID MIN/MAX:  2  2 REFERENCE NUMBER QUALIFIER
    
```

Figure 4.5 EDI Data Print Report

*Field Definitions***Version:**

This field displays the version identification.

**Agency:**

This field displays the agency code associated with the Version ID.

**Transaction:**

This field displays the transaction identification from the transaction record.

**Segment ID:**

This field displays the segment identifier for the specified elements.

**Segment Ver:**

This field displays the version of the segment displayed.

**(EDI Data Field):**

This field identifies the data within each element along with its corresponding element sequence number.

**ID:**

This field displays the element identification for the data field.

**Type:**

This field displays the data type for the Element ID.

**Min/Max:**

This field displays the minimum and maximum lengths for this element as specified by the standard.

**(Description):**

This field is a description of the element.

## Error Message Change Audit Inquiry

**EBDI069**

The Error Message Change Audit Inquiry Report program (EBDI069) enables you to produce reports for maintenance updates that have been applied to the Error Message records that reside on the Error Message and Control file. This program produces a detail report that shows all update activity, including before and after images of any “edit” updates.

Update functions via the online Error Message subsystem result in records being written to the Error Message Change Audit file. Also, the batch utility (EBDIT13) will generate change audit records when error messages are added or replaced.

### Operation

The Error Message Change Audit Inquiry program runs as a batch job step.

### Files Used

#### Input

The following files are used as input:

Filename	Description
EDIECHA	6.4 VSAM Error Message Change Audit File
EDICNTL	Control Card File
ERRCTL	Error Message and Control File
EDICFG	System Configuration File

#### Output

None.

#### Reports

Program EBDI069 produces the following reports:

Report Name	Description
EDIRPT	Error Message Change Audit Inquiry Report
EDILOG	Error Message Change Audit Processing Log
EDISUM	Error Message Change Audit Summary Report



## Control Card Options

The following Error Message Change Audit Inquiry operations and selection criteria are specified by control cards.

### Operations

SELECT (Default Operation)

### Selection Criteria

#### Defaults

FROM-UPDATE-DATE	01/01/1900
FROM-UPDATE-TIME	00:00
TO-UPDATE-DATE	12/31/2099
TO-UPDATE-TIME	23:59
USER	ALL
MESSAGE-NUMBER	ALL
RECORD-TYPE	ALL

### Valid Criteria Values

#### From-Update--Date

Specifies the date that the update was applied to the Error Message record. Starts the range. Used with To-Update-Date.

MM/DD/YYYY

#### From-Update-Time

Specifies the time that the update was applied to the Error Message record. Starts the range. Used with To-Update-Time.

HH:MM

#### To-Update--Date

Specifies the date that the update was applied to the Error Message record. Ends the range. Used with From-Update-Date.

MM/DD/YYYY

#### To-Update-Time

Specifies the time that the update was applied to the Error Message record. Ends the range. Used with From-Update-Time.

HH:MM

#### User

Specifies the ID of online user who applied the change to the Error Message records.

ALL

#### Message-Number

Specifies the Error Message number used to restrict which Error Message change audit records are used to create the inquiry report.

ALL All Error Message records are used to create the inquiry report.

XXXXX

**Record-Type**

Specifies the type of Error Message used to create the inquiry report.

ALL	All record types are used to create the inquiry report.
G	Only Gentran messages are used to create the inquiry report.
U	Only user messages are used to create the inquiry report.

## Global Parameter Change Audit Inquiry

**EBDI071**

The Global Parameter Change Audit Inquiry Report program (EBDI071) enables you to produce reports for maintenance updates that have been applied to the Global Parameter records that reside on the System Configuration file. This program produces detail report that shows all update activity, including before and after images of any “edit” updates.

Update functions via the online Global Parameter subsystem result in records being written to the Global Parameter Change Audit file.

### Operation

The Global Parameter Change Audit Inquiry program runs as a batch job step.

### Files Used

#### Input

The following files are used as input:

Filename	Description
EDIGCHA	6.4 VSAM Global Parameter Change Audit File
EDICNTL	Control Card File
ERRCTL	Error Message and Control File
EDICFG	System Configuration File

#### Output

None.

#### Reports

Program EBDI071 produces the following reports:

Report Name	Description
EDIRPT	Global Parameter Change Audit Inquiry Report
EDILOG	Global Parameter Change Audit Processing Log
EDISUM	Global Parameter Change Audit Summary Report

## Control Card Options

The following Global Parameter Change Audit Inquiry operations and selection criteria are specified by control cards.

### Operations

SELECT (Default Operation)

### Selection Criteria

#### Defaults

FROM-UPDATE-DATE	01/01/1900
FROM-UPDATE-TIME	00:00
TO-UPDATE-DATE	12/31/2099
TO-UPDATE-TIME	23:59
USER	ALL
RECORD -TYPE	ALL

### Valid Criteria Values

#### From-Update--Date

Specifies the date that the update was applied to the Global Parameter record. Starts the range. Used with To-Update-Date.

MM/DD/YYYY

#### From-Update-Time

Specifies the time that the update was applied to the Global Parameter record. Starts the range. Used with To-Update-Time.

HH:MM

#### To-Update--Date

Specifies the date that the update was applied to the Global Parameter record. Ends the range. Used with From-Update-Date.

MM/DD/YYYY

#### To-Update-Time

Specifies the time that the update was applied to the Global Parameter record. Ends the range. Used with From-Update-Time.

HH:MM

#### User

Specifies the ID of online user who applied the change to the Global Parameter records.

ALL

#### Record-Type

Identifies the system configuration record type used to restrict which Global Parameter change audit records are used to create the inquiry report.

ALL	Indicates that all Global Parameter records will be used to create the inquiry report.
GI	Inbound global parameters
GO	Outbound global parameters

## JCL File Report

## EDIRJCLR

The JCL File Report program (EDIRJCLR) provides a list of all JCL members that exist in the online JCL file (EDIRJCL).

### Files Used

#### *Input*

The following files are the inputs to the JCL File Report program:

Filename	Description
EDICFG	System Configuration file
ERRCTL	Error message and control file
EDIRJCL	Online JCL file

#### *Output*

The following files are the outputs from the JCL File Report program:

Filename	Description
SYS005	Summary Report
SYS006	JCL File Report

## JCL File Report Process

Figure 4.6 illustrates the Mapping Cross Reference report process.

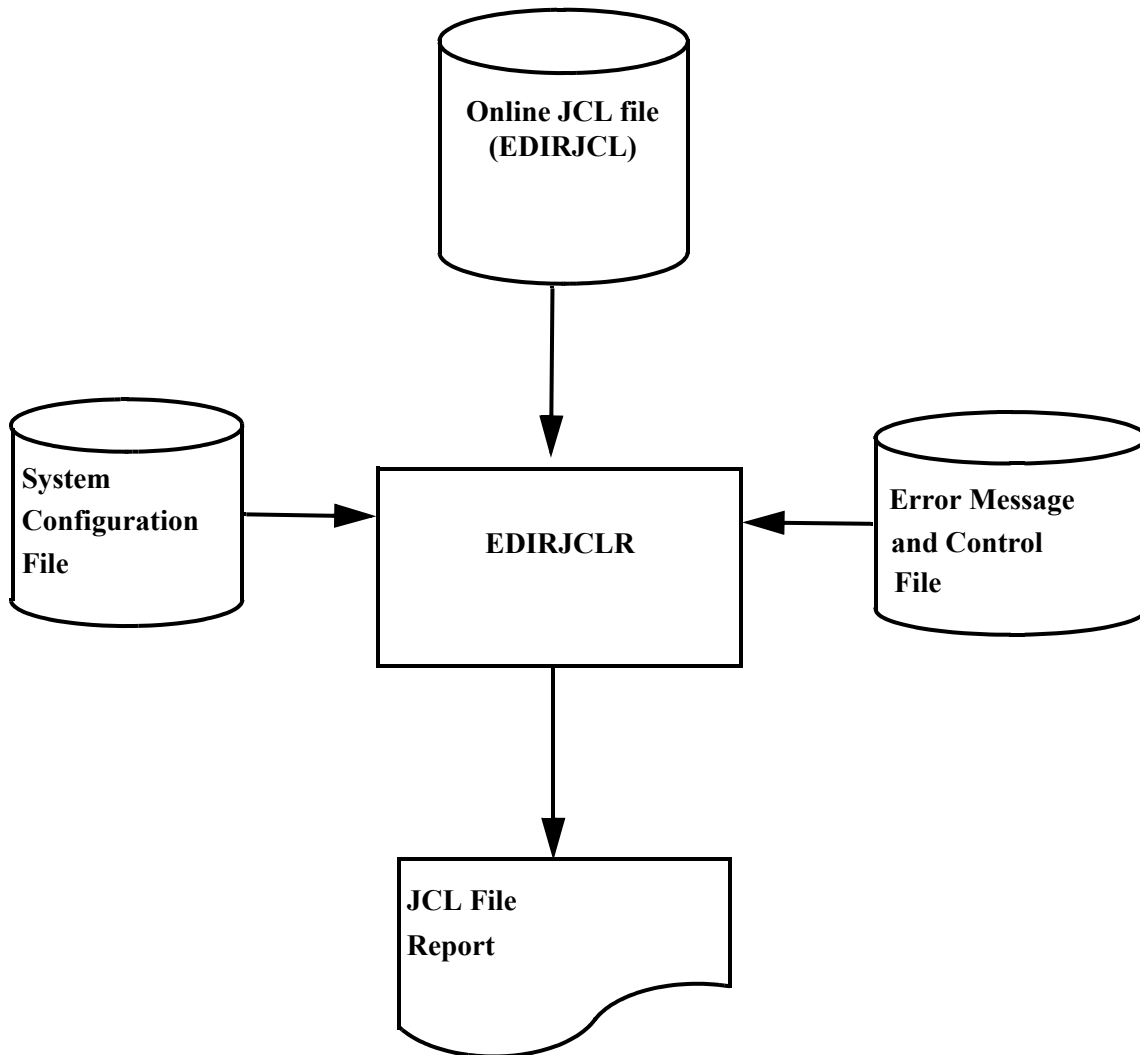


Figure 4.6 JCL File Report – Program EDIRJCLR

## Sample Reports

Figure 4.7 and Figure 4.8 illustrate the JCL File report and the Summary report.

```

EDIRJCLR  RUN 12/01/2005    TIME 12:00    GENTRAN: BASIC  EDIRJCL JCL FILE REPORT          PAGE    1
JCL MEMBER                RECORD COUNT
-----
EXECRPTM                    330
EXEC0060                    164
EXEC0360                    133
EXEC306                      66
EDIJACK                     100
EDIJIB                       84
EDIJIBX                     146
EDIJOB                       84
EDIJOBX                     147
EDIJRPT                      91
EDIJSB                       74
NO ERRORS OCCURRED DURING PROCESSING
PROCESSING ENDED NORMALLY - PROCESSING COUNTS BELOW
                                INPUT RECORDS READ----- 1419
                                PROGRAM RETURN CODE-----    0

```

**Figure 4.7 JCL File Report**

```

EDIRJCLR  RUN 12/01/2005    TIME 12:00    SUMMARY REPORT - GENTRAN: BASIC JCL REPORT          PAGE    1
PROCESSING BEGAN ON 12/01/2005 AT 12:00 PM.
INPUT RECORDS READ----- 1,419
JCL MEMBERS FOUND----- 11
PROCESSING ENDED ON 12/01/2005 AT 12:00 PM.

```

**Figure 4.8 JCL Summary Report**

## Mapping Cross Reference Report

**EBDI091**

The Mapping Cross Reference report is produced by executing the EBDI091 batch program. This report lists application, map, and trading partner associations. The Mapping Cross Reference report can be run three ways:

- *By Application.* A parameter record specifies the Application ID. The report lists all translation maps associated with the application and for partner-specific maps, all associated trading partners.
- *By Translation Map.* A parameter record specifies the Translation ID. The report lists the application associated with the map and for partner-specific maps, all associated trading partners.
- *By Trading Partner ID.* A parameter record specifies the Trading Partner ID (Partner ID/Qualifier in the Partner/Qualifier mode and User/Partner ID in the Relationship mode). The report lists each Translation Map and Application associated with the trading partner (relationship).

**Note:** You can only enter one parameter at a time in the parameter file.

### Files Used

#### Input

The following files are used as input:

Filename	Description
SYS020	Application Header File This file contains all the application header records.
SYS023	Application Link File This file contains all the application link records.
SYS040	Transaction Header File This file contains all the transaction header records.
SYS090	Partner Profile This file contains all partner profile records.
SYS030	Parameter File This file contains all the parameter file records.
EDICFG	System Configuration File



*Output*

The following file is used as output:

Filename	Description
SYS010	Output Report File This file contains the Mapping Cross Reference report.

*Parameter Record Layout – Application*

The following table describes the application parameter record layout for the Partner/Qualifier mode.

Position	Field Content	COBOL Picture	Format	Length	Description
001–012	Application=	X(12)	AN	12	APPLICATION=
013–022	Application ID	X(10)	AN	10	Application Identifier
023–027	␣S/R=	X(05)	AN	05	␣S/R=
028-028	Send or Receive Flag	X(01)	AN	01	Send or Receive flag. Valid values: <b>S</b> or <b>R</b>

The following table describes the application parameter record layout for the Relationship mode.

Position	Field Content	COBOL Picture	Format	Length	Description
001–005	User=	X(05)	AN	5	USER=
006–020	User ID	X(15)	AN	15	User portion of User/Partner ID
021–029	␣PARTNER=	X(09)	AN	09	␣PARTNER=
030-044	Partner ID	X(15)	AN	15	Partner portion of User/Partner ID

*Parameter Record Layout – Translation*

The following table describes the translation parameter record layout.

Position	Field Content	COBOL Picture	Format	Length	Description
001-004	Map=	X(04)	AN	04	<b>MAP=</b>
005-014	Translation ID	X(10)	AN	10	Translation identifier
015-019	↳S/R=	X(05)	AN	05	↳S/R=
020-020	Send or Receive Flag	X(01)	AN	01	Send or Receive flag. Valid values: <b>S</b> or <b>R</b>

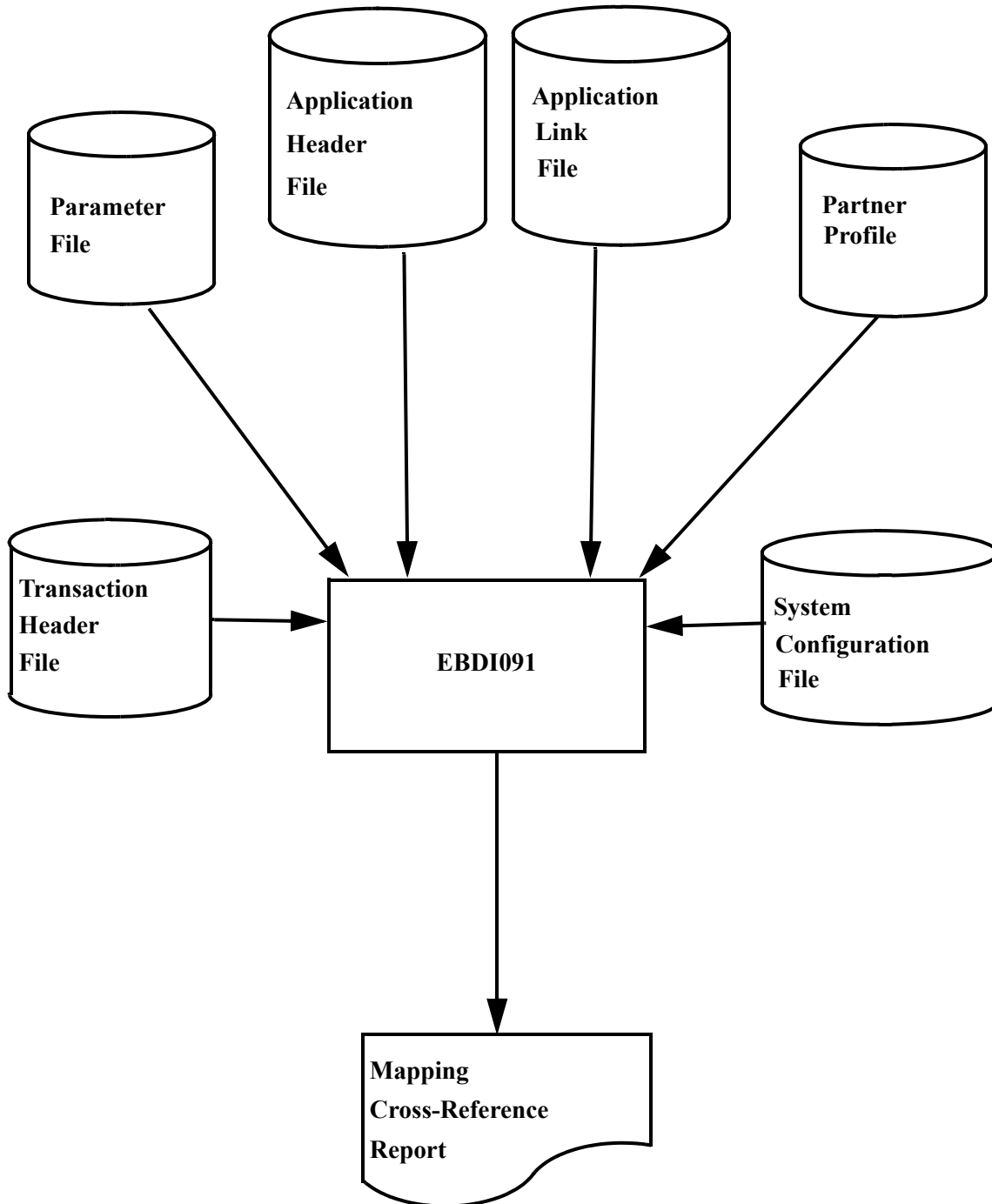
*Parameter Record Layout – Partner*

The following table describes the partner parameter record layout.

Position	Field Content	COBOL Picture	Format	Length	Description
001-008	Partner=	X(08)	AN	08	<b>PARTNER=</b>
009-043	Partner ID	X(35)	AN	35	Partner identifier
044-049	↳QUAL=	X(06)	AN	06	↳QUAL=
050-053	Partner Qualifier	X(04)	AN	04	Partner/Qualifier

*Mapping Cross Reference Report Process*

Figure 4.9 illustrates the Mapping Cross Reference report process.



**Figure 4.9 Program EBDI091 – Mapping Cross Reference Report**

Example Reports

Figure 4.10 through Figure 4.12 illustrate the Mapping Cross Reference report (by application), the Mapping Cross Reference report (by translation), and the Mapping Cross Reference report (by partner).

```

EBDI091      DATE: 12/01/2005      MAPPING CROSS-REFERENCE REPORT      PAGE 1
GENTRAN MAPPING INTEGRATION      BY APPLICATION      TIME: 12:00

APPLICATION ID.: PROFILE      DESCRIPTION.....: PURCHASE ORDER MASTER FILE      RECORD LENGTH.....: 250
SEND OR RECEIVE: S      FUNCTIONAL GROUP: PO ORDER      FIXED OR VARIABLE: V

-----
TRAN ID      TRAN STATUS      PARTNER SPECIFIC      STANDARD VERSION      TRANS SET      DESCRIPTION      PARTNER ID      PARTNER QUAL
-----
ANSI2040PO  PROD      N      002040      /X 850      ANSI 002040      OUTBOUND POS
ANSI3020SR  PROD      N      003020      /X 850      ANSI 002040      OUTBOUND POS
ANSI3030FO  PROD      N      003030      /X 850      ANSI 003030      OUTBOUND POS
DIV123      PROD      N      ANA001      /A ORDHDR      ANSI 002040      OUTBOUND POS
EDFCTD93PO  TEST      N      D 93A      /UN ORDERS      EDIFACT D93A      OUTBOUND ORDER
HCWT3010    PROD      N      003010      /X 850      ANSI 002040      OUTBOUND POS
TESTMAPOB  DEVEL      N      003042      /X 142      TEST MAP LOC SEGMENT      OUTBOU
    
```

Figure 4.10 Mapping Cross Reference Report (By Application)

```

EBDI091      DATE: 12/01/2005      MAPPING CROSS-REFERENCE REPORT      PAGE 1
GENTRAN MAPPING INTEGRATION      BY TRANSLATION      TIME: 12:00

TRANSLATION ID.: ANSI3030FO      DESCRIPTION: ANSI 003030 OUTBOUND POS      STANDARD VERSION: 003030
SEND OR RECEIVE: S      USE CODE...: GENERAL      TRANSACTION SET.: 850      /X

APPL REC FIXED      PARTNER
ID LEN /VAR FUNCTIONAL GROUP DESCRIPTION PARTNER ID QUAL
-----
POFILE 250 V PO ORDER PURCHASE ORDER MASTER FILE
    
```

Figure 4.11 Mapping Cross Reference Report (By Translation)

```

EBDI091    DATE: 12/01/2005    TIME: 11:47    PAGE 1
GENTRAN MAPPING INTEGRATION

MAPPING CROSS-REFERENCE REPORT
BY PARTNER

PARTNER ID.: CHRDETM101    QUAL:

TRAN ID S/R STATUS STANDARD TRANS
-----
858R302003 R TEST 003010 /X 858
                                INBOUND BILLS OF LADING
                                BIFILE1 825 F INBOUND BILLS OF LADING
                                REC LEN /VAR APPLICATION DESCRIPTION
                                -----

```

Figure 4.12 Mapping Cross Reference Report (By Partner)

*Field Definitions***By Application****Application ID:**

This field identifies the user-defined application definition name.

**Description:**

This field provides descriptive information about the Application ID.

**Record Length:**

This field displays the length of the application record.

**Send or Receive:**

This field specifies whether this is an inbound or an outbound application definition.

Valid values are:

S	–	Outbound Application Definition
R	–	Inbound Application Definition

**Functional Group:**

This field specifies the functional group identifiers defined on the application header record.

**Fixed or Variable:**

This field specifies whether the application record length is fixed or variable.

Valid values are:

F	–	Fixed
V	–	Variable

**Tran ID:**

This field specifies the translation map identification associated with the Application ID.

**Tran Status:**

This field displays the status of the transaction. Valid values are:

PROD	–	Production status
DEVEL	–	Development status
TEST	–	Test status

**Partner Specific:**

This field specifies whether the map is partner-specific (must be used with specific trading partners). Valid values are:

Y	–	Map is partner-specific
N	–	No, map is not partner-specific

**Standard Version:**

This field identifies the version of the EDI Standard defined by the translation map.

**Tran Set:**

This identifies the transaction set, as defined by the translation map.

**Description:**

Display a description of the Translation ID.

**Partner ID:**

Displays the partner identification associated with the translation map.

**Partner Qual:**

Displays the qualifier identification associated with the Partner ID, if applicable.

**By Translation****Translation ID:**

This field identifies the user-defined translation definition name.

**Description:**

This field provides descriptive information about the Translation ID.

**Standard Version:**

This field identifies the version of the EDI Standard defined by the translation map.

**Send or Receive:**

This field specifies whether this is an inbound or outbound translation map. Valid values are:

S	–	Outbound translation map
R	–	Inbound translation map

**Use Code:**

This field defines the transaction map use code. The Use Code identifies whether the transaction map definition may be used for general transaction purposes or for translating data for a particular group of trading partners. Valid values are:

GENERAL	–	General (may be used with any trading partner)
PARTNER-SPECIFIC	–	Partner-specific (must be used for a specified trading partner.)

**Transaction Set:**

This identifies the transaction set, as defined by the translation map.

**Appl ID:**

This field identifies the user-defined application definition name associated with the Translation ID.

**Rec Len:**

Displays the length of the application record.

**Fixed/Var:**

This field specifies whether the application record length is fixed or variable. Valid values are:

F	–	Fixed
V	–	Variable

**Functional Group:**

This field specifies the functional group identifier(s) defined on the application header record.

**Description:**

This field provides descriptive information about the Application ID.



**Partner ID:**

Lists the Partner IDs associated with the Translation ID.

**Partner Qual:**

Displays the qualifier identification associated with the Partner ID, if applicable.

**By Partner****Partner ID:**

This field identifies the user-defined Partner ID name.

**Partner Qual:**

Displays the qualifier identification associated with the Partner ID, if applicable.

**Tran ID:**

This field defines the translation map identification associated with the Partner ID.

**S/R:**

This field specifies whether the translation map is defined as inbound.

Valid values are:

S	–	Outbound translation map
R	–	Inbound translation map

**Trans Status:**

This field displays the status of the transaction. Valid values are:

PROD	–	Production status
DEVEL	–	Development status
TEST	–	Test status

**Standard Version:**

This field identifies the version of the EDI Standard defined by the translation map.

**Trans Set:**

This identifies the transaction set, as defined by the translation map.

**Translation Description:**

This field provides descriptive information about the translation map.

**Appl ID:**

This field defines the application identification associated with the translation map.

**Rec Len:**

Displays the length of the application record.

**Fixed/Var:**

This field specifies whether the application record length is fixed or variable.

Valid values are:

F	–	Fixed
V	–	Variable

**Application Description:**

This field provides descriptive information about the Application ID.

## Mapping Report – Application Sequence

**EBDI053**

The Mapping report shown in Application Sequence illustrates the current map of the records, along with the fields in your application files to the segments and data elements in the EDI Standard for the selected transaction. The report is sorted by application record sequence number, illustrating each application field within the record. The right half of the report shows the corresponding standard segment and element to or from which the application fields are mapped.

The Application Mapping report is produced by using the Data Mapping Report program – Application Sequence (EBDI053). This program produces a complete report detailing the mapping of a transaction from an Application file to a Standard file (outbound) or from a Standard file to an Application file (inbound), in Application Sequence.

The summary file created by EBDI053 is passed to EBDI061. This program prints a cross reference report of all constants, accumulators, hash total, tables, and user exits used in the map.

### Files Used

#### *Input*

The following files are used as input:

Filename	Description
SYS001	Parameter file This file contains the parameters that determine what is produced on the report.
SYS020	Application Data Definition This file contains the definition of the application data file.
SYS021	Application Record Definition This file contains the definition of the records comprising the application data file.
SYS0211	Application Record Definition Path
SYS022	Application Field Definition This file contains the definition of the fields comprising the application data file.
SYS0221	Application Field Definition Path
SYS040	Transaction Definition This file contains the definition of the transaction to be mapped.
SYS041	Transaction Segment Definition This file contains the definition of the segments to be mapped for the specified transaction.
SYS0411	Transaction Segment Definition Path
SYS042	Transaction Element Definition This file contains the definition of the elements to be mapped for the specified transaction.

Filename	Description
SYS0421	Transaction Element Definition Path
EDICFG	System Configuration File

### Output

The following files are used as output:

Filename	Description
SYS005	Mapping Report – Application Sequence This file contains the Mapping report in Application Sequence.
SYS010	Summary Report File This file contains all summary information of hash-totals, constants, accumulators, user-exits, and tables.

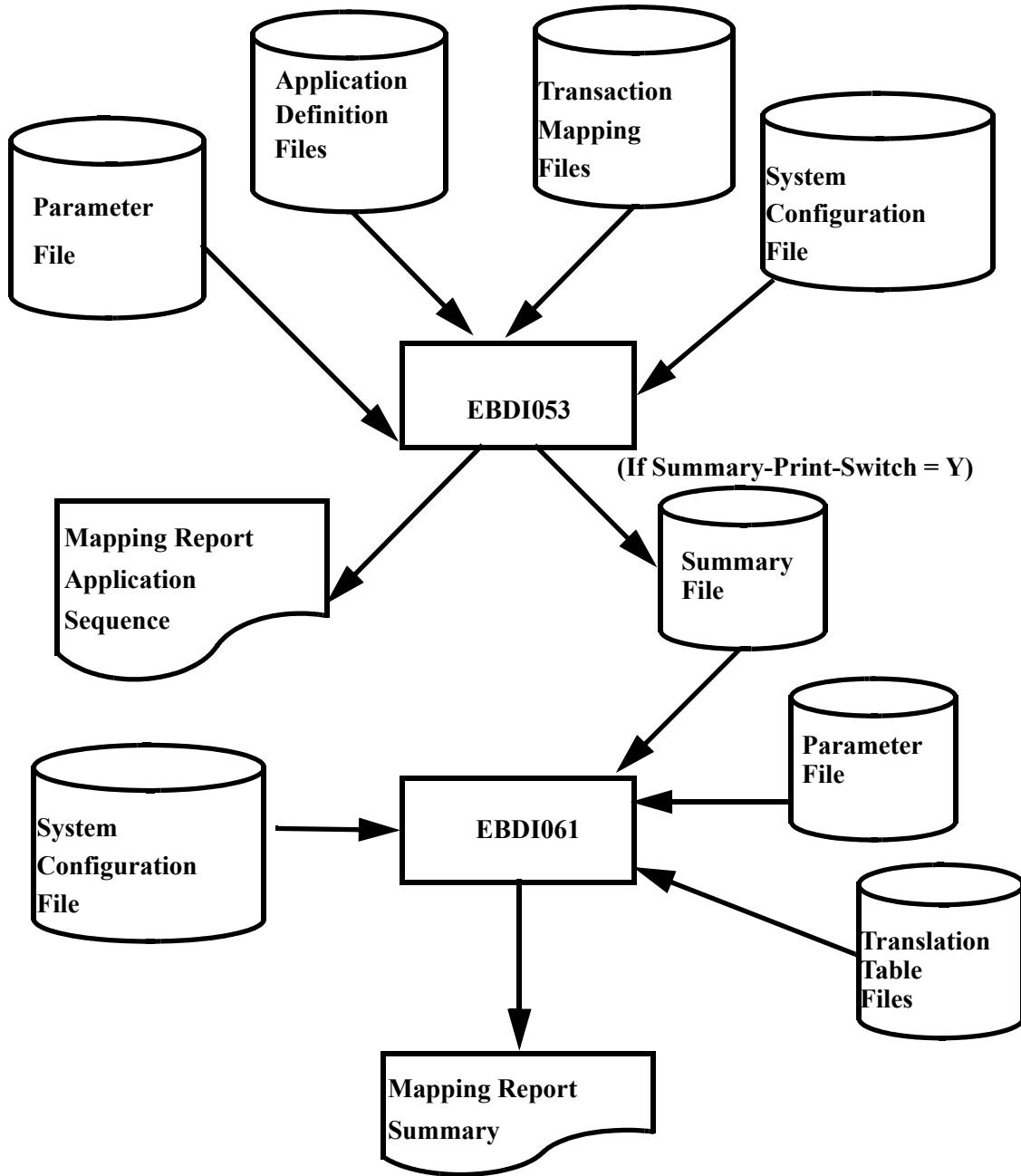
### Parameter File Layout

The following table describes the parameter record layout.

Position	Field Content	COBOL Picture	Format	Length	Description
001 – 010	Transaction ID	X(10)	AN	10	The name assigned to the definition of the transaction mapped.
011 – 011	Transaction Send/Receive Flag	X(01)	AN	01	Used to specify if the transaction is inbound or outbound. Valid values: <b>S</b> – Outbound transaction <b>R</b> – Inbound transaction.
012 – 014	Report Option	X(03)	AN	03	Used to specify which elements to display on the report. Valid values: <b>ALL</b> – To display all elements on the report. <b>MAP</b> – To display only the elements that were mapped.
015 – 015	Summary Print Switch	X(01)	AN	01	Used to specify whether to produce a summary report. Valid values: <b>Y</b> – Produce a summary report. <b>N</b> – Do not produce summary report.
016 – 080	Filler	X(65)	AN	65	Not used

*Mapping Report in Application Sequence Process*

Figure 4.13 illustrates the Mapping report in Application Sequence process.



**Figure 4.13 Program EBDI053 – Mapping Report – Application Sequence**

Example Report

Figure 4.14 illustrates the Mapping report in Application Sequence.

SEQ	M	TYPE	H/D	LOOP	MAX	U	DESCRIPTION	ALT	SEG	WRITE	FLAG	DESCRIPTION	TYPE	MIN	MAX	MSG
SEQ	C	ELEMENT		TYP	LEN	START		MAP	SEQ	SEG/ELEM						
010	M	001	H	AN	8	1	PO HEADER RECORD						AN	2	17	
020	M	001-PO-NUMBER		AN	12	9	PO NUMBER (SORT KEY)		03800	N1	00	040	IDENTIFICATION CODE	AN	1	22
040	M	001-PO-DATE		MM	6	24	PO DATE		00100	BEG	00	030	PURCHASE ORDER NUMBER	AN	1	22
050	M	001-BILL-NAME		AN	25	30	BILL TO NAME		03100	N1	00	020	NAME	AN	1	35
060	O	001-BILL-ADDR		AN	25	55	BILL TO ADDRESS		03300	N3	00	010	ADDRESS INFORMATION	AN	1	35
070	O	001-BILL-CITY		AN	25	80	BILL TO CITY		03400	N4	00	010	CITY NAME	AN	2	30
080	O	001-BILL-STATE		AN	2	105	BILL TO STATE		03400	N4	00	020	STATE OR PROVINCE CODE	AN	2	2
090	O	001-BILL-ZIP		AN	5	107	BILL TO ZIP		03400	N4	00	031	STANDARD ZIP CODE	AN	5	5
100	M	001-VEND-NAME		AN	25	112	VENDOR NAME		03800	N1	00	020	NAME	AN	1	35
110	O	001-VEND-ADDR		AN	25	137	VENDOR ADDRESS		03805	N3	00	010	ADDRESS INFORMATION	AN	1	35
120	O	001-VEND-CITY		AN	25	162	VENDOR CITY		03810	N4	00	010	CITY NAME	AN	2	30
130	O	001-VEND-STATE		AN	2	187	VENDOR STATE		03810	N4	00	020	STATE OR PROVINCE CODE	AN	2	2
140	O	001-VEND-ZIP		AN	5	189	VENDOR ZIP		03810	N4	00	031	STANDARD ZIP CODE	AN	5	5
150	O	001-BILL-PHONE		AN	12	194	BILL TO PHONE NUMBER		03700	PER	00	040	COMMUNICATION NUMBER	AN	1	25
160	O	001-VEND-PHONE		AN	12	206	VENDOR PHONE NUMBER		03820	PER	00	040	COMMUNICATION NUMBER	AN	1	25
170	O	001-SHIP-DATE		MM	6	218	REQUESTED SHIP DATE		01500	DTM	00	020	DATE	DT	6	6
180	O	001-BILL-ZIP-XT		AN	4	224	BILL TO ZIP EXTENSION		03400	N4	00	032	ZIP CODE EXTENSION	AN	4	4
190	O	001-VEND-ZIP-XT		AN	4	228	VENDOR ZIP EXTENSION		03810	N4	00	032	ZIP CODE EXTENSION	AN	4	4
020	M	002	H	AN	60	24	PO COMMENTS/INSTRUCTIONS		00200	NTE	00	020	FREE FORM MESSAGE	AN	1	60
030	M	005	D	0300		1	PO DETAIL RECORD							R	1	9
040	M	005-QUANTITY		S2	7	24	LINE ITEM QUANTITY		04400	P01	00	020	QUANTITY ORDERED	ID	2	2
050	M	005-UNT-OF-WEAS		AN	4	31	QUANTITY UNIT OF MEASURE		04400	P01	00	030	UNIT OR BASIS FOR MEASUREMENT	ID	2	2
060	M	005-PRICE		S2	7	35	UNIT PRICE		04400	P01	00	040	UNIT PRICE	R	1	14
070	M	005-ITEM-NUMBER		AN	15	42	INTERNAL ITEM IDENTIFIER		04400	P01	00	070	PRODUCT/SERVICE ID	AN	1	30
080	O	005-ITEM-DESC		AN	50	57	OPTIONAL ITEM DESCRIPTION		05000	PID	00	050	DESCRIPTION	AN	1	80

Figure 4.14 Mapping Report – Application Sequence

*Field Definitions***Translation ID:**

This field identifies the transaction map name.

**Send or Receive:**

This field identifies whether the transaction map is used for inbound (R) or outbound (S) translations.

**Name:**

This field displays the user-defined name.

**Date:**

This field displays the date that the report was printed. The format is MM/DD/YYYY.

**Standard Version:**

This field displays the standards version number.

**Transaction Set:**

This field identifies a specific document within a version (i.e., 850 identifies a purchase order transaction set).

**Use Code:**

This field identifies whether the transaction map definition may be used for general translation or only for translating data for a particular group of trading partners. Valid values are:

G	–	General (may be used with any trading partner)
P	–	Partner-specific (must be used for specific trading partners)

**Envelope Type:**

This field identifies the data envelope type used with this transaction map.

Valid values are:

E	–	Edifact
X	–	X12
U	–	UCS
G	–	GS
A	–	ANA TRADACOMS
D	–	User-defined (Only if Gentran:Structure is installed)

**Time:**

This field displays the time the report was printed. The format is HH:MM:SS.

**Envelope Format: (Gentran:Structure only)**

This field identifies whether non-user envelopes (e.g., ISA) should be written as variable or fixed segments. Fixed segments contain fixed length elements with no delimiters or terminators.

F	–	Fixed segments
V	–	Variable segments

**Segment ID Start: (Gentran:Structure only)**

This field identifies the position in the segment where the Segment ID begins.

**Segment ID Length: (Gentran:Structure only)**

This field identifies the length of the Segment ID field.

**Application ID:**

This field identifies the internal application file that the specified transaction map is set to convert from or to.

**Send or Receive:**

This field identifies whether the application file is a send file (outbound) or a receiving file (inbound).

**Name:**

This field identifies the application file.

**RSGRSG Level:**

This field identifies the RSGRSG level format generated by program EBDI056 (the Generate Control Segments subroutine). RSGRSG is a TRADACOMS reconciliation message that can be included as the last message before the END trailer. It confirms that a second transmission was not embedded within an interchange.

**Trading Partner Fields:**

This field identifies the field name in the application file that contains the trading partner identification. This field is added by updating the Application Partner Reference screen.

**Qual:**

This field identifies the field name in the application file that contains the Partner/Qualifier, if applicable. This field is added by updating the Application Partner Reference screen.

**Internal Fields:**

This field identifies the field name in the application file that contains your internal identification. This field is added by updating the Application Partner Reference screen.

**Qual:**

This field identifies the field name in the application file that contains your internal Partner/Qualifier, if applicable. This field is added by updating the Application Partner Reference screen.

**Application Record Line****Seq:**

This field identifies the application record sequence number.

**MC:**

This field identifies whether or not the record is mandatory (M) or optional (O).

**Type:**

This field identifies the type of record, as it exists on your application file.

**H/D:**

This field displays a code that identifies whether the record is a header record (H), detail record (D), or a summary (S) record.

**Loop:**

This field display the loop identification, if applicable.

**Max U:**

This field identifies the maximum number of records of this type that may occur within a data structure (i.e., the header area, summary area, or an application loop).

**Description:**

This field provides descriptive information about the application record.

**Write Flag:**

This field identifies whether or not that record is to be written. It displays which records are being used as dummy mapping records. Valid values are (inbound maps only):

YES

NO

**Application Field Line (illustrated below the application record line)****Seq:**

This field identifies the application field sequence number.

**(MC):**

This field identifies whether or not the field is mandatory (M) or optional (O).

**Element:**

This field identifies the application field name as it exists in your application files.

**Typ:**

This field identifies the field's application data type. Valid values are any of the application data types found in the Application Data Type list.

See the Glossary for a complete listing of Application Data Types.

**Len:**

This field identifies the total length of the field. One field may contain up to 80 characters.

**Start:**

This field identifies the first (starting) position of the specified field as it exists in the application record.

**(Description):**

This field defines the application field description.

**User Exit:**

This field defines the user exit for this application field. The value for this field results from the User Exit field on the Application Records screen (jump code 5.1.3).

**Map Segment/Element Line (illustrated directly across from each application field line)****Alt Map:**

This field identifies an alternate map, if it exists for the application to the standard map illustrated.

**Seg Seq:**

This field identifies the standard's segment sequence number.

**Seg/Elem:**

This field identifies the Segment ID followed by the element sequence number (i.e., BEG030; where BEG identifies the Segment ID, and 030 identifies the element sequence number).



**Description:**

This field identifies the segment element description.

**Type:**

This field identifies the EDI data type, as defined by the standard. Valid values are any of the EDI data types found in the EDI Data Types list.

See the Glossary for a complete listing of EDI Data Types.

**Min:**

This field defines the segment element's minimum field length.

**Max:**

This field defines the segment element's maximum field length.

**Msg:**

This field displays a warning flag value of W indicating a possible non-standard application to standards mapping. The mapping process is completed even when a warning flag displays. For example, if you designate an application field as a 20-position field and the standards designate the segment element as having a maximum length of 19 positions, the system completes the map; however, the data is truncated to 19 positions. The warning also alerts you when invalid constant codes have been used.

See the Glossary for a complete listing of Reserved Word Constants.

**Note:** All table and extended mapping operations are listed immediately below the MAP SEGMENT/ELEMENT line.

## Mapping Report – Standard Sequence

**EBDI052**

The Mapping report shown in Standard Sequence illustrates the current mapping status of segments and elements in the EDI Standard to the records and fields in your application files. The report is sorted by segment sequence number, illustrating each element within the segment. All conditions and accumulators are also shown. The Translation ID or standard version from which the translation was created is shown in the report header. The example reports on the following pages show a report for a translation created from standards version 002040. The right half of the report shows the corresponding field or constant in your application files to or from which the standard segment elements are mapped.

See the Glossary for a complete listing of Reserved Word Constants.

The Standard Mapping report is produced by using the Data Mapping Report program – Standard Sequence (EBDI052). This program produces a concise and complete report detailing the mapping from a transaction to an Application file (inbound) or from an Application file to a transaction (outbound), in standard sequence. This report is an important tool not only for development and debugging, but also for internal system and procedural documentation.

The summary file created by EBDI053 is passed to EBDI061. This program prints a cross-reference report of all constants, accumulators, hash totals, tables, and user exits used in the map.

### Files Used

#### *Input*

The following files are used as input:

Filename	Description
SYS001	Parameters File This file contains the parameters that determine what is produced on the report.
EDISCD1	Standard Codes File 1 This file contains the definition of all the standards codes between 1 and 5.
EDISCD2	Standard Codes File 2 This file contains the definition of all the standards codes between 6 and 10.
SYS020	Application Data Definition This file contains the definition of the application data file.
SYS022	Application Field Definition This file contains the definition of the fields comprising the application data file.
SYS0221	Application Field Definition Path
SYS040	Transaction Definition This file contains the definition of the transaction to be mapped.

Filename	Description
SYS041	Transaction Segment Definition This file contains the definition of the segments to be mapped for the specified transaction.
SYS0411	Transaction Segment Definition Path
SYS042	Transaction Element Definition This file contains the definition of the elements to be mapped for the specified transaction.
SYS0421	Transaction Element Definition Path
EDICFG	System Configuration File

### Output

The following files are used as output:

Filename	Description
SYS005	Mapping Report – Standard Sequence This file contains the Mapping report in standard sequence.
SYS010	Summary Report File This file contains all summary information on hash-totals, constants, accumulators, user-exits, and tables.

### Parameter File Layout

The following table describes the parameter file layout.

Position	Field Content	COBOL Picture	Format	Length	Description
001 – 010	Transaction ID	X(10)	AN	10	The name assigned to the definition of the transaction mapped.
011 – 011	Transaction Send/Receive Flag	X(01)	AN	01	Used to specify if the transaction is inbound or outbound. Valid values: <b>S</b> – Outbound transaction <b>R</b> – Inbound transaction.
012 – 014	Report Option	X(03)	AN	03	Used to specify which elements to display on the report. Valid values: <b>ALL</b> – To display all elements on the report. <b>MAP</b> – To display only the elements that were mapped.

---

Position	Field Content	COBOL Picture	Format	Length	Description
015 – 015	Summary Print Switch	X(01)	AN	01	Used to specify whether to produce a summary report. Valid values: <b>Y</b> – Produce a summary report. <b>N</b> – Do not produce summary report.
016 – 080	Filler	X(65)	AN	65	Not used

Standard Sequence Process

Figure 4.15 illustrates the Standard Sequence process.

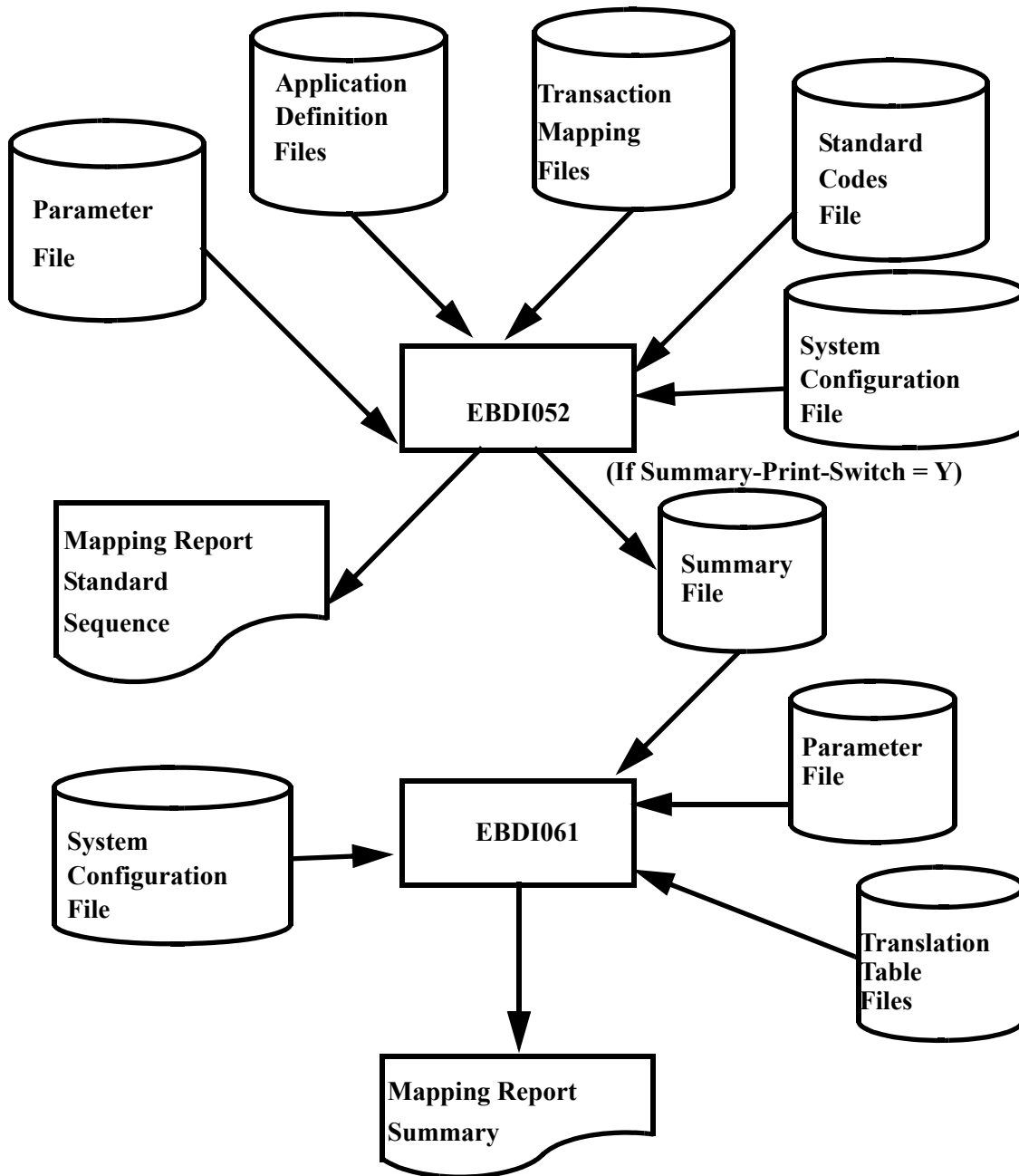


Figure 4.15 Program EBDI052 – Mapping Report – Standard Sequence



*Field Definitions***Translation ID:**

This field identifies the transaction map name.

**Send or Receive:**

This field identifies whether the transaction map is used for inbound (R) or outbound (S) transaction translations.

**Name:**

This field displays the user-defined name.

**Date:**

This field displays the date that the report was printed. The format is MM/DD/YYYY.

**Standard Version:**

This field identifies the standards version number.

**Transaction Set:**

This field identifies a specific document version (e.g., 850 identifies a purchase order transaction set).

**Use Code:**

This field identifies whether the transaction map definition may be used for general translation or only for translating data for a particular group of trading partners.

Valid values are:

G	–	General (may be used with any trading partner)
P	–	Partner-specific (must be used for specific trading partners)

**Envelope Type:**

This field identifies the data envelope type used with this transaction map. Valid values are:

E	–	Edifact
X	–	X12
U	–	UCS
G	–	GS
A	–	ANA TRADACOMS
D	–	User-defined (Only if Gentran:Structure is installed)

**Time:**

This field identifies the time the report was printed. The format is HH:MM:SS.

**Envelope Format: (Gentran:Structure only)**

This field identifies whether non-user envelopes (e.g., ISA) should be written as variable or fixed segments. Fixed segments contain fixed length elements with no delimiters or terminators. Valid values are:

F	–	Fixed segments
V	–	Variable segments

**Segment ID Start: (Gentran:Structure only)**

This field identifies the position in the segment where the Segment ID begins.

**Segment ID Length: (Gentran:Structure only)**

This field identifies the length of the Segment ID field.

**Application ID:**

This field identifies the internal application file that the specified transaction map converts from or to.

**Send or Receive:**

This field identifies whether the application file is a send file (outbound) or a receiving file (inbound).

**Name:**

This field identifies the application file.

**RSGRSG:**

This field identifies the RSGRSG level format generated by program EBDI056 (the Generate Control Segments subroutine). RSGRSG is a TRADACOMS reconciliation message that can be included as the last message before the END trailer. It is used to confirm that a second transmission was not embedded within an interchange.

**Translation Created from Standards Version:**

This field is only displayed if the map was created using a standards version, and the map was created in Gentran:Basic version 5.5 or later. This field identifies the version from which the map was created. If the map was created from a Translation ID, the following field is displayed instead.

**Translation Created from Translation ID:**

This field is only displayed if the map was created using a Translation ID, and the map was created in Gentran:Basic version 5.5 or later. This field identifies the Translation ID from which the map was created. If the map was created from a standards version, the previous field is displayed instead.

**Standards Version:**

This field is only displayed if the map was created using a Translation ID, and the map was created in Gentran:Basic version 5.5 or later. This field identifies the standards version from which the map was created.

**Trading Partner Fields:**

This field identifies the field name in the application file that contains the trading partner identification. This field is added by updating the Application Partner Reference screen.

**Qual:**

This field identifies the field name in the application file that contains the trading partner Qualifier, if applicable. This field is added by updating the Application Partner Reference screen.

**Internal Fields:**

This field identifies the field name in the application file that contains your internal identification. This field is added by updating the Application Partner Reference screen.

**Qual:**

This field identifies the field name in the application file that contains your internal partner qualifier, if applicable. This field is added by updating the Application Partner Reference screen.



**Map Segment Line****Seq:**

This field identifies the standards segment sequence number.

**H/D:**

This field displays a code that identifies whether the segment is a header (H) segment, detail (D) segment, float (F) segment, or a summary (S) segment.

**ID:**

This field identifies the segment identifier (i.e., BEG)

**Ver:**

This field identifies the version of the Segment ID.

**MC:**

This field identifies whether the segment is mandatory (M), conditional (C), or optional (O).

**Max:**

This field defines the maximum number of times that the segment may be repeated in a data structure.

**Loop:**

This field identifies the loop identification, if applicable.

**Max L:**

This field defines the maximum number of times that a specified loop may be repeated within a data structure.

**Write Flag:**

This field identifies whether that record is to be written. It shows what records are being used as summary mapping records. Valid values are (Outbound Maps only):

YES

NO

**Description:**

This field provides descriptive information about the segment.

**Segment Element Line (illustrated below the segment line)****Seq:**

This field identifies the element sequence number.

**(MC):**

This field identifies whether or not the element is mandatory (M), conditional (C), or optional (O).

**Elem:**

This field identifies the element identification, as it exists in the Standards file.

**Ver:**

This field identifies the version of the Element ID.

**Type:**

This field identifies the element's EDI data type. Valid values are any of the EDI data types found in the EDI Data Types list.

See the Glossary for a complete listing of EDI Data Types.

**Min:**

This field defines the element's minimum field length.

**Max:**

This field defines the element's maximum field length.

**(Description):**

This field provides descriptive information about the element.

**Application Field Line (illustrated directly across from each segment element line)****Alt Map:**

This field identifies an alternate map, if it exists.

**Field/Constant:**

This field identifies either a constant or an application field.

See the Glossary for a complete listing of Reserved Word Constants.

**Note:** All table and extended mapping operations are listed immediately below the APPLICATION FIELD LINE.

**Description:**

This field provides descriptive information about the application field.

**Length:**

This field identifies the total length of the application field.

**Type:**

This field identifies the application data type. Valid values are any of the application data types found in the Application Data Types list.

See the Glossary for a complete listing of Application Data Types.

**Msg:**

Displays a warning flag value of W indicating a possible non-standard application to standards mapping. The mapping process is completed even when a warning flag displays. For example, if you designate an application field as a 20 position field and the standards designate the segment element as having a maximum length of 19 positions, the system completes the map; however, the data is truncated to 19 positions. The warning also alerts you when invalid constant codes have been used.

See the Glossary for a complete listing of Reserved Word Constants.

---

## Mapping Report Summary

**EBDI061**

This report executes the option to print a summary section at the end of either the Mapping report – Standard Sequence or the Mapping report – Application Sequence. This report details:

- A cross–reference where all constants, hash–totals, and accumulators are loaded, and when they are moved to the output data.
- A listing of what user exits are used and where they are invoked in the map.
- A listing of all tables used in the map with the option of printing a list of the entries in each table.

To produce a Mapping Report Summary created by the EBDI061 program, the Summary Print Switch must be set to the **Y** value on the parameter record. The switch is on the parameter record that progresses into the EBDI052 or EBDI053 programs, which causes the Mapping Report programs to produce a summary file. This summary file is sorted and then run into the EBDI061 program.

The parameter file that progresses into the EBDI061 program controls the table printing process. If the Table Print Switch is set to the **Y** value, all entries contained in the tables are printed.

**The following condition(s) are checked and an appropriate message is written on the output report for each condition:**

- If the table print switch is turned on, the table is validated to see if it exists. If the table does not exist, the table name, type, and a message that it was not found is printed on the report.
- If the table print switch is turned on and the table exists but does not contain any records, an appropriate message stating that no records exist for this table is printed on the report.
- If the table print switch is turned off, no table validation is done. Only a simple list of the tables used in the map is printed, whether they exist or not.
- Invalid entries in the summary option and table print switch fields are considered garbage and the default value is used. The default is also used if the fields are left blank.

**Processing terminates if any of the following conditions are true:**

- All files were not successfully opened.
- An invalid return code is received on a READ or WRITE statement.

This report is an important tool not only for development and debugging, but also for internal system and procedural documentation.

*Parameter File Layout for Input to EBDI061*

The following table describes the parameter file layout.

<b>Position</b>	<b>Field Content</b>	<b>Cobol Picture</b>	<b>Format</b>	<b>Length</b>	<b>Description</b>
001 – 001	Table Print Switch	X(01)	AN	01	Used to specify how to obtain table information. Valid values: <b>Y</b> – To print a list of the table entries for the summary report. <b>N</b> – To obtain only a list of tables used in the map.

Example Reports

Figure 4.17 and Figure 4.18 illustrate the Mapping report – Summary Section (Cross-reference of Constants, Hash-totals, and Accumulators) and the Mapping report – Summary Section (Table Entries).

```

GENTRAN MAPPING INTEGRATION EBDI061          M A P P I N G   R E P O R T   (SUMMARY SECTION)          PAGE   1
TRANSLATION ID: ANSI3030PO          SEND OR RECEIVE: S          NAME: ANSI 003030 OUTBOUND POS          DATE 12/01/2005
STANDARD VERSION: 003030 /X          TRANSACTION SET: 850          USE CODE: G          ENVELOPE TYPE: X          TIME 18:37:42
APPLICATION ID: POFILE          SEND OR RECEIVE: S          NAME: PURCHASE ORDER MASTER FILE
-----
SEGMENT ID  SEGMENT SEQ  ELEMENT SEQ  MAPPING NO  -----
ADD 1 TO ACCUMULATOR #: 01
POL/00      04400      010          00
          MOVED ACCUMULATOR #: 01
POL/00      04400      010          00
CTT/00      10600      010          00
          ADD TO HASH TOTAL #: 01
POL/00      04400      020          00          ELEMENT = 005-QUANTITY
          MOVED HASH TOTAL #: 01
CTT/00      10600      020          00          ELEMENT = HASH-TOTAL-01
-----
CONDITIONAL STATEMENTS
    
```

**Figure 4.17 Mapping Report – Summary Section (Cross-reference of Constants, Hash-totals, and Accumulators)**



*Field Definitions***Cross-reference of Constants, Hash Totals, and Accumulators****Translation ID:**

This field identifies the transaction map name.

**Send or Receive:**

This field identifies whether the transaction map is used for inbound (R) or outbound (S) transaction translations.

**Name:**

This field displays the user-defined name.

**Date:**

This field displays the date that the report was printed. The format is MM/DD/YYYY.

**Standard Version/Agency:**

This field displays the standards Version ID and the agency code associated with that Version ID number.

**Transaction Set:**

This field identifies a specific document within a version (e.g., 850 identifies a purchase order transaction set).

**Use Code:**

This field identifies whether the transaction map definition may be used for general translation or only for translating data for a particular group of trading partners. Valid values are:

G	–	General (may be used with any trading partner)
P	–	Partner-specific (must be used for specific trading partners)

**Envelope Type:**

This field identifies the data envelope type used with this transaction map. Valid values are:

E	–	Edifact
X	–	X12
U	–	UCS
G	–	GS
A	–	ANA TRADACOMS
D	–	User-defined (Only if Gentran:Structure is installed)

**Time:**

This field displays the time the report was printed. The format is HH:MM:SS.

**Envelope Format: (Gentran:Structure only)**

This field identifies whether non-user envelopes (e.g., ISA) should be written as variable or fixed segments. Fixed segments contain fixed length elements with no delimiters or terminators.

F	–	Fixed segments
V	–	Variable segments

**Segment ID Start: (Gentran:Structure only)**

This field identifies the position in the segment where the Segment ID begins.

**Segment ID Length: (Gentran:Structure only)**

This field identifies the length of the Segment ID field.

**Application ID:**

This field identifies the internal application file that the specified transaction map is set to convert from or to.

**Send or Receive:**

This field identifies whether the application file is outbound (send) or inbound (receive).

**RSGRSG Level:**

This field identifies the RSGRSG level format generated by program EBDI056 (the Generate Control Segments subroutine). RSGRSG is a TRADACOMS reconciliation message that can be included as the last message before the END trailer. It is used to confirm that a second transmission was not embedded within an interchange.

**Name:**

This field identifies the application file.

**Segment ID:**

This field displays the segment identification code, as it is defined on the EDI Standards table.

**Segment Seq:**

This field displays the segment's sequence number. The segment sequence number determines the order in which the segment is listed in the EDI table.

**Element Seq:**

This displays the data element's sequence number. The data element sequence number determines the order in which the element is listed in the EDI Standard table.

**Mapping No:**

This field displays the mapping number for the data element mapping definition.

**Conditional Statements:**

This field displays the arguments of a logical expression.

**Add 1 to Accumulator #:**

These are used as a cross-reference listing to show where and how in the map accumulators were used.

**Save in Constant #:**

These are used as a cross-reference listing to show where and how in the map constants were used.

**Add to Hash-Total #:**

These are used as a cross-reference listing to show where and how in the map hash-total were used.

See the Glossary for a complete listing of these Reserved Word Constants.

**Table Entries****Translation Table ID:**

This field identifies the translation table name.



**Type:**

This field identifies the type of translation table. Valid values are:

C	–	Code
D	–	Data
V	–	Validation

**Desc:**

This field identifies the user-defined name for the table.

**Standard Code:**

This field indicates the EDI Standard code value.

**User Code:**

This field displays the user-defined code value for the specified Standard code.

**Description:**

This field provides descriptive information about the Standard code.

## Mapping Validation Report

**EBDI082**

The Mapping Validation report lists errors detected by the system after it examines how the elements are mapped within the transaction for an outbound map (this report is valid for outbound transactions only). The top portion of the report shows general translation, standards, and application information. The details of the report include the Segment ID, its sequence number, the segment type, the specific element, element sequence number, and element description of the element within the segment that has caused the error. In addition, this reports contains essential information that defines the mapping error relationship.

The Mapping Validation report is produced using the Mapping Validation Report (EBDI082) program. The transaction to be validated is specified in the parameter record. Only one transaction may be specified for each execution of the program. A header record, segments, and elements must already exist for the transaction to produce a meaningful report.

### Files Used

#### *Input*

The following files are used as input:

Filename	Description
SYS001	Parameter File This file contains the Transaction ID and Send/Receive Code for the transaction to be validated.
EDISTRN	Standards Transaction File This file contains all the transactions defined for a standard.
EDISSEG	Standards Segment File This file contains all the segments defined for a standard.
EDISSGD	Standards Segment Description File This file contains all the descriptions standard segments.
EDISCD1	Standards Code File 1 This file contains all the codes defined for a standard between 1 and 5.
EDISCD2	Standards Code File 2 This file contains all the codes defined for a standard between 6 and 10.
SYS040	Transaction Header File This file contains all the header records for transactions which have been identified to the system.
SYS041	Transaction Segment File This file contains all the segment records for transactions which have been identified to the system.
SYS042	Transaction Element File This file contains all the element records for transactions which have been identified to the system.
EDICFG	System Configuration File

*Output*

The following files are used as output:.

Filename	Description
SYS005	Map Validation report This is a spool file for the output of this program.

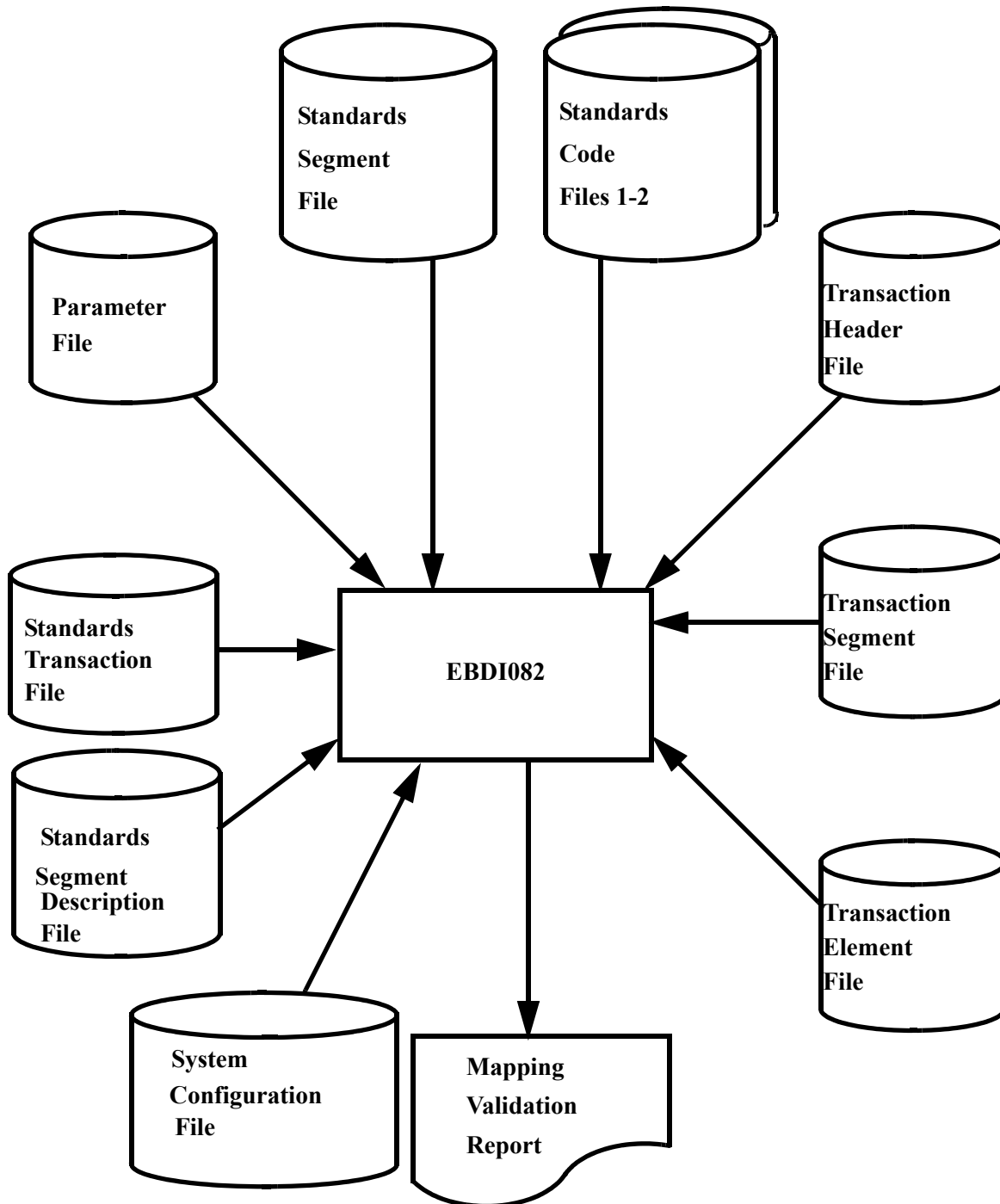
*Parameter File Layout*

The following table describes the parameter file layout.

Position	Field Content	COBOL Picture	Format	Length	Description
001 – 010	Transaction ID	X(10)	AN	10	The name assigned to the definition of the transaction mapped.
011 – 011	Transaction Send/Receive Flag	X(01)	AN	01	The Mapping Validation report is valid only for outbound transactions. Valid value: Must be S
012 – 080	Filler	X(69)	AN	69	Not used

*Mapping Validation Report Process*

Figure 4.19 illustrates the Mapping Validation Report process.



**Figure 4.19 Program EBD082 –Mapping Validation Report**

Example Report

Figure 4.20 illustrates the Mapping Validation report.

```

GENTRAN MAPPING INTEGRATION          M A P P I N G   V A L I D A T I O N   R E P O R T          PAGE      1
TRANSLATION ID : BADI3030PO          SEND OR RECEIVE: S          NAME: ANSI 003030 OUTBOUND POS          DATE 12/01/2005
STANDARD VERSION: 003030            X          TRANSACTION SET: 850          USE CODE: G          ENVELOPE TYPE: X          TIME 12:00:00
APPLICATION ID : POFILLE
SEG VER H/D SEG SEQ ELEM VER ELE SEQ DESCRIPTION          MAPPING ERROR/RELATIONSHIP
CTT 00 S          TRANSACTION TOTALS          MANDATORY SEGMENT NOT MAPPED
```

Figure 4.20 Mapping Validation Report

*Field Definitions***Translation ID:**

This field identifies the transaction map name.

**Send Or Receive:**

This field identifies whether the transaction map is used for inbound (R) or outbound (S) transaction translations. This field always contains a value of S.

**Name:**

This field contains the user-defined transaction name.

**Date:**

This field identifies the date that the report was printed. The format is MM/DD/YYYY.

**Standard Version:**

This field identifies the standards version number.

**Transaction Set:**

This field identifies a transaction type (i.e., 850 identifies a purchase order transaction set).

**Use Code:**

This field identifies whether the transaction map definition may be used for general transaction purposes or only for translating data for a particular trading partner profile. Valid values are:

G	–	General (may be used with any trading partner)
P	–	Partner-specific (must be used for a specified trading partner)

**Envelope Type:**

This field identifies the data envelope type used with this transaction map. Valid values are:

E	–	EDIFACT
X	–	X12
U	–	UCS
G	–	GS
A	–	ANA TRADACOMS
D	–	User-defined (Only if Gentran:Structure is installed)

**Time:**

This field identifies the time the report was printed. The format is HH:MM:SS.

**Envelope Format:**

This field identifies whether non-user envelopes (e.g., ISA) should be written as variable or fixed segments. Fixed segments contain fixed length elements with no delimiters or terminators. This field is used only if Gentran:Structure is installed.

F	–	Fixed segments
V	–	Variable segments

**Segment ID Start:**

This field identifies the position in the segment where the Segment ID begins. This field is used only if Gentran:Structure is installed.

**Segment ID Length:**

This field identifies the length of the Segment ID field. This field is used only if Gentran:Structure is installed.

**Application ID:**

This field identifies the internal application file that the specified transaction map is set to convert.

**Seg ID:**

This field identifies the Segment ID that caused the error condition.

**Ver:**

This field identifies the version of the Segment ID.

**H/D:**

This field displays a code that identifies whether the segment is a header record (H), detail record (D), or a summary (S) record.

**Seg Seq:**

This field identifies the sequence number of the segment that caused the error condition.

**Elem:**

This field identifies the reference number of the element that caused the error condition.

**Ver:**

This field identifies the version of the Element ID.

**Ele Seq:**

This field identifies the sequence number of the element that caused the error condition.

**Description:**

This field provides descriptive information about the element.

**Mapping Error/Relationship:**

This field provides descriptive information about the error condition.

## Partner Change Audit Inquiry Report Program

**EBDI092**

The Partner Change Audit Inquiry Report program (EBDI092) enables you to produce reports for Partner maintenance updates that have been applied to the online Partners files (Partner, Inbound Control, Outbound control, and Xref/Partrel). This program produces either a summary level report of activity or a detail report that shows all update activity, including before and after images of any "edit" updates.

The Partner change audit file contains records that track changes made to the Partners files. Add, delete and update functions via the online Partner subsystem result in records being written to the Partner Change Audit file. The Batch copy utility (EBDI087) and the online copy utility (EDIR087) will also write records to the Partner change audit file indicating add, delete and replace functions. The online Upload process (EDIX235), using component EDIR087, will also write a change audit record to indicate an add or replace event.

### Operation

The Partners Change Audit Inquiry program runs as a batch job step.

### Files Used

#### *Input*

The following files are used as input to the Partner Change Audit Inquiry Report program:

Filename	Description
EDIPCHA	Partner Change Audit File
EDICNTL	Control Card File
ERRCTL	Error Message and Control File
EDICFG	System Configuration File

#### *Reports*

Program EBDI092 produces the following reports:

Filename	Description
EDIRPT	Partner Change Audit Inquiry Report
EDILOG	Partner Change Audit Processing Log
EDISUM	Partner Change Audit Summary Report



## Control Card Options

The following Partners Change Audit Inquiry operations and selection criteria are specified by control cards.

### Operations

SELECT (Default Operation)

### Selection Criteria

#### Defaults

FROM-UPDATE-DATE	01/01/1900
FROM-UPDATE-TIME	00:00
TO-UPDATE-DATE	12/31/2099
TO-UPDATE-TIME	23:59
USER	ALL
REPORT -TYPE	SUMMARY
RECORD-TYPE	P
PARTNER-ID	ALL
PARTNER-QUAL	ALL
USER-PARTNER-ID	ALL
XREF-ID	ALL
XREF-QUAL	ALL
USER-ID	ALL
USER-QUAL	ALL

### Valid Criteria Values

#### From-Update-Date

Specifies the date that the update was applied to the Partner. Starts the range. Used with To-Update-Date.

MM/DD/YYYY

#### From-Update-Time

Specifies the time that the update was applied to the Partner. Starts the range. Used with To-Update-Time.

HH:MM

#### To-Update-Date

Specifies the date that the update was applied to the Partner. Ends the range. Used with From-Update-Date.

MM/DD/YYYY

#### To-Update-Time

Specifies the time that the update was applied to the Partner. Ends the range. Used with From-Update-Time.

HH:MM

**User**

Specifies the ID of online user who applied the change to the Partner records.

ALL

XXX

**Report-Type**

Identifies the level at which the Inquiry should report.

FULL Includes detailed information for each update.

SUMMARY Includes summary list of updates.

**Record Type**

Used to restrict the type of change audit records that will be used to create the inquiry report. The Partner change audit file contains records for partner and Xref, or Partrel files.

P – Partner

X – Xref (only valid for Partner Qualifier mode)

R – Partrel (only valid for Relationship mode)

**Partner-Id**

Contains the 35-position Partner ID.

**Note:** Only valid when using Record Type = Partner and Trading Profile Mode is Partner Qualifier/Mixed **OR** when using Record Type = Partrel and trading profile mode is Relationship.

When used with **Report-Type = Partner** and when processing in Partner Qualifier Mode, this field restricts which partner change audit records are used to create the inquiry report.

When used with **Record-Type=Partrel** and when processing in Relationship Mode, this field is used with User Id to restrict which Partner change audit records are used to create the inquiry report.

ALL All Partner records will be used to create the inquiry report.

XX

**Partner-Qual**

Contains the 4-position qualifier associated with the Partner ID.

**Note:** Only valid when using Record Type = Partner and Trading Profile Mode is Partner Qualifier/Mixed **OR** when using Record Type = Partrel and trading profile mode is Relationship.

ALL All records that satisfy the **Partner ID** control card are used to create the inquiry report regardless of the Partner-Qual value.

XXX Only change audit records containing the Partner-Qual that matches the specified Partner-Qual are used to create the inquiry report.

**User-Partner-Id**

Used with **Record Type = P** and trading profile mode Relationship only. This field contains the 15-position user ID and the 15-position partner ID that combined define the key to the Relationship partner records.

ALL All partner records are used to create the inquiry report.

XXXXXXXXXXXXXXXXXX/XXXXXXXXXXXXXXXXXX

The specified User-Partner ID is used to restrict which User-Partner records are used to create the inquiry report.

**Xref-Id**

This field contains the 35-position Xref ID that defines the key to the Xref file.

**Note:** Only valid when using **Record Type = Xref** and Trading Profile Mode is Partner Qualifier/Mixed.

ALL All Xref change audit records are used to create the inquiry report.

XX

The specified Xref ID is used to restrict which Xref change audit records are used to create the inquiry report.

**Xref-Qual**

Contains the 4-position qualifier associated with the Xref ID.

**Note:** Only valid when using **Record Type = Partner** and Trading Profile Mode is Partner Qualifier/Mixed.

ALL All records that satisfy the **Xref ID** control card are used to create the inquiry report regardless of the Xref-Qual value.

XX Only change audit records containing the Xref-Qual that match the specified Xref-Qual are used to create the inquiry report.

**User-Id**

When combined with Partner ID, this field defines the key to the Partrel file.

**Note:** Only valid when using **Record Type = Partrel** and trading profile mode is Relationship.

ALL All Partrel records are used to create the inquiry report.

XXXXXXXXXXXXXXXXXX/XXXXXXXXXXXXXXXXXX

The specified User ID is used to restrict which Partrel change audit records are used to create the inquiry report.

**User-Qual**

Contains the 4-position qualifier associated with the User ID.

ALL All records that satisfy the **User ID** control card are used to create the inquiry report regardless of the User-Qual value.

XXXX The specified User Qual is used to restrict which change audit records are used to create the inquiry report.

## Partner File Inquiry Report

**EBDI008**

The Partner File Inquiry program (EBDI008) allows you to run an inquiry report against the online partner file using selected criteria. The output is a report that lists the partner profile, group, and transaction information for those partner file records that match your selection criteria. For example, you could inquire against your partner file to identify all partners that have standards version 004030 coded as the compliance version on the group records. This can be helpful when migrating to a new version. Many selection criteria parameters are available for this program (see EDICNTL—Partner Inquiry Report Processing Parameters on page 7 - 77).

### Files Used

#### *Input*

The following files are used as input to the Partner File Inquiry Report program:

Filename	Description
SYS090	6.4 VSAM Partner File
EDICNTL	Parameter File
ERRCTL	Error Message and Control File
EDICFG	System Configuration File

#### *Reports*

Program EBDI008 produces the following reports:

Filename	Description
EDIRPT	Partner Inquiry Report
EDILOG	Processing Log
EDISUM	Summary Report

---

## EDICNTL—Partner Inquiry Report Processing Parameters

The following describe the Partner Inquiry Report processing parameters.

**Note:** KEYWORD should start in Column 3 of EDICNTL cards.

INQUIRY VALUE should start in Column 36 of EDICNTL cards. It must be the exact value that is found on the file. Wildcard values are not supported.

### Compliance-Version-Id

This parameter will generate a list of all the partner IDs that contain the indicated 12-character version ID as found in the **Compliance Version** field on the Group Information screen (EDIM030).

### Division

This parameter will generate a list of all the partner IDs that contain the indicated 3-character division as found in the **Division** field on the Header Information screen (EDIM026).

### Group-Envelope-Version-ID

This parameter will generate a list of all the partner IDs that contain the indicated 12-character version ID as found in the **Version (GS08)** field on the Group Information screen (EDIM033 for GS envelopes) or the **Version Number, Release, and Association (UNG07)** fields on Group information screen (EDIM032 for UNG envelopes).

### Group-ID

This parameter will generate a list of all the partner IDs that contain the indicated 6-character group ID as found in the **Group ID field** on the Group Information Screen (EDIM030) or the **Functional Group ID** field on the Transaction Information screen (EDIM040).

### Group-Receiver-ID

This parameter will generate a list of all the partner IDs that contain the indicated receiver ID as found in the **Application Receivers Code (GS03)** field on the Group Information screen (EDIM033 for GS envelopes) or the **Recipients ID** field on the Group Information screen (EDIM032 for UNG envelopes). The ID can be up to:

- 15 characters for GS envelopes
- 35 characters for UNG envelopes

### Group-Sender-ID

This parameter will generate a list of all the partner IDs that contain the indicated 35-character sender ID as found in the **Application Senders Code (GS03)** field on the Group Information screen (EDIM033 for GS envelopes) or the **Sender ID** field on the Group Information screen (EDIM032 for UNG envelopes).

### Interchange-Header-Option

This parameter will generate a list of all the partner IDs that contain the indicated 3-character version ID as found in the **Interchange Header Option** field on the Control Information screen (EDIM015). Valid values:

ISA  
ICS

BG  
GS  
UNA  
UNB  
STX  
SCH  
GEN

**Interchange-Receiver**

This parameter will generate a list of all the partner IDs that contain the indicated receiver ID as found in the **Receiver ID (ISA08)** field on the Control Information screen (EDIM016 for ISA envelopes), the **Application Receiver Code (BG04)** field on the Control Information screen (EDIM017 for BG envelopes), the **Receiver ID (ICS08)** field on the Control Information screen (EDIM018 for ICS envelopes), the **Recipient ID Name (STX03)** field on the Control Information screen (EDIM019 for STX envelopes), or the **Recipient ID (UNB03)** field on the Control Information screen (EDIM021 for UNB envelopes). The ID can be up to:

- 15 characters for ISA and ICS envelopes
- 35 characters for STX and UNB envelopes
- 12 characters for BG envelopes

**Interchange-Sender**

This parameter will generate a list of all the partner IDs that contain the indicated sender ID as found in the **Sender ID (ISA06)** field on the Control Information screen (EDIM016 for ISA envelopes), the **Application Sender Code (BG03)** field on the Control Information screen (EDIM017 (or BG envelopes), the **Sender ID (ICS06)** field on the Control Information screen (EDIM018 for ICS envelopes), the **Sender ID Name (STX02)** field on the Control Information screen (EDIM019 for STX envelopes), or the **Sender ID (UNB02)** field on the Control Information screen (EDIM021 for UNB envelopes). The ID can be up to:

- 15 characters for ISA and ICS envelopes.
- 12 characters for BG envelopes
- 15 characters for ISA and ICS envelopes.
- 35 characters for STX and UNB envelopes.

**Interchange-Version**

This parameter will generate a list of all the partner IDs that contain the indicated 5-character version ID as found in the **Version (ISA12)** field on the Control Information screen (EDIM016 for ISA envelopes) or the **Version (ICS04)** field on the Control Information screen (EDIM018 for ICS envelopes).

**Mailbox-ID**

This parameter will generate a list of all the partner IDs that contain the indicated 8-character mailbox ID as found in the **Mailbox/Remote ID (for Plus)** field on the Control Information screen (EDIM015).

**Multiple-Envelope-ID**

This parameter will generate a list of all the partner IDs that contain the indicated 3-character envelope ID as found in the **Multiple Envelope ID** field on the Control Information screen (EDIM015), Group Information screen (EDIM030), or Transaction Information screen (EDIM040). Valid values:

ISA  
ICS  
BG  
GS  
UNA  
UNB  
STX  
SCH  
GEN

**Network-ID**

This parameter will generate a list of all the partner IDs that contain the indicated 15-character network ID as found in the **Network ID (for Plus)** field on the Control Information screen (EDIM015).

**Partner-Version**

This parameter will generate a list of all the partner IDs that contain the indicated 12-character version ID as found in the **Version** field on the Control Information screen (EDIM015), Group Information screen (EDIM030), or Transaction Information screen (EDIM040).

**Reject-Error-Number**

This parameter will generate a list of all the partner IDs that contain the indicated 5-digit error number as found in the **Error Number** field on the Error Rejection screen (EDIM055). Valid values:

00000 – 99999

**Split-File-Number**

This parameter will generate a list of all the partner IDs that contain the indicated 4-digit file number as found in the **Output File Identifier** field on the Data Separation screen (EDIM050). Valid values:

0001 – 0099

**Test-Production-IND**

This parameter will generate a list of all the partner IDs that contain the indicated 1-character production ID as found in the **Test or Production (ISA15)** field on the Control Information screen (EDIM016). Valid values:

T	Test
P	Production

**Trans-Envelope-Version-ID**

For UNH envelopes only. This parameter will generate a list of all the partner IDs that contain the indicated 12-character version ID as found in the **Version Number, Version Release Number, Version Association Code (UNH02)** fields on the Transaction Information screen (EDIM042).

**Transaction-ID**

This parameter will generate a list of all the partner IDs that contain the indicated 6-character transaction ID as found in the **Transaction Id** field on the Transaction Information screen (EDIM040).

**Translation-Map-ID-Inbound**

This parameter will generate a list of all the partner IDs that contain the indicated 10-character map ID as found in the **Translation-Map-ID-Inbound** field on the Transaction information screen (EDIM040).

**Translation-Map-ID-Outbound**

This parameter will generate a list of all the partner IDs that contain the indicated 10-character map ID as found in the **Outbound** field on the Transaction information screen (EDIM040).

**User-Record-ID**

This parameter will generate a list of all the partner IDs that contain the indicated 4-character record ID as found in the **Record ID** field of the User Defined screen (EDIM045).



Example Reports

Figure 4.21 is an example of the Partner File Inquiry report.

```

PAGE : 00001
PROGRAM: EBDI008
VERSION: 6.4

REPORT DATE: 12/01/2005
REPORT TIME: 12:00:00

GENTRAN: BASIC
PARTNER FILE INQUIRY

=====+
| COMPLIANCE-VERSION-ID: 004030 |
=====+

PARTNER ID          QUAL      MULTIPLE  VERSION  TRANS  TRANS  GROUP  TRANS  TRANS
-----+-----+-----+-----+-----+-----+-----+-----+-----+
ISA-PARTNER        |||DFT
VENDOR-1           |||DFT
VENDOR-2           |||DFT
VENDOR-3           |||DFT
VENDOR-4           |||DFT
=====+
| COMPLIANCE-VERSION-ID: D 99B |
=====+

PARTNER ID          QUAL      MULTIPLE  VERSION  TRANS  TRANS  GROUP  TRANS  TRANS
-----+-----+-----+-----+-----+-----+-----+
VENDOR-5           |||DFT
CONTROL
ORDERS
    
```

Figure 4.21 Partner File Inquiry Report

## Partner Profile Print Report

**EBDI006**

Gentran:Basic enables you to print a hard-copy report displaying the contents of the partner profile for one or more trading partners.

### *Partner Profile Print Processing*

Generating the Partner Profile Print report is a three-step process:

1. The EBDI006A program selects partner records from the partner profile, labeled with a record identifier, and places them into an output file.
2. The partner records are sorted by the Partner ID, Partner ID/Qualifier, and record identifier when processing in the Partner/Qualifier mode. The partner records are sorted by the User/Partner ID and record identifier when processing in the Relationship mode.
3. The sorted file is passed to the EBDI006B program, which generates the Partner Profile Print report.

The report is designed to duplicate the format of the online Partner Maintenance Menu screens for easier readability.

See Chapter 9 for information on the screen layouts and field definitions.



Filename	Description
SYS091	Partner Outbound Control File – Input This file contains the outbound control records for the partner profile.
SYS090	Partner Profile – Input The partner profile is input to the first program (EDBI006A). All partner profile records are used. No records are updated.
SYS050	Output File (556 character Output) The Output file contains the partner records selected from the partner profile.
SYS095	Partner Cross Reference File This file contains the partner X-Ref records and is used only when processing the Partner/Qualifier mode.
EDIPREL	Partner Relationship File This file is used when processing in the Relationship mode that cross-references EDI IDs to User/Partner IDs.
EDICFG	System Configuration File

**Sort Step**

Filename	Description				
SORTIN	Input File (556 character Input) This is the output file generated from program EDBI006A. Sort criteria: <table style="margin-left: 100px; border: none;"> <tr> <td style="padding-right: 40px;">001–039</td> <td>Partner Profile ID</td> </tr> <tr> <td>555–556</td> <td>Records Identifier</td> </tr> </table>	001–039	Partner Profile ID	555–556	Records Identifier
001–039	Partner Profile ID				
555–556	Records Identifier				
SORTOUT	Output File (556 character Output) This is the sorted output file that is passed on to program EDBI006B.				

**Program EBDI006B**

Filename	Description
ERRCTL	Error Message and Control File This file is used as input. The Error Message file is read when an error reject record is found on the input file. The error message is written beside the error number on the Error Reject record.
SYS010	Partner Profile Print Report (133 character report) This is the output report, printed in the format of the online Partner Maintenance Menu screens.
SYS050	Input File (556 character input) This file contains the sorted extraction partner record from EBDI006A.

*Partner Print Processing Parameters*

The following tables describe partner print processing parameters.

**Parameter Record 1**

Position	Field Content	COBOL	Format	Length	Description
1 – 6	PRINTb	X(06)	AN	6	Constant PRINTb
7 – 41	Beginning Partner ID	X(35)	AN	35	Partner identification of key beginning search through the partner profile. Valid value: <b>ALL</b> – print all partners.
42 – 42	Separator	X(01)	AN	1	Constant “—”.
43 – 46	Beginning Partner Qualifier	X(04)	AN	4	Partner ID/Qualifier of beginning partner profile search.
47 – 80	Filler	X(34)	AN	34	Not used

**Parameter Record 2**

Position	Field Content	COBOL Picture	Format	Length	Description
1 – 4	bTOb	X(04)	AN	4	Constant bTOb
5 – 39	Ending Partner ID	X(35)	AN	35	Partner identification of key ending search through the partner profile.
40 – 40	Separator	X(01)	AN	1	Constant “—”
41 – 44	Ending Partner Qualifier	X(04)	AN	4	Partner ID qualifier of key ending partner profile search.
45 – 80	Filler	X(36)	AN	36	Not used

**Parameter Record 3**

Position	Field Content	COBOL Picture	Format	Length	Description
1 – 6	PRINTb	X(06)	AN	06	Constant PRINTb.
7 – 11	USERb	X(05)	AN	05	Constant USERb
12 – 26	Beginning User ID	X(15)	AN	15	User ID of key beginning search through the Relationship – partner profile

Position	Field Content	COBOL Picture	Format	Length	Description
27 – 32	␣PART␣	X(06)	AN	06	Constant ␣PART␣
33 – 47	Beginning Partner ID	X(15)	AN	15	Partner ID of the beginning search through the Relationship – partner profile
48 – 80	Filler	X(33)	AN	33	Not used

**Parameter Record 4**

Position	Field Content	COBOL Picture	Format	Length	Description
1 – 4	␣TO␣	X(04)	AN	04	Constant ␣TO␣
5 – 9	USER␣	X(05)	AN	05	Constant USER␣
12 – 26	Ending User ID	X(15)	AN	15	User ID of key ending search through the Relationship – partner profile
27 – 32	␣PART␣	X(06)	AN	06	Constant ␣PART␣
33 – 47	Ending Partner ID	X(15)	AN	15	Partner ID of the ending search through the Relationship – partner profile
48 – 80	Filler	X(33)	AN	33	Not used

**Parameter Record 5**

Position	Field Content	COBOL Picture	Format	Length	Description
1 – 9	INTERCHG␣	X(09)	AN	9	Constant INTERCHG␣.
10 – 12	Interchange Header Option	X(03)	AN	03	Indicates Interchange header option to be selected (ICS, ISA, BG, UNA, UNB, GS, GEN, STX, SCH)
13 – 80	Filler	X(68)	AN	68	Not used

## Parameter Record 6

Position	Field Content	COBOL Picture	Format	Length	Description
1 – 8	VERSIONb	X(08)	AN	8	Constant VERSIONb
9 – 20	Interchange Version	X(12)	AN	12	Indicate Interchange version to be selected (U2/6, C2/5)
21 – 80	Filler	X(60)	AN	60	Not used

Partner Profile Print File Process

Figure 4.22 illustrates the Partner Profile Print report.

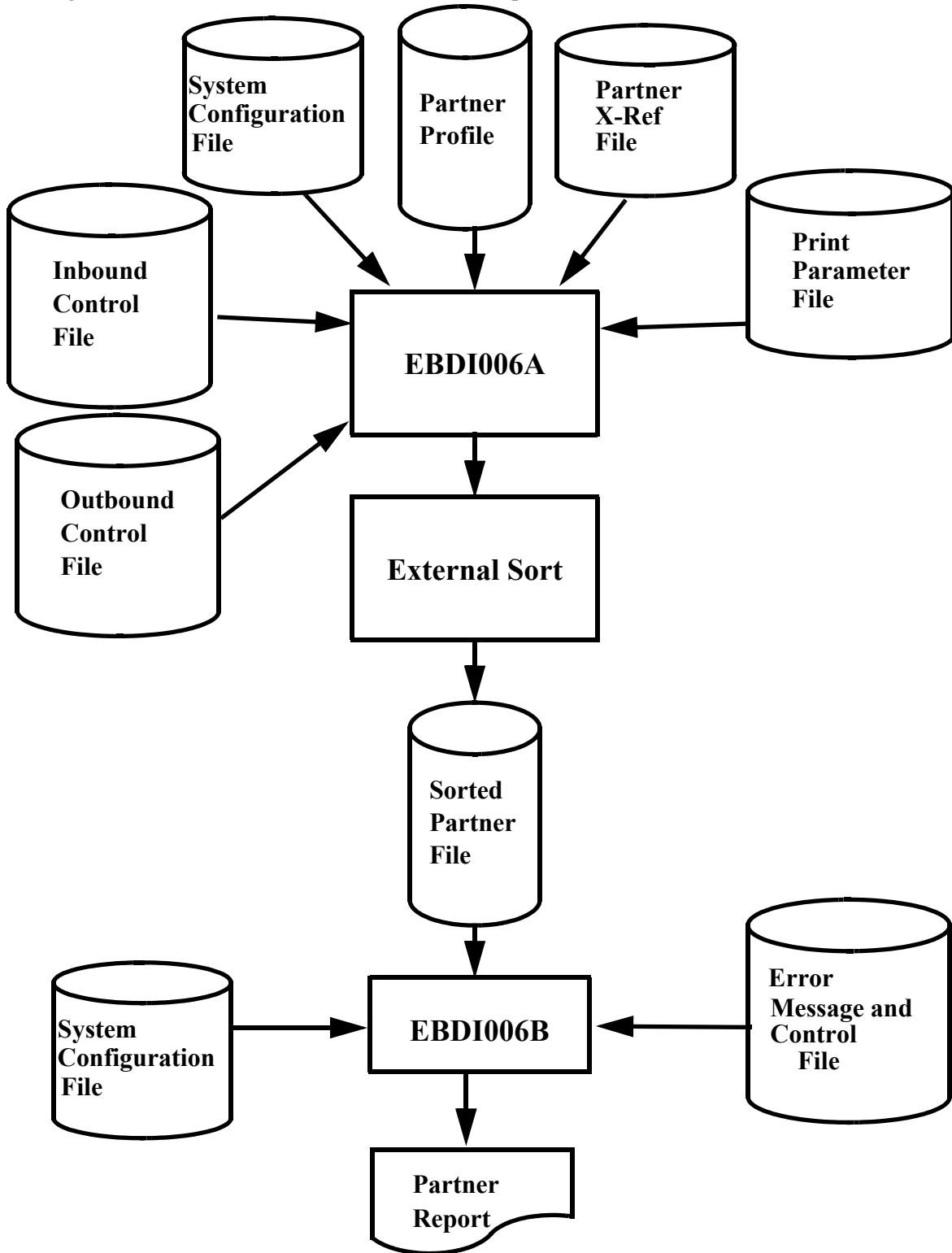


Figure 4.22 Program EBDI006 – Partner Profile Print Report



Example Reports

Figure 4.23 through Figure 4.26 are examples of the Partner Profile Print report.

```

EBDI006B  RUN DATE 12/01/2005  FORMATTED PARTNER FILE REPORT  PAGE 1
PARTNER ID: !!!GENTRAN-RESERVED-PARTNER-ID-1  QUAL:  RUN TIME 12:00:00

      P A R T N E R  H E A D E R  I N F O R M A T I O N
      *****
DESCRIPTION:  GENTRAN RESERVED PARTNER
DIVISION.....: 000
UPDATE ALLOWED.....: Y
VIEWPOINT - EXCEPTION.....:
VIEWPOINT - TRACKING.....:
EBDI006B  RUN DATE 12/01/2005  FORMATTED PARTNER FILE REPORT  PAGE 2
PARTNER ID: !!!GENTRAN-RESERVED-PARTNER-ID-1  QUAL:  RUN TIME 08:55:51

      P A R T N E R  C O N T R O L  I N F O R M A T I O N
      *****
MULTIPLE ENVELOPE ID.....:
INTERCHANGE HEADER OPTION....: ISA
LAST INCOMING SEQUENCE NUMBER:
EDI DATABANK INBOUND.....:
EXPECT A TAL OR ACL.....:
ACKNOWLEDGE INTERCHANGE.....:
ALTERNATE ACKNOWLEDGE PARTNER:
ALTERNATE PARTNER QUALIFIER..:
LAST INCOMING BG PASSWORD....:
REMOTE ID..(FOR PLUS).....:
NETWORK ID..(FOR PLUS).....:
VIEWPOINT.....:
TRACKING.....:

OUTBOUND ENVELOPE INFORMATION FOR ISA SEGMENT:
AUTHORIZATION QUAL....:ISA01:  AUTHORIZATION.ISA02:
SECURITY CODE QUAL....:ISA03:  SECURITY CODE.ISA04:
SENDER ID QUAL.....:ISA05:  SENDER ID....:ISA06:
RECEIVER ID QUAL.....:ISA07:  RECEIVER ID...:ISA08:
CONTROL STANDARD ID...:ISA11:  USE.....:ISA12:
VERSION.....:ISA12:  ACK REQUESTED.ISA14: 0
CONTROL NUMBER.....:ISA13:
TEST OR PRODUCTION.....:ISA15:  OR HEX
SUBELEMENT SEPARATOR...:ISA16:  OR HEX
ELEMENT SEPARATOR.....:  OR HEX
SEGMENT TERMINATOR.....:
    
```

Figure 4.23 Partner Profile Print Report (Partner Control Information)

PARTNER ID: VENDOR-1

QUAL:

PARTNER GROUP INFORMATION  
\*\*\*\*\*

GROUP ID.: !!!DFT VERSION:  
 COMPLIANCE VERSION..... 002040  
 COMPLIANCE VERSION USE..... A  
 ACCEPT FLAG..... Y  
 SEND FLAG..... Y  
 EXPECT AK1 OR B5 ACKNOWLEDGEMENT... Y  
 ACKNOWLEDGE GROUP OR TRANSACTION... T  
 ACKNOWLEDGEMENT OVERDUE AFTER (HRS):  
 ALTERNATE ACKNOWLEDGEMENT PARTNER...  
 ALTERNATE PARTNER QUALIFIER.....  
 LAST INCOMING CONTROL NUMBER..... 000000000  
 EDI DATABANK INBOUND..... D OUTBOUND....: D  
 VIEWPOINT..... Y

GROUP ID.: !!!DFT VERSION:

OUTBOUND ENVELOPE INFORMATION FOR GS SEGMENT:

FUNCTIONAL GROUP ID..... !!!DFT  
 APPLICATIONS SENDERS CODE..... GS01: STERLING  
 APPLICATIONS RECEIVERS CODE..... GS02: ABC  
 CONTROL NUMBER..... GS03: 000000006  
 RESPONSIBLE AGENCY CODE..... GS06: X  
 VERSION..... GS07:  
 TRANSACTION SEGMENT ID.....  
 TRANSACTION CONTROL NUMBER..... 000000000

**Figure 4.24 Partner Profile Print Report (Partner Group Information)**

EBDI006B RUN DATE 12/01/2005 FORMATTED PARTNER FILE REPORT PAGE 3

PARTNER ID: VENDOR-1 QUAL:

PARTNER TRANSACTION INFORMATION  
\*\*\*\*\*

TRANSACTION ID: 850 VERSION:

FUNCTIONAL GROUP ID.....:	
TEST OR PRODUCTION.....:	T
TRANSLATION MAP ID INBOUND.....:	
EDI DATABANK INBOUND.....:	D
APPLICATION DATABANK INBOUND.....:	F
LAST INCOMING CONTROL NUMBER.....:	
ACCEPT TRANSACTION INBOUND.....:	Y
SEND TRANSACTION OUTBOUND.....:	Y
ACKNOWLEDGE THIS TRANSACTION.....:	Y
EXPECT AK2 OR A2 ACKNOWLEDGEMENT.....:	N
TRANSACTION ACKNOWLEDGEMENT TYPE.....:	
TRANSACTION RELEASE NUMBER.....:	
VIEWPOINT.....:	

TRANSACTION ID: 850 VERSION:

OUTBOUND ENVELOPE INFORMATION FOR ST SEGMENT:  
TRANSACTION SET IDENTIFIER.....ST01: 850  
CONTROL NUMBER.....ST02: 000000006  
VERSION.....:

Figure 4.25 Partner Profile Print Report (Partner Transaction Information)

```

EBDI006B  RUN DATE 12/01/2005          RUN TIME 12:00:00          PAGE 4
          PARTNER ID: VENDOR-1          QUAL:
          PARTNER NAME AND ADDRESS
          *****
          NAME...: TUTORIAL - ABC COMPUTER COMPANY
          ADDRESS:
          :
          :
          :
          CITY...:
          STATE...:
          ZIP...: -          COUNTRY CODE:
          CONTACT: ( ) - EXT.
          INTERNATIONAL DIAL CODE: 000
          *****
          ***** END OF TRADING PARTNER *****
    
```

**Figure 4.26 Partner Profile Print Report (Partner Name and Address)**

*Field Definitions*

See the sub-topic “Field Descriptions” in Chapter 2 of the *Gentran:Basic for zSeries Release 6.4 User’s Guide* for a description of the fields listed on the report.

## Security Change Audit Inquiry

**EBDI073**

The Security Change Audit Inquiry Report program (EBDI073) enables you to produce reports for maintenance updates that have been applied to the Security profiles that reside on the Security file. This program produces detail report that shows all update activity, including before and after images of any “edit” updates.

Update functions via the online Security subsystem result in records being written to the Security Change Audit file.

### Operation

The Security Change Audit Inquiry program runs as a batch job step.

### Files Used

#### *Input*

The following files are used as input:

Filename	Description
EDIXCHA	6.4 VSAM Security Change Audit File
EDICNTL	Control Card File
ERRCTL	Error Message and Control File
EDICFG	System Configuration File

#### *Output*

None.

#### *Reports*

Program EBDI073 produces the following reports:

Report Name	Description
EDIRPT	Security Change Audit Inquiry Report
EDILOG	Security Change Audit Processing Log
EDISUM	Security Change Audit Summary Report

## Control Card Options

The following Security Change Audit Inquiry operations and selection criteria are specified by control cards.

### Operations

SELECT (Default Operation)

### Selection Criteria

#### Defaults

FROM-UPDATE-DATE	01/01/1900
FROM-UPDATE-TIME	00:00
TO-UPDATE-DATE	12/31/2099
TO-UPDATE-TIME	23:59
USER	ALL
USER-ID	ALL

### Valid Criteria Values

#### From-Update--Date

Specifies the date that the update was applied to the Security record. Starts the range. Used with To-Update-Date.

MM/DD/YYYY

#### From-Update-Time

Specifies the time that the update was applied to the Security record. Starts the range. Used with To-Update-Time.

HH:MM

#### To-Update--Date

Specifies the date that the update was applied to the Security record. Ends the range. Used with From-Update-Date.

MM/DD/YYYY

#### To-Update-Time

Specifies the time that the update was applied to the Security record. Ends the range. Used with From-Update-Time.

HH:MM

#### User

Specifies the ID of online user who applied the change to the Security records.

ALL

#### User ID

Identifies the Security User ID used to restrict which Security change audit records are used to create the inquiry report.

ALL Indicates that all Security records will be used to create the inquiry report.

XXXXXXXX

## Security File Display Report

**EBDI009**

The Security File Display program (EBDI009) allows you to run an inquiry report against the online security file. This program creates a detailed list of all User information found on the security file in User Id order. Passwords are not displayed on either report.

### Files Used

#### *Input*

The following files are used as input:

Filename	Description
EDISECR	6.4 VSAM Security File
ERRCTL	Error Message and Control File
EDICFG	System Configuration File

#### *Reports*

Program EBDI009 produces the following reports:

Report Name	Description
SECDTL	Security File Display Report
EDILOG	Processing Log

*Example Reports*

Figure 4.27 is an example of the Security Detail report.

```

12/01/2005          GENTRAN: BASIC          PAGE   : 00001
12:00:00           SECURITY DETAIL REPORT   PROGRAM : EBDI009
                                           VERSION : 6.4

USER ID:  ADMIN      DIV: 000

      NAME:  LAST, FIRST M.                  INIT: XXX

PART  STD  DBK  MAP  ADM  SEC  MSG  CFG  GLB  PLUS  CTL  RTE  RDB  VPT
Y 1   Y 1   Y 1   Y 1   Y 1   Y 1   Y 1   Y 1   Y 1   N 3   N 3   N 3   N 6   N 4

*-----*

USER ID:  TESTID1    DIV: 000

      NAME:  TEST ID # 1 (FULL ACCESS), FIRST M.  INIT: XXX

PART  STD  DBK  MAP  ADM  SEC  MSG  CFG  GLB  PLUS  CTL  RTE  RDB  VPT
Y 1   Y 1   Y 1   Y 1   N 3   Y 1   N 3   N 3   N 3   N 3   N 3   N 3   N 6   N 4

*-----*

USER ID:  TESTID2    DIV: 000

      NAME:  TEST ID # 2 (LIMITED ACCESS), SECOND M.  INIT: XXX

PART  STD  DBK  MAP  ADM  SEC  MSG  CFG  GLB  PLUS  CTL  RTE  RDB  VPT
Y 2   Y 2   Y 2   Y 2   Y 2   Y 2   Y 2   Y 2   Y 2   N 3   N 3   N 3   N 6   N 4

*-----*

USER ID:  TESTID3    DIV: 000

      NAME:  TEST ID # 3 (READ ONLY ACCESS), THIRD M.  INIT: XXX

PART  STD  DBK  MAP  ADM  SEC  MSG  CFG  GLB  PLUS  CTL  RTE  RDB  VPT
Y 3   Y 3   Y 3   Y 3   N 3   Y 3   N 3   N 3   N 3   N 3   N 3   N 3   N 6   N 4

*-----*

```

**Figure 4.27 Security Detail Report**



## Separator Change Audit Inquiry

**EBDI072**

The Separator Change Audit Inquiry Report program (EBDI072) enables you to produce reports for maintenance updates that have been applied to the Priority Options that reside on the Separator file. This program produces detail report that shows all update activity, including before and after images of any “edit” updates.

Update functions via the online Separator subsystem result in records being written to the Separator Change Audit file.

### Operation

The Separator Change Audit Inquiry program runs as a batch job step.

### Files Used

#### Input

The following files are used as input:

Filename	Description
EDIZCHA	6.4 VSAM Separator Change Audit File
EDICNTL	Control Card File
ERRCTL	Error Message and Control File
EDICFG	System Configuration File

#### Output

None.

#### Reports

Program EBDI072 produces the following reports:

Report Name	Description
EDIRPT	Separator Change Audit Inquiry Report
EDILOG	Separator Change Audit Processing Log
EDISUM	Separator Change Audit Summary Report

## Control Card Options

The following Separator Change Audit Inquiry operations and selection criteria are specified by control cards.

### Operations

SELECT (Default Operation)

### Selection Criteria

#### Defaults

FROM-UPDATE-DATE	01/01/1900
FROM-UPDATE-TIME	00:00
TO-UPDATE-DATE	12/31/2099
TO-UPDATE-TIME	23:59
USER	ALL
TRANS-GROUP-ID	ALL
SENDER-ID	ALL
SENDER-QUAL	ALL
RECEIVER-ID	ALL
RECEIVER-QUAL	ALL
VERSION	ALL
TEST-PROD-IND	ALL
IGT-LEVEL-IND	ALL

### Valid Criteria Values

#### From-Update-Date

Specifies the date that the update was applied to the Separator record. Starts the range. Used with To-Update-Date.

MM/DD/YYYY

#### From-Update-Time

Specifies the time that the update was applied to the Separator record. Starts the range. Used with To-Update-Time.

HH:MM

#### To-Update-Date

Specifies the date that the update was applied to the Separator record. Ends the range. Used with From-Update-Date.

MM/DD/YYYY

#### To-Update-Time

Specifies the time that the update was applied to the Separator record. Ends the range. Used with From-Update-Time.

HH:MM

#### User

Specifies the ID of online user who applied the change to the Separator records.

ALL

**Trans-Group-ID**

The Transaction or Group ID used to restrict which Separator change audit records are used to create the inquiry report.

ALL  
XXXXXX

**Sender-Id**

Identifies the Sender Id used to restrict which Separator change audit records are used to create the inquiry report.

ALL  
XXXXX

**Sender-Qual**

The qualifier associated with the Sender Id used to restrict which Separator change audit records are used to create the inquiry report.

ALL  
XXXX

**Receiver-Id**

Identifies the Receiver Id used to restrict which Separator change audit records are used to create the inquiry report.

ALL  
XXXXX

**Receiver-Qual**

The qualifier associated with the Receiver Id used to restrict which Separator change audit records are used to create the inquiry report.

ALL  
XXXX

**Version**

The Version Id used to restrict which Separator change audit records are used to create the inquiry report.

ALL  
XXXXXXXXXXXXX

**Test-Prod-Ind**

The Test/Production Indicator used to restrict which Separator change audit records are used to create the inquiry report.

ALL  
T                    Test  
P                    Production

**IGT-Level-Ind**

Identifies the Interchange/Group/Transaction Level Indicator used to restrict which Separator change audit records are used to create the inquiry report.

ALL                    Indicates that all Separator records will be used to create the inquiry report.  
I                        Interchange  
G                        Group  
T                        Transaction

## Standard Change Audit Inquiry Report Program

**EBDI037**

The Standard Change Audit Inquiry Report program (EBDI037) enables you to produce reports for standards maintenance updates that have been applied to the 13 online standards files. This program produces either a summary level report of activity or a detail report that shows all update activity, including before and after images of any “edit” updates.

The Standard Change Audit file contains records that track changes made to the thirteen online Standards files. Add, delete, and update functions via the online Standard subsystem result in records being written to the Standard Change Audit file. The Batch copy utility (EBDI032) and the Batch Maintenance utility (EBDI039) will also write records to the Standard change audit file indicating add, delete, and replace functions.

### Operation

The Standards Change Audit Inquiry program runs as a batch job step.

### Files Used

#### *Input*

The following files are used as input:

Filename	Description
EDISCHA	Standards Change Audit File
EDICNTL	Control Card File
ERRCTL	Error Message and Control File
EDICFG	System Configuration File

#### *Output*

None.

#### *Reports*

Program EBDI037 produces the following reports:

Report Name	Description
EDIRPT	Databank Change Audit Inquiry Report
EDILOG	Databank Change Audit Processing Log
EDISUM	Databank Change Audit Summary Report

## Control Card Options

The following Standards Change Audit Inquiry operations and selection criteria are specified by control cards.

### Operations

SELECT (Default Operation)

### Selection Criteria

#### Defaults

FROM-UPDATE-DATE	01/01/1900
FROM-UPDATE-TIME	00:00
TO-UPDATE-DATE	12/31/2099
TO-UPDATE-TIME	23:59
USER	ALL
REPORT -TYPE	SUMMARY
REPORT -BY	VERSION
RECORD-TYPE	V
VERSION-ID	ALL
AGENCY	ALL
TRANSACTION-ID	ALL
TRANSACTION-REL-NUM	ALL
SEGMENT-ID	ALL
SEGMENT-VERSION	ALL
ELEMENT-ID	ALL
ELEMENT-VERSION	ALL
ASSOCIATION	ALL

### Valid Criteria Values

#### From-Update--Date

Specifies the date that the update was applied to the Standard. Starts the range. Used with To-Update-Date.

MM/DD/YYYY

#### From-Update-Time

Specifies the time that the update was applied to the Standard. Starts the range. Used with To-Update-Time.

HH:MM

#### To-Update--Date

Specifies the date that the update was applied to the Standard. Ends the range. Used with From-Update-Date.

MM/DD/YYYY

**To-Update-Time**

Specifies the time that the update was applied to the Standard. Ends the range. Used with From-Update-Time.

HH:MM

**User**

Specifies the ID of online user who applied the change to the Standards records.

ALL

XXX

**Report-Type**

Identifies the level at which the Inquiry should report.

FULL            Includes detailed information for each update.

SUMMARY       Includes summary list of updates.

**Report By**

Identifies the information that is to be included in the Inquiry report.

VERSION            All change audit records that satisfy the **Version ID** control card are used to create the inquiry report.

RECORD-TYPE       All change audit records that satisfy the **Record Type** control card (such as Code or Transaction) are used to create the inquiry report.

**Record Type**

Used with Report By control card to restrict the type of change audit records that are used to create the inquiry report.

V	–	Version
T	–	Transaction
S	–	Segment
E	–	Element
D	–	Dictionary
A	–	Activity
C	–	Code
L	–	Code List
O	–	Association
U	–	User Envelope (Structure)

**Version-Id**

Contains the 12-position version ID.

When used with the **Report-By = Version** control card, ALL indicates that all records are used to create the inquiry report. A specific version ID is used to restrict which records are used to create the inquiry report.

When used with the **Report-By = Record-Type** and **Record-Type=V** control cards, ALL indicates that only version records are used to create inquiry report. A specific version ID is used to restrict which version records are used to create the inquiry report.

ALL

XXXXXXXXXXXX

**Agency**

Contains the 3-position agency associated with the version ID.

ALL	All records that satisfy the <b>Version ID</b> control card are used to create the inquiry report regardless of the Agency value.
XXX	Only change audit records containing the agency that matches the specified agency are used to create the inquiry report.

**Transaction-Id**

Used with **Report By** and **Record Type = T** control cards.

ALL	All transaction records are used to create the inquiry report.
XXXXXX	The specified transaction ID is used to restrict which transaction records are used to create the inquiry report.

**Transaction-Rel-Num**

Contains the 1-position numeric transaction release number associated with the Transaction ID. Used for Tradacoms transactions only.

ALL	All transactions that satisfy the <b>Transaction ID</b> control card are used to create the inquiry report regardless of the Transaction Release Number value.
X	Only change audit records containing the Transaction release number that matches the specified Transaction release number are used to create the inquiry report.

**Segment-Id**

Used with **Report By** and **Record Type = S** control cards.

ALL	All segment records are used to create the inquiry report.
XXXXXX	The specified Segment ID is used to restrict which records are used to create the inquiry report.

**Segment-Version**

Contains the 2-position numeric segment version number associated with the Segment Id.

ALL	All segment records that satisfy the <b>Segment ID</b> control card are used to create the inquiry report regardless of the Segment Version value.
XX	Only change audit records containing the Segment Version number that match the specified Segment Version are used to create the inquiry report.

**Element-Id**

Used with **Report By** and **Record Type = E** control cards.

ALL	All Element records are used to create the inquiry report.
XXXXXX	The specified Element ID is used to restrict which element records are used to create the inquiry report.

**Element-Version**

Contains the 2-position numeric element version number associated with the Element Id.

ALL	All element records that satisfy the <b>Element Id</b> control card are used to create the inquiry report regardless of the <b>Element Version</b> value.
XX	Only change audit records containing the Element Version number that matches the specified Element version are used to create the inquiry report.

**Association**

Used with **Report By** and **Record Type = A** control cards.

ALL	All association records are used to create the inquiry report.
XXXXXX	A specified Association is used to restrict which association records are used to create the inquiry report.



## Standard Optimized Table Report

## EBDI035

The Standard Optimized Table Report (EBDI035) program produces a report that resembles an ASC X12 Standards Guide. The standards optimized table (created by EBDI030) is used as input. The layout of the report is similar to that of the Standards Report (EBDI036), which uses the online standard files for input.

Version ID, agency, and transaction ID are entered in the parameter records. Up to one hundred parameter records can be processed at one time. The program generates a separate report for each version/agency combination found in the parameter file. Versions processed under the \*ALL option, however, are presented as separate reports and are not combined with requests from other parameter records for the same version and agency.

The Standard Optimized Table report is printed by version using a hierarchical approach within each version. This report can be generated at three different levels, depending on the input parameter: A report level of 'S' results in a list of the selected transaction IDs. A report level of 'T' results in a list of the selected transaction IDs and a list of the segments for each selected transaction. A report level of 'D' results in a full report that contains detail information about each element/segment for selected transaction IDs.

Codes are not listed in this report. For code information, see the Standard Report EBDI036.

### Files Used

#### Input

The files in the following table are used as input.

Filename	Description
SYS030	Parameters File This file contains the Version ID, agency, and the transaction set that identify the standard report requested.
SYS001	Standards Optimized Table File This file is created by program EBDI030 and is used by the editor programs.
EDICFG	System Configuration File
EDISVER	Standard Version File This file contains the definition of all the standard versions. (See Note below.)
EDISTRN	Standard Transaction File This file contains all transaction sets defined for a standard. (See Note below.)
EDISSGD	Standard Segment Description File This file contains the descriptions for all segments. (See Note below.)
EDISELD	Standard Element Description File This file contains the descriptions for all elements. (See Note below.)

**Note:** This file is used for descriptions only.

*Input/Output*

The following file is used as output and input:

Filename	Description
SYS040	Temporary Sort File

*Output*

The following file is used as output:

Filename	Description
REPORT	Output Report File This is a spool file for output from the program.

*Parameter Record Layout*

The following table describes the parameter record layout.

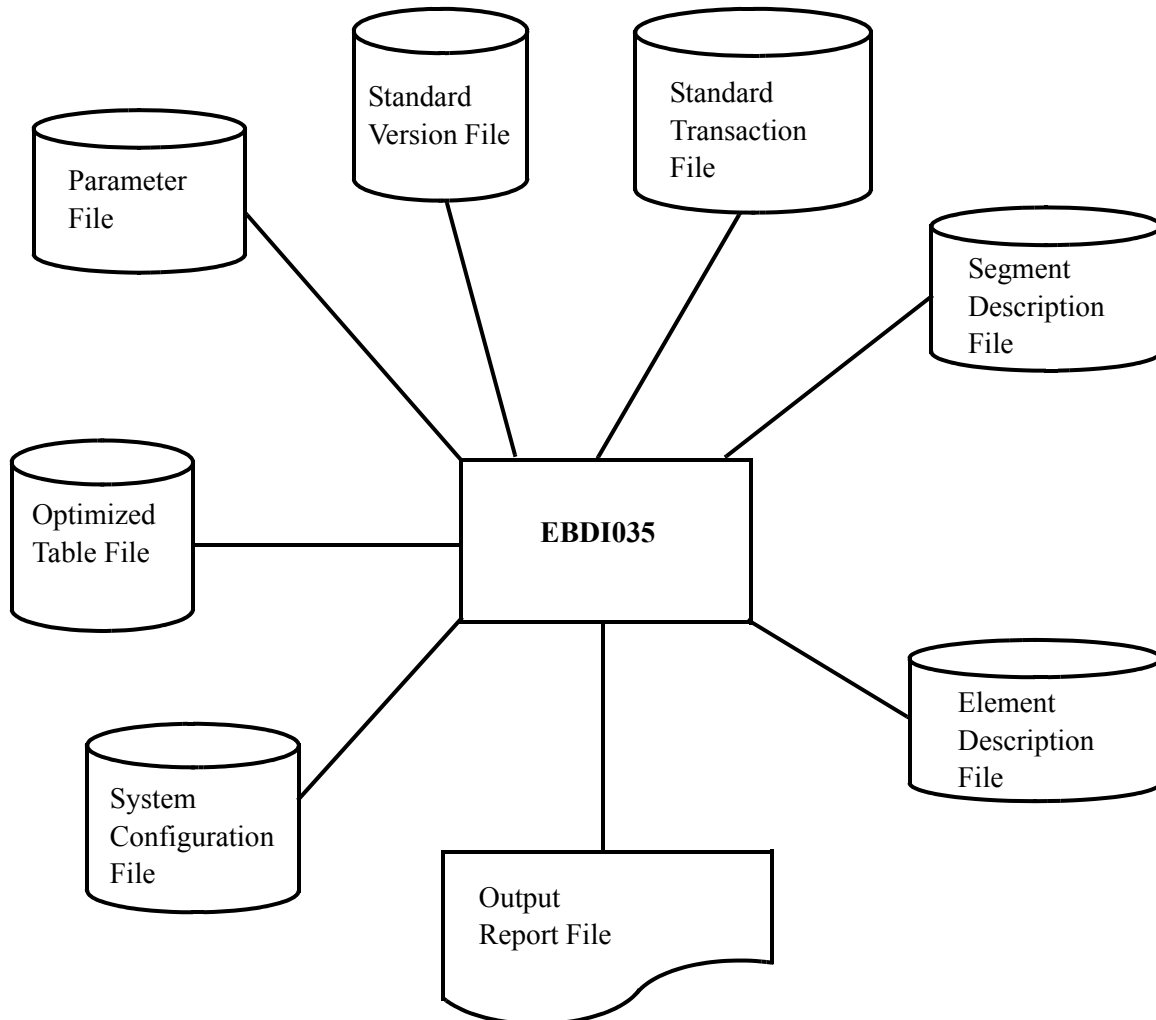
Position	Field Content	COBOL Picture	Format	Length	Description
01-12	Version ID	X(12)	AN	12	Version ID selected or *ALL
14-16	Agency	X(03)	AN	03	Agency code associated with the Version ID
18-23	Transaction Set	X(06)	AN	06	Transaction Set ID selected or *ALL For Tradacoms and Brokernet, see note below.
25	Release Number	X(01)	AN	01	Release number of the transaction set
27	Report Level	X(01)	AN	01	Report level for the selection: S – Summary, list of transaction IDs T – Transaction, list segments in transaction D – Detail, list elements within segments

**Note:** For Tradacoms or Brokernet versions, the transaction set field must be \*ALL or be the *first* transaction (message) in a Tradacoms logical document. For example, INVFIL would be the first message in the Tradacoms file format that is used to transmit invoices. The “chain” of messages that makes up the “invoice” logical document would be

INVFIL, INVOIC, VATTLR, and INVTLR. If INVFIL and the appropriate release number are entered in the parameter record, information about all the messages in the associated file format will be displayed. If transaction INVOIC is entered in a parameter record for version ANA001, the transaction will not be found.

*Standard Optimized Table Report Process*

Figure 4.28 illustrates the Standard Optimized Table report process.



**Figure 4.28 Program EBDI035 – Standard Optimized Table Report**

Sample Reports

Figure 4.29 through Figure 4.31 illustrate the Standard Optimized Table report in X12 Format. The figures include: Document for Version, Segments for Document, and Elements for Segment.

```

EBDI035  RUN DATE 12/01/2005  GENTRAN STANDARDS REPORT  RUN TIME 12:00:00  PAGE 1
          AGENCY: T          DOCUMENTS FOR VERSION: U3/1  AGENCY: T
VERSION: U3/1          AGENCY: T  TDCC (UCS) VERSION 3 RELEASE 1
          AGENCY: T          NTE NO.  FUNCT  IS/LE  REP  *-----ANA CHAIN-----*
DOCUMENT  RI SE  DESCRIPTION  FLT  SEG  IDENT  BOUND  IND  TRAN-REL  TRAN-REL  TRAN-REL
-----
875      0  PURCHASE ORDER (UCS)  N  29  OG  Y  N
    
```

Figure 4.29 Standard Optimized Table Report (Documents for Version)

GENTRAN STANDARDS REPORT  
SEGMENTS FOR DOCUMENT: 875

EBDI035 RUN DATE 12/01/2005

VERSION : U3/1 AGENCY : T RLSE : 0  
DOCUMENT : 875 PURCHASE ORDER (UCS)  
NO OF SEG : 0029

SEG SEQ	SEG ID	VER	SEG Y	T Q	R CD	M CD	MAN USE	MIN USE	MAX USE	** MIN	LOOP MAX	** MAX	ELE CNT	SEG GRP	LOOP ID	LOOP B/E	ACT CNT	DESCRIPTION
0001	G50	00			M	1	1	0	1				7					PURCHASE ORDER IDENTIFICATION
0002	LS	00			C	0	1	0	1				1					LOOP HEADER
0003	N1	00			C	0	1	0	10				4		0100	B		NAME
0004	N2	00			O	0	0	0	1				2		0100			ADDITIONAL NAME
0005	N3	00			O	0	0	0	2				2		0100			ADDRESS INFORMATION
0006	N4	00			O	0	0	0	1				6		0100	E		GEOGRAPHIC LOCATION
0007	LE	00			C	0	0	0	1				1					LOOP TRAILER
0008	N9	00			O	0	0	0	10				5					REFERENCE NUMBERS
0009	G61	00			O	0	0	0	3				5					CONTACT
0010	G62	00			O	0	0	0	10				5					DATE/TIME
0011	NTE	00			O	0	0	0	200				2					NOTE/SPECIAL INSTRUCTION
0012	G66	00			O	0	0	0	1				7					TRANSPORTATION INSTRUCTIONS
0013	LS	00			C	0	0	0	1				1					LOOP HEADER
0014	G68	00			C	0	0	0	1	999			15		0200	B		LINE ITEM DETAIL - PRODUCT
0015	G69	00			O	0	0	0	5				1		0200			LINE ITEM DETAIL - DESCRIPTION
0016	G70	00			O	0	0	0	2				9		0200			LINE ITEM DETAIL - MISCELLANEOUS
0017	N9	00			O	0	0	0	10				5		0200			REFERENCE NUMBERS
0018	LS	00			C	0	0	0	1				1		0200			LOOP HEADER
0019	G72	00			N	0	0	0	1	100			11		0210	B		ALLOWANCE OR CHARGE
0020	G73	00			C	0	0	0	10				1		0210	E		ALLOWANCE OR CHARGE DESCRIPTION
0021	LE	00			C	0	0	0	1				1		0200			LOOP TRAILER
0022	G23	00			O	0	0	0	20				16		0200	E		TERMS OF SALE
0023	LE	00			C	0	0	0	1				1					LOOP TRAILER
0024	LS	00			C	0	0	0	1				1					LOOP HEADER
0025	G72	00			N	0	0	0	1	100			11		0300	B		ALLOWANCE OR CHARGE
0026	G73	00			O	0	0	0	1				1		0300	E		ALLOWANCE OR CHARGE DESCRIPTION
0027	LE	00			C	0	0	0	1				1					LOOP TRAILER
0028	G23	00			O	0	0	0	20				16					TERMS OF SALE
0029	G76	00			M	1	1	1	1				9					TOTAL PURCHASE ORDER

Figure 4.30 Standard Optimized Table Report (Segments for Document)



*Field Definitions***Documents for Version (Displays all transactions selected for that version.)****Version:**

This field displays the Version ID selected.

**Agency:**

This field displays the agency code associated with the selected Version ID.

**(Description):**

This field displays a description of the standard version.

**Document:**

This field identifies the transaction.

**Rlse:**

This field displays the release number of a specific transaction.

**Description:**

This field displays a description of the transaction set.

**NTE Flt:**

This field indicates if the NTE (note) segment is permitted to appear outside of a defined sequence order in the transaction (to float). Valid values are:

Y	–	Yes
N	–	No
blank	–	NTE segment does not appear on transaction record.

**No. Seg:**

This field displays the number of segments in a transaction.

**Funct Ident:**

This field displays the functional identification of the transaction set.

**LS/LE Bound:**

This field identifies if LS/LE segments are used to show begin and end of loop structures within the transaction set.

**ANA Chain:**

For Tradacoms version ANA001 only: This field displays linkage of Tradacoms transactions (messages) that work together to accomplish a function as defined in a Tradacoms “file format”.

**Tran-Rel:**

This field displays the transaction ID and release number for a Tradacoms message that is part of the ANA chain.

**Segments for Document (Displays all segments within one transaction.)****Version:**

This field displays the Version ID selected.

**Agency:**

This field displays the agency code associated with the selected Version ID.

**Rlse:**

This field displays the release number of a specific transaction.

**Document:**

This field identifies the transaction.

**(Description):**

This field displays a description of the transaction.

**No. Of Seg:**

This field displays the number of segments in a transaction.

**Segment Seq:**

This field displays the sequence number of the segment within the transaction.

**Seg ID:**

This field displays the segment identifier.

**Seg Ver:**

This field displays the version number associated with the Segment ID.

**TY:**

This field displays the Segment type. Valid values are:

H	–	Header
D	–	Detail
S	–	Summary
F	–	Float

**RQ:**

This field indicates whether the first segment is required in a loop. Valid values are:

blank	–	First segment is required
N	–	First segment is not required

**Man. Code:**

This field displays the Segment Requirement Designator. Valid values are:

M	–	Mandatory
O	–	Optional
C	–	Conditional

**Min Use:**

This field displays the minimum number of times the segment may occur at that specific location within the transaction.

**Max Use:**

This field displays the maximum number of times the segment may occur at that specific location within the transaction.

**Loop – Min:**

This field displays the minimum number of times the loop may occur at that specific location within the transaction. The field is blank if the loop is not mandatory.

**Loop – Max:**

This field displays the maximum number of times the loop may occur at that specific location within the transaction.



**Ele Cnt:**

This field displays the total number of data elements within the segment.

**Loop ID:**

This field displays the identification for the loop.

**Loop B/E:**

This field indicates whether the segment is a beginning or ending segment in a loop. Valid values are:

B	–	Begin
E	–	End

**Act Cnt:**

This field displays the number of Element Activity records associated with this segment.

**Description:**

This field displays a description of the segment.

**Elements for Segment (Displays all elements within each segment within one transaction.)****Version:**

This field displays the version identification selected.

**Agency:**

This field displays the agency code associated with the selected Version ID.

**Rlse:**

This field displays the release number of a specific transaction.

**Document:**

This field identifies the transaction.

**(Description):**

This field displays a description of the transaction set.

**Seg ID:**

This field displays the segment identifier for the specified elements.

**Seg Ver:**

This field displays the version number associated with the Segment ID.

**Ele Seq:**

This field displays the sequence number of the element within the segment.

**Sub Ele:**

This field specifies a sub-element within an element.

**Man Ele:**

This field displays the Segment Requirement Designator. Valid values are:

M	–	Mandatory
O	–	Optional
C	–	Conditional

**Com Ele: (used for simple and composite elements)**

This field defines the specified Composite Element. Valid values are:

M	–	Mandatory
O	–	Optional
C	–	Conditional

**Aud Ind:**

This field displays an Audit Indicator. The Audit Indicator determines whether detail information (such as Invoice Number, PO Number) about the segment element is also written to the Info field on the Databank Directory. Valid values are:

Y	Yes, write the associated detail on the Databank Directory.
blank	Do not write the detail.

**Ele ID:**

This field displays the dictionary reference number for the specific data element record.

**Ele Ver:**

This field displays the version number associated with the Element ID.

**Grp Dsg:**

This designator displays a description of the group that contains the element.

**Grp Typ:**

This field displays a description of positioning of sub-elements within a composite element (group). Valid values are:

G	–	First sub-element within the group
blank	–	Last sub-element within the group
R	–	Any other sub-element within the group (not first or last)

**Rep Ind:**

This field indicates the maximum number of times an element or composite may repeat.

**Ele Type:**

This field displays the code for the element type. Valid values are:

AN	–	Alphanumeric
ID	–	Identifier type
R	–	Decimal type
Rn	–	Decimal type with “n” explicit decimal places
Nn	–	Numeric with “n” decimal places implied
Dn	–	Numeric with “n” decimal places implied
DT	–	Date
D8	–	Date
TM	–	Time

**Min:**

This field displays the minimum length for the element.

**Max:**

This field displays the maximum length for the element.

**Description:**

This field displays a description of the element.

**Cond Codes:**

This field displays the conditional codes for the element.

**Grp**

This field indicates whether or not this element is part of a group element:

Y	Yes, this element is part of a group element.
blank	This is a simple element.

**Cde Val:**

This field indicates whether or not element codes appears on the Code file. Valid values are:

Y	Yes, element codes appear on Code file.
N	Element type is "ID" but there are no codes for the element in the Code file.
blank	Element codes do not appear on Code file.

## Standard Report

**EBDI036**

The Standard Report (EBDI036) program produces a complete report that resembles an ASC X12 Standards Guide. The online standard files are used to generate the report. Enter the version and transaction set of the selected standard report in the parameter record. You can enter up to five parameter records at once. The program generates a separate report for each parameter record containing a different Version ID. The program combines parameter records entered with the same Version ID into one report.

The Standard report is printed by version using a hierarchical approach within each version. The details of this report include each transaction selected for a version, the segments within each transaction, the elements within each segment, and a list of codes and their descriptions (if requested). A field in the parameter record enables you to specify whether to print the standard codes for that version and transaction set. If you specify that the standard codes should be printed, the standard codes file is read by the program if the element type is equal to an ID-type field. Those codes are written to a temporary VSAM file and then written to the standards report from the temporary file when all transaction sets are processed for one version.

### Files Used

#### *Input*

The following files are used as input:

Filename	Description
SYS030	Parameters File This file contains the Version ID, agency, and the transaction set that identify the standard report requested.
EDICFG	System Configuration File
EDISVER	Standard Version File This file contains the definition of all the standard versions.
EDISTRN	Standard Transaction File This file contains all transaction sets defined for a standard.
EDISSEG	Standard Segment File This file contains all segments defined for a standard.
EDISSGD	Standard Segment Description File This file contains the descriptions for all segments.
EDISACT	Standard Activity File This file contains all element activity records.
EDISELE	Standard Element File This file contains all the elements defined for a standard.
EDISELD	Standard Element Description File This file contains the descriptions for all elements.

Filename	Description
EDISDIC	Standard Dictionary File This file contains all data elements defined for a standard, identified by reference number.
EDISCD1	Standard Code File 1 This file contains all the codes defined for an element with a maximum length of 1 – 5.
EDISCD2	Standard Code File 2 This file contains all the codes defined for an element with a maximum length of 6 – 10.
EDISCD3	Standard Code File 3 This file contains all the codes defined for an element with a maximum length of 11 – 20.
EDISCD4	Standard Code File 4 This file contains all the codes defined for an element with a maximum length of 21 – 40.

### *Input/Output*

The following file is used as input and output:

Filename	Description
SYS020	Temporary Reference File This file contains the standard codes of ID-type element fields with a key of release number and dictionary reference number. These codes are written to the standards report when all transaction sets for one version are processed.

### *Output*

The following file is used as output:

Filename	Description
SYS010	Output Report File This is a spool file for the output of this program, sorted hierarchically within each version.

*Parameter Record Layout*

The following table describes the parameter record layout.

Position	Field Content	COBOL Picture	Format	Length	Descriptions
001 – 012	Version ID	X(12)	AN	12	The name assigned to the definition of the version selected.
013 – 015	Agency	X(03)	AN	03	Agency code associated with the Version ID.
016 – 021	Transaction Set	X(06)	AN	06	The name assigned to the definition of the transaction set or *ALL.
022 – 022	Release Number	X(01)	AN	01	The release number of the transaction set.
023 – 023	Print Codes Flag	X(01)	AN	01	Specifies whether standard codes are printed for each version and transaction set. Valid values: <b>Y</b> or space.

Standard Report Process

Figure 4.32 illustrates the Standard report (Documents for Version) process.

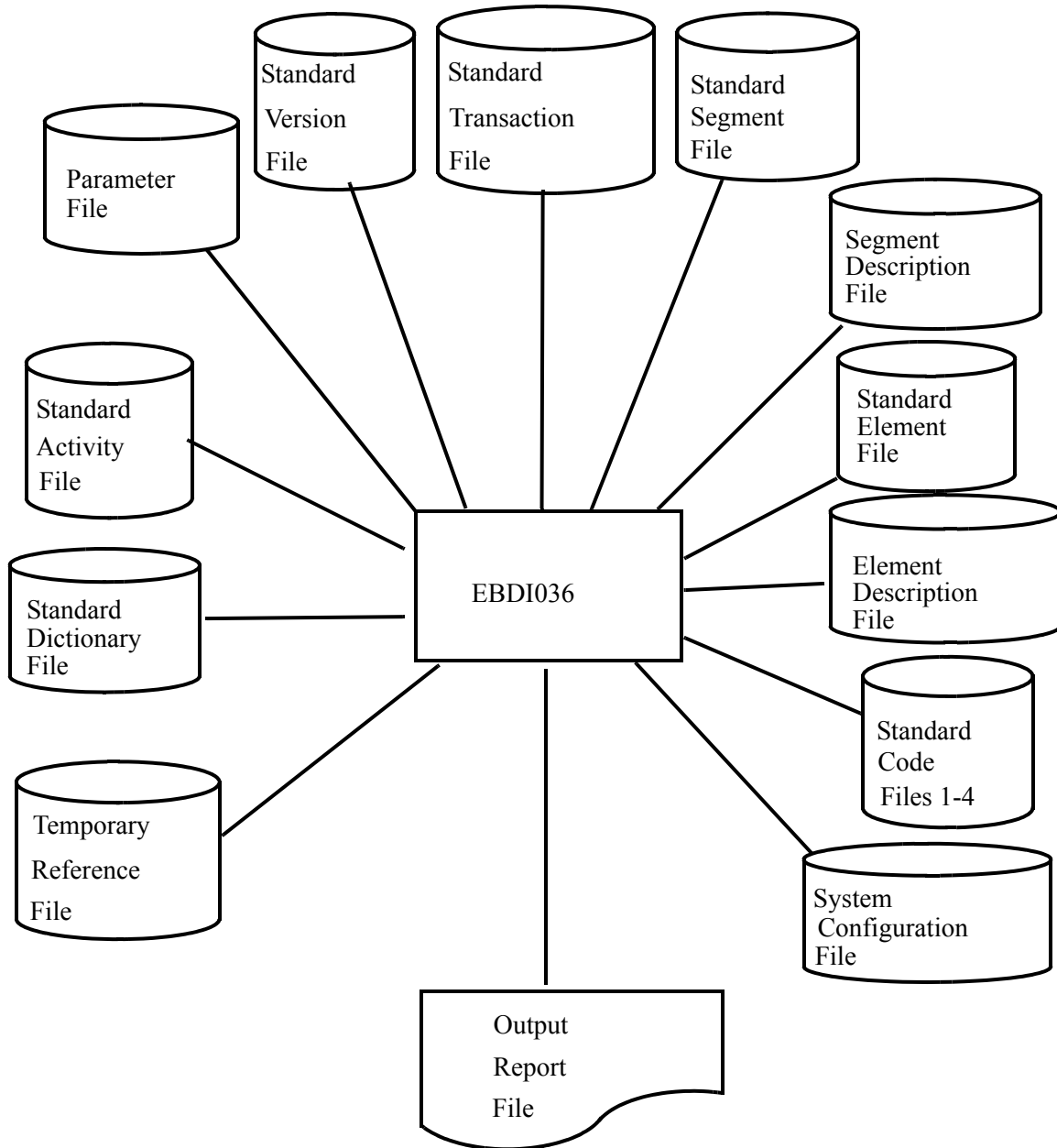


Figure 4.32 Program EBDI036 – Standard Report

Example Reports

Figure 4.33 through Figure 4.36 illustrate the Standard report. They include: Document for Version, Segments for Document, Elements for Segment, and the Codes for Version.

```

EBDI036  RUN DATE 12/01/2005  AGENCY: T  TDCC (UCS) VERSION 3 RELEASE 1  PAGE 1
          RUN TIME 18:45:12
          AGENCY: T
          GENTRAN STANDARDS REPORT
          DOCUMENTS FOR VERSION: U3/1
          VERSION: U3/1
          DOCUMENT  RLSE  DESCRIPTION  NTE  NO. OF  FUNCT  LS/LE
          875      0  PURCHASE ORDER (UCS)  FLOAT  SEGMENTS  IDENT  BOUND
          -----
  
```

Figure 4.33 Standard Report (Documents for Version)



EBDI036 RUN DATE 12/01/2005 GENTRAN STANDARDS REPORT RUN TIME 12:00:00 PAGE 2  
 SEGMENTS FOR DOCUMENT: 875

VERSION : U3/1 AGENCY : T  
 DOCUMENT : 875 PURCHASE ORDER (UCS) RLSE: 0  
 NO OF SEG: 0029

SEG	SEG	SEG	MAN.	MAX	MAX	MAX	ELE	LOOP	LOOP	ACT	DESCRIPTION
SEQ	ID	VER	TYPE	REQ	CODE	USE	LOOPS	CNT	ID	B/E	CNT
0001	G50	00	H		M	1	1	7			PURCHASE ORDER IDENTIFICATION
0002	LS	00	H		C	1	1	1			LOOP HEADER
0003	N1	00	H		C	1	10	4	0100	B	NAME
0004	N2	00	H		O	1		2	0100		ADDITIONAL NAME
0005	N3	00	H		O	2		2	0100		ADDRESS INFORMATION
0006	N4	00	H		O	1		6	0100	E	GEOGRAPHIC LOCATION
0007	LE	00	H		C	1		1			LOOP TRAILER
0008	N9	00	H		O	10		5			REFERENCE NUMBERS
0009	G61	00	H		O	3		5			CONTACT
0010	G62	00	H		O	10		5			DATE/TIME
0011	NTE	00	H		O	200		2			NOTE/SPECIAL INSTRUCTION
0012	G66	00	H		O	1		7			TRANSPORTATION INSTRUCTIONS
0013	LS	00	H		C	1		1			LOOP HEADER
0014	G68	00	D		C	1	999	15	0200	B	LINE ITEM DETAIL - PRODUCT
0015	G69	00	D		O	5		1	0200		LINE ITEM DETAIL - DESCRIPTION
0016	G70	00	D		O	2		9	0200		LINE ITEM DETAIL - MISCELLANEOUS
0017	N9	00	D		O	10		5	0200		REFERENCE NUMBERS
0018	LS	00	D		C	1		1	0200		LOOP HEADER
0019	G72	00	D	N	O	1	100	11	0210	B	ALLOWANCE OR CHARGE
0020	G73	00	D		O	10		1	0210	E	ALLOWANCE OR CHARGE DESCRIPTION
0021	LE	00	D		C	1		1	0200		LOOP TRAILER
0022	G23	00	D		O	20		16	0200	E	TERMS OF SALE
0024	LS	00	S		C	1		1			LOOP HEADER
0025	G72	00	S	N	O	1	100	11	0300	B	ALLOWANCE OR CHARGE
0026	G73	00	S		O	1		1	0300	E	ALLOWANCE OR CHARGE DESCRIPTION
0027	LE	00	S		C	1		1			LOOP TRAILER
0028	G23	00	S		O	20		16			TERMS OF SALE
0029	G76	00	S		M	1		9			TOTAL PURCHASE ORDER

Figure 4.34 Standard Report (Segments for Document)

GENTRAN STANDARDS REPORT  
ELEMENTS FOR SEGMENT

EBDI036 RUN DATE 12/01/2005

VERSION : U3/1 AGENCY : T RLSE: 0  
DOCUMENT : 875 PURCHASE ORDER (UCS)

SEG ID	ELE SEQ	SUB ELE	MAN ELE	COM ELE	AUD IND	ELE ID	ELE VER	GRP DSG	ELE DESCRIPTION	COND CODES	CDE VAL
G50	00	PURCHASE ORDER IDENTIFICATION									
	001	00	M	M		473	00		ID 001 0001 ORDER STATUS CODE		
	002	00	M	M		323	00		DT 006 0006 PURCHASE ORDER DATE		
	003	00	M	M	Y	324	00		AN 001 0022 PURCHASE ORDER NUMBER		
	004	00	O	O		441	00		ID 001 0001 TAX EXEMPT CODE		
	005	00	C	C		474	00		AN 001 0022 MASTER REFERENCE (LINK) NUMBER	P	005006
	006	00	C	C		472	00		NO 006 0006 LINK SEQUENCE NUMBER	P	005006
	007	00	O	O		92	00		ID 002 0002 PURCHASE ORDER TYPE CODE		
LS	00	LOOP HEADER									
	001	00	M	M		447	00		ID 001 0004 LOOP IDENTIFIER CODE		
N1	00	NAME									
	001	00	M	M		98	00		ID 002 0002 ENTITY IDENTIFIER CODE		
	002	00	C	C		93	00		AN 001 0035 NAME	R	002003
	003	00	C	C		66	00		ID 001 0002 IDENTIFICATION CODE QUALIFIER	R	002003
	004	00	C	C		67	00		ID 002 0017 IDENTIFICATION CODE	P	003004
N2	00	ADDITIONAL NAME									
	001	00	M	M		93	00		AN 001 0035 NAME		
	002	00	O	O		93	00		AN 001 0035 NAME		
N3	00	ADDRESS INFORMATION									
	001	00	M	M		166	00		AN 001 0035 ADDRESS INFORMATION		
	002	00	O	O		166	00		AN 001 0035 ADDRESS INFORMATION		
N4	00	GEOGRAPHIC LOCATION									
	001	00	C	C		19	00		AN 002 0019 CITY NAME	R	001005
	002	00	C	C		156	00		ID 002 0002 STATE OR PROVINCE CODE	C	001002
	003	00	O	O		116	00		ID 004 0009 POSTAL CODE		
	004	00	O	O		26	00		ID 002 0002 COUNTRY CODE		
	005	00	C	C		309	00		ID 001 0002 LOCATION QUALIFIER	R	001005
	006	00	C	C		310	00		AN 001 0025 LOCATION IDENTIFIER	P	005006
LE	00	LOOP TRAILER									
	001	00	M	M		447	00		ID 001 0004 LOOP IDENTIFIER CODE		
N9	00	REFERENCE NUMBERS									
	001	00	M	M		128	00		ID 002 0002 REFERENCE NUMBER QUALIFIER		
	002	00	C	C		127	00		AN 001 0030 REFERENCE NUMBER	R	002003
	003	00	C	C		369	00		AN 001 0045 FREE-FORM DESCRIPTION		

Figure 4.35 Standard Report (Elements for Segment)

PAGE 9

RUN TIME 12:00:00

GENTRAN STANDARDS REPORT  
CODES FOR VERSION: U3/1

RUN DATE 12/01/2005

EBDI036

VERSION: U3/1 Agency: T

ELEMENT ID.	RLSE	COMPOSITE /CODE	DEFINITION
128	00	AA	ACCOUNTS RECEIVABLE STATEMENT
		AB	ACCEPTABLE SOURCE BUYER ID
		AC	AIR CARGO TRANSFER MANIFEST
		AD	ACCEPTABLE SOURCE DUNS NUMBER

Figure 4.36 Standard Report (Codes for Version)

*Field Definitions***Documents for Version (Displays all transactions selected for that version)****Version:**

This field displays the Version ID selected.

**Agency:**

This field displays the agency code associated with the selected Version ID.

**(Description):**

This field is used to display a description of the standard version.

**Document:**

This field identifies the transaction.

**Rlse:**

This field displays the release number of a specific transaction.

**Description:**

This field provides descriptive information about the transaction set.

**NTE Float:**

This field is used to indicate if the NTE (note) segment is permitted to appear outside of a defined sequence order in the transaction (to float). Valid values are:

Y	–	Yes
N	–	No
blank	–	NTE segment does not appear on transaction record.

**No. Of Segments:**

This field displays the number of segments in a transaction.

**Funct Ident:**

This field displays the functional identification of the transaction set.

**LS/LE Bound:**

This field identifies if LS/LE segments are used to show begin and end of loop structures within the transaction set.

**Segments for Document (Displays all segments within one transaction)****Version:**

This field displays the Version ID selected.

**Agency:**

This field displays the agency code associated with the selected Version ID.

**Rlse:**

This field displays the release number of a specific transaction.

**Document:**

This field identifies the transaction.

**(Description):**

This field provides descriptive information about the transaction.

**No. Of Seg:**

This field displays the number of segments in a transaction.

**Segment Seq:**

This field displays the sequence number of the segment within the transaction.

**Seg ID:**

This field displays the segment identifier.

**Seg Ver:**

This field displays the version number associated with the Segment ID.

**Type:**

This field displays the Segment type. Valid values are:

H	–	Header
D	–	Detail
S	–	Summary
F	–	Float

**Req:**

This field indicates whether the first segment is required in a loop. Valid values are:

blank	–	First segment is required
N	–	First segment is not required

**Man. Code:**

This field displays the Segment Requirement Designator. Valid values are:

M	–	Mandatory
O	–	Optional
C	–	Conditional

**Max Use:**

This field displays the maximum number of times the segment may occur at that specific location within the transaction.

**Max Loops:**

This field displays the maximum number of times the loop may occur at that specific location within the transaction.

**Ele Cnt:**

This field displays the total number of data elements within the segment.

**Loop ID:**

This field displays the identification for the loop.

**Loop B/E:**

This field indicates whether the segment is a beginning or ending segment in a loop. Valid values are:

B	–	Begin
E	–	End

**Act Cnt:**

This field displays the number of Element Activity records associated with this segment.

**Description:**

This field provides descriptive information about the segment.

**Elements for Segment (Displays all elements within each segment within one transaction)****Version:**

This field displays the version identification selected.

**Agency:**

This field displays the agency code associated with the selected Version ID.

**Rlse:**

This field displays the release number of a specific transaction.

**Document:**

This field identifies the transaction.

**(Description):**

This field is used to display a description of the transaction set.

**Seg ID:**

This field displays the segment identifier for the specified elements.

**Seg Ver:**

This field displays the version number associated with the Segment ID.

**Ele Seq:**

This field displays the sequence number of the element within the segment.

**Sub Ele:**

This field specifies a sub–element within an element.

**Man Ele:**

This field displays the Segment Requirement Designator. Valid values are:

M	–	Mandatory
O	–	Optional
C	–	Conditional

**Com Ele: (used for simple and composite elements)**

This field defines the specified Composite Element. Valid values are:

M	–	Mandatory
O	–	Optional
C	–	Conditional

**Aud Ind:**

This field displays an Audit Indicator. The Audit Indicator determines whether detail information (i.e., Invoice Number, PO Number) about the segment element is also written to the Info field on the Databank Directory. Valid values are:

Y	Yes, write the associated detail on the Databank Directory
blank	Do not write the detail

**Ele ID:**

This field displays the dictionary reference number for the specific data element record.

**Ele Ver:**

This field displays the version number associated with the Element ID.

**Grp Dsg:**

This designator field is used to describe the group that contains the element.

**Ele Type:**

This field displays the code for the element type. Valid values are:

AN	–	Alphanumeric
ID	–	Identifier Type
R	–	Decimal Type
Rn	–	Decimal type with “n” explicit decimal places
Nn	–	Numeric with “n” decimal places implied
Dn	–	Numeric with “n” decimal places implied
DT	–	Date
D8	–	Date
TM	–	Time

**Ele Size Min:**

This field displays the minimum length for the element.

**Ele Size Max:**

This field displays the maximum length for the element.

**Description:**

This field provides descriptive information about the element.

**Cond Codes:**

This field displays the conditional codes for the element.

**Grp**

This field indicates whether or not this element is part of a group element:

Y	Yes, this element is part of a group element.
blank	This is a simple element.

**Cde Val:**

This field indicates whether or not element codes appears on the Code file.

Valid values are:

Y	Yes, element codes appear on Code file
blank	Element codes do not appear on Code file

**Codes for Version (Displays a list of standard codes for one version)****Version:**

This field displays the version identification of the standard.

**Element ID:**

This field displays the dictionary reference number for the specific Data Element record.

**Rlse:**

This field displays the release number of a specific transaction.

**Composite/Code:**

This field identifies the part of the defined data element code (Composite), followed by a space and then the 10–position alphanumeric field used to define the composite value (Code).

**Definition:**

This field provides descriptive information about the composite/code record.

## Transaction Change Audit Inquiry Report Program

**EBDI063**

The Transaction Change Audit Inquiry Report program (EBDI063) enables you to produce reports for Transaction maintenance updates that have been applied to the three Mapping Transaction files. This program produces either a summary level report of activity or a detail report that shows all update activity, including before and after images of any “edit” updates.

The Transaction Change Audit file contains records that track changes made to the three Mapping Transaction files (transaction, header, segment, and element). Add, delete, and update functions via the online Transaction subsystem result in records being written to the Transaction Change Audit file. The Batch Copy utility (EBDI080), Batch Version Comparison and Conversion utility (EBDI1088), and the Batch Transaction Re-sequence (EBDI089) will also write records to the Transaction change audit file indicating add, delete, and replace functions. The online Upload process (EDIX235), using component EDIR080, will also write a change audit record to indicate an add or replace event.

### Operation

The Transaction Change Audit Inquiry program runs as a batch job step.

### Files Used

#### *Input*

The following files are used as input:

Filename	Description
EDITCHA	Transaction Change Audit File
EDICNTL	Control Card File
ERRCTL	Error Message and Control File
EDICFG	System Configuration File

#### *Output*

None.

#### *Reports*

Program EBDI063 produces the following reports:

Report Name	Description
EDIRPT	Transaction Change Audit Inquiry Report
EDILOG	Transaction Change Audit Processing Log
EDISUM	Transaction Change Audit Summary Report



## Control Card Options

The following Transaction Change Audit Inquiry operations and selection criteria are specified by control cards.

### Operations

SELECT (Default Operation)

### Selection Criteria

#### Defaults

FROM-UPDATE-DATE	01/01/1900
FROM-UPDATE-TIME	00:00
TO-UPDATE-DATE	12/31/2099
TO-UPDATE-TIME	23:59
USER	ALL
REPORT -TYPE	SUMMARY
TRANSACTION-ID	ALL
SEND-RECEIVE	ALL

### Valid Criteria Values

#### From-Update--Date

Specifies the date that the update was applied to the Transaction. Starts the range. Used with To-Update-Date.

MM/DD/YYYY

#### From-Update-Time

Specifies the time that the update was applied to the Transaction. Starts the range. Used with To-Update-Time.

HH:MM

#### To-Update--Date

Specifies the date that the update was applied to the Transaction. Ends the range. Used with From-Update-Date.

MM/DD/YYYY

#### To-Update-Time

Specifies the time that the update was applied to the Transaction. Ends the range. Used with From-Update-Time.

HH:MM

#### User

Specifies the ID of online user who applied the change to the Transaction records.

ALL  
XXX

**Report-Type**

Identifies the level at which the Inquiry should report.

FULL	Includes detailed information for each update.
SUMMARY	Includes summary list of updates.

**Transaction-Id**

Identifies the Transaction ID used to restrict which Transaction change audit records are used to create the inquiry report.

ALL	Indicates that all Transaction records will be used to create the inquiry report.
XXXXXXXXXX	Includes only the change audit records found for the specific Application ID.

**Send-Receive**

Specifies the direction of the Transaction. Used with Transaction-Id to restrict the records used to create the inquiry report.

S	–	Send
R	–	Receive
ALL	–	Send and Receive

# Databank Files

## Overview

This chapter contains the following topics related to the databank files:

Topic	Page
Acknowledgment Reconciliation File ..... EDIDAR .....	8-3
Change Audit Databank Archive File ..... EDICAAR.....	8-4
Formatted Network File..... EDIFNF .....	8-5
Inbound Application Databank Archive File..... EDIIAAR.....	8-6
Inbound Application Databank Change Audit File ..... EDIICA .....	8-7
Inbound Application Databank Directory ..... EDIIAA.....	8-9
Inbound Application Databank Message Store ..... EDIIAS .....	8-11
Inbound Application Databank Pending File ..... EDIIAP .....	8-13
Inbound Application Databank Transaction Queue ..... EDIQ095.....	8-15
Inbound EDI Databank Archive File..... EDIIEAR .....	8-17
Inbound EDI Databank Change Audit File ..... EDIIECA .....	8-18
Inbound EDI Databank Directory File ..... EDIIEA .....	8-20
Inbound EDI Databank Link File ..... EDIIEL .....	8-22
Inbound EDI Databank Message Store File ..... EDIIES.....	8-24
Inbound EDI Databank Pending File..... EDIIEP.....	8-27
Inbound EDI Databank Transaction Queue..... EDIQ097.....	8-28
Network Reconciliation File..... EDINRC .....	8-30
Network Response File..... EDINR .....	8-31
Outbound Application Databank Archive File..... EDIOAAR .....	8-32
Outbound Application Databank Change Audit File ..... EDIOACA .....	8-33
Outbound Application Databank Directory..... EDIOAA .....	8-35
Outbound Application Databank Link File ..... EDIOAL .....	8-37
Outbound Application Databank Message Store..... EDIOAS.....	8-39
Outbound Application Databank Pending File..... EDIOAP.....	8-41
Outbound Application Databank Transaction Queue..... EDIQ093.....	8-42
Outbound EDI Databank Archive File ..... EDIOEAR.....	8-44
Outbound EDI Databank Change Audit File..... EDIOECA.....	8-45
Outbound EDI Databank Directory..... EDIOEA .....	8-47

Outbound EDI Databank Message Store.....	EDIOES .....	8-50
Outbound EDI Databank Pending File.....	EDIOEP .....	8-53
Outbound EDI Databank Transaction Queue.....	EDIQ091 .....	8-54
Reprocess Work File .....	EDIRW .....	8-56

## Acknowledgment Reconciliation File

**EDIDAR**

### Description

The Acknowledgment Reconciliation (EDIDAR) file contains information from inbound acknowledgments, which is used to reconcile back to the original outbound EDI transactions.

The Acknowledgment Reconciliation file is a standard sequential file with a record length of 215 (DSORG=PS,LRECL=215, RECFM=FB).

### Activity

This file is relatively stable.

### Program References

The following programs reference the Acknowledgment Reconciliation file:

Program	Purpose
Inbound Editor	Creates the data set
Acknowledgment Reconciliation/Monitor	Reads the file

### Disk Space Requirements

The space requirements varies according to the number of inbound acknowledgments. Space requirements can be calculated based on:

$$\begin{array}{rcl}
 \# \text{ of records} & = & \# \text{ interchange acknowledgments} \\
 + & & \# \text{ of group acknowledgments} \\
 + & & \# \text{ of transaction acknowledgments}
 \end{array}$$

TRACKS(20,5) should accommodate most situations.

## Change Audit Databank Archive File

**EDICAAR**

### Description

The Change Audit Databank Archive (EDICAAR) file is a standard sequential data set that contains documents removed from the databank using the Databank Change Audit Maintenance program with the Archive Control option.

The Archive file is a standard sequential file with a maximum record length of 5082 (DSORG=PS,LRECL=5082,RECFM=VB, BLKSIZE=32004).

### Activity

The volume of the file depends on the number of documents being removed from the databank.

### Reference

The following program references the Archive file:

Program	Purpose
Change Audit Databank Maintenance	Creates the data set (Purge or Housekeeping mode)
	Reads the data set (Restore mode)

### Disk Space Requirements

The space requirements depend on databank activity. The Archive file contains images of the Databank Change Audit records deleted by databank maintenance.

---

## Formatted Network File

**EDIFNF**

### Description

The Formatted Network (EDIFNF) file is used as a temporary sequential file to collect network status information.

The Formatted Network file is a standard sequential file with a record length of 600 (DSORG=PS, LRECL=600).

### Activity

This is an operational file which normally has low activity.

### Program References

The following programs reference the Formatted Network file:

Program	Purpose
Network Reconciliation Program	Reads the file
Network Response Programs	Creates the file

### Data Set Recovery

You can re-create the Formatted Network file by re-running the appropriate Network Response program with the appropriate data.

### Disk Space Requirements

The space requirements for the Formatted Network file depends on the anticipated outbound interchange volumes and the frequency of retrieving and processing the network response files.

A guideline to use is one record for each outbound interchange that is being tracked and for which a response has been received from the network.

TRACKS(5,2) addresses most requirements.

**Inbound Application Databank Archive File****EDIIAAR****Description**

The Inbound Application Databank Archive (EDIIAAR) file is a standard sequential data set that contains documents removed from the databank using the Inbound Application Databank Maintenance program with the Archive Control option.

The Archive file is a standard sequential file with a maximum record length of 5082 (DSORG=PS,LRECL=5082,RECFM=VB).

**Activity**

The Archive file results from document transactions. The volume of the file depends on the number of documents being removed from the databank.

**Reference**

The following program references the Archive file:

<b>Program</b>	<b>Purpose</b>
Inbound Application Databank Maintenance	Creates the data set (Purge or Housekeeping mode) Reads the data set (Restore mode)

**Disk Space Requirements**

The space requirements depend on databank activity. The Archive file contains images of the Inbound Application Databank Directory and Databank Message Store records deleted by databank maintenance.



## Inbound Application Databank Change Audit File

## EDIACA

### Description

The Inbound Application Databank Change Audit (EDIACA) file contains detailed audit information of all online updates which are applied to the databank. This information is available for online or batch inquiry.

### Activity

This is a low activity operational file.

### Program References

The following programs reference the Inbound Application Databank Change Audit file:

Program	Purpose
Inbound Application Databank Extract	Updates the file
CICS Databank Maintenance	Reads the file
Databank Change Audit Inquiry	Reads the file
Databank Change Audit Maintenance	Updates the file

### Data Set Recovery

Before any files are restored, the online Change Audit should be reviewed and any outstanding online updates for the Inbound Application Databank noted (outstanding updates have an applied date and a time of zeros).

The Inbound Application Databank Directory, Message Store, Change Audit, Pending and Transaction Queue files, as well as the Inbound EDI Databank Link file are all closely related. If you need to recover the Inbound Application Databank Change Audit file and the Inbound Application Databank Directory and Inbound Application Databank Message Store files, you must recover the Inbound EDI Databank Link file at the same time, using the same method.

**Note:** If you are **re-initializing** the Change Audit file, you do not need to re-initialize your Directory, Message, Store and Link files.

The Inbound Application Databank Pending and Inbound Application Databank Transaction Queue files must be re-initialized and any pending online changes re-applied.

The Inbound Application Databank Directory, Message Store, and Change Audit files, and Inbound EDI Databank Link file can be recreated from either a recent backup or re-loaded from the Gentran:Basic system tape.

## Disk Space Requirements

One Change Audit record is created for each update performed against the Inbound Application Databank. Space utilization for this file is maintained with retention periods.

**Note:** In calculating the number of records, there are several equivalent methods of accounting for system activity, retention period, and cycle. For simplicity, the following calculations do not include these factors. System activity, retention period, and cycle must be factored into these calculations. The example shows one possible method.

### Example

An average of 2 updates are performed each day against the Inbound Application Databank.

The Databank Change Audit Maintenance program is run once a month with a retention period of 30 days specified.

First we calculate the average monthly volume.

Updates per month = 60 (30 days \* 2 updates per day)

Then we determine the number of records on file, given the retention options chosen. To simplify the calculation, assume that all updates are successfully processed within one day.

Records created over retention period = 60

Records created between maintenance runs = 60

Records on file = 120 (60 + 60)

## VSAM Data Set Definition

```

DEFINE          CLUSTER(NAME(@@@@@.EDI.EDIACA)          -
                VOLUME(VVVVVV)                        -
                RECORDS(120 30)                        -
                FREESPACE(30 10)                       -
                KEYS(116 0)                            -
                RECORDSIZE(1700 2100)                  -
                SHR(2 3))                              -
                DATA(NAME(@@@@@.EDI.EDIACA.DATA)     -
                CISZ(4096))                            -
                INDEX(NAME(@@@@@.EDI.EDIACA.INDEX))

```

## Inbound Application Databank Directory

EDIIAA

### Description

The Inbound Application Databank Directory (EDIIAA) contains status and control information for inbound application data. The Inbound Application Databank Directory also maintains records for controlling data recovery.

### Activity

This is an operational file with high activity.

### Program References

The following programs reference the Inbound Application Databank Directory:

Program	Purpose
Inbound Mapper	Updates the directory
Inbound Application Databank Maintenance	Updates the directory
Inbound Application Databank Extract	Updates the directory
Application Databank Inquiry	Reads, and optionally updates the directory
CICS Databank Maintenance	Reads the directory

### Data Set Recovery

Before any files are recovered, you should review the online Change Audit and note any outstanding updates (outstanding updates have an applied date and a time of zeros).

The Inbound Application Databank Directory, Message Store, Change Audit, Pending and Transaction Queue files, as well as the Inbound EDI Databank Link file are all closely related. If you need to recover the Inbound Application Databank Directory, Inbound Application Databank Message Store, and Inbound Application Databank Change Audit files, you must recover the Inbound EDI Databank Link file at the same time, using the same method.

**Note:** The Inbound Application Databank Pending and Inbound Application Databank Transaction Queue files must be re-initialized and any pending online changes re-applied.

The Inbound Application Databank Directory, Message Store, Change Audit files and the Inbound EDI Databank Link file can be recreated from either a recent backup or re-loaded from the Gentran:Basic system tape.

## Disk Space Requirements

The exact space required depends on the level of system activity, the retention period specified, and the frequency of Databank Maintenance. One record is required for each document on the Inbound Application Databank.

**Note:** Maintaining Directory data on the Inbound Application Databank is optional. Directory usage is controlled by a combination of trading profile and control card options. The calculation shown below assumes that all inbound EDI data received is being placed on the Inbound Application Databank Directory.

Approximate space requirements can be calculated based on the following formula:

**# of Records = # inbound EDI transactions mapped and placed on the Inbound Application Databank**

**Note:** In calculating the number of records, there are several equivalent methods of accounting for system activity, retention period, and cycle. For simplicity, the following calculations do not include these factors. System activity, retention period, and cycle must be factored into these calculations. The example shows one possible method.

### Example

An average of 150 Purchase Orders are received each day.

The Inbound Application Databank Maintenance program is run twice per week, and it is run with a Retention-Days-Loaded control card specifying 4 days.

- Calculate the average daily volumes.  
# of documents = 150 purchase orders  
Average # of records per day = 150
- Determine the number of records on file, given the retention options chosen. To simplify the calculation, assume that all documents are successfully processed within one day.  
Records created over retention period = 600 (150 \* 4 days)  
Records created between maintenance runs = 600 (150 \* 4 days)  
Records on file = 1200 (600 + 600)

## VSAM Data Set Definition

```

DEFINE          CLUSTER(NAME(@@@@@.EDI.EDIIAA          -
                 VOLUME(VVVVVV)                       -
                 RECORDS(1200 300)                     -
                 FREESPACE(40 20)                      -
                 KEYS(110 0)                           -
                 RECORDSIZE(800 800)                   -
                 SHR(2 3))                              -
                 DATA(NAME(@@@@@.EDI.EDIIAA.DATA)     -
                 CISZ(4096))                           -
                 INDEX(NAME(@@@@@.EDI.EDIIAA.INDEX))

```

## Inbound Application Databank Message Store

## EDIIAS

### Description

The Inbound Application Databank Message Store (EDIIAS) contains the application data as it is received by the user application from the Inbound Application Extract program. The application data on the Inbound Application Databank Message Store file can be viewed and optionally edited using CICS Databank Maintenance.

### Activity

This file is optional. If your installation maintains inbound application data at the message store-level, this file must be defined.

If used, this is a high activity operational file.

### Program References

The following programs reference the Inbound Application Databank Message Store:

Program	Purpose
Inbound Mapper	Updates the file
Inbound Application Databank Maintenance	Updates the file
Inbound Application Databank Extract	Updates the file
CICS Databank Maintenance	Reads the file

### Data Set Recovery

Before recovering files, you should review the online Change Audit and note any outstanding online updates (outstanding updates have an applied date and a time of zeros).

The Inbound Application Databank Directory, Message Store, Change Audit, Pending and Transaction Queue files, as well as the Inbound EDI Databank Link file are all closely related. If you need to recover the Inbound Application Databank Message Store, Inbound Application Databank Directory, and Inbound Application Databank Change Audit files, you must recover the Inbound EDI Databank Link file at the same time, using the same method.

**Note:** The Inbound Application Databank Pending and Inbound Application Databank Transaction Queue files must be re-initialized and any pending online changes re-applied.

The Inbound Application Databank Directory, Message Store, and Change Audit files and the Inbound EDI Databank Link file can be recreated from either a recent backup or re-loaded from the Gentran:Basic system tape.

## Disk Space Requirements

The exact space required depends on the level of system activity, the retention period specified, and the frequency of databank maintenance.

**Note:** Maintaining Message Store data on the Inbound Application Databank is optional. Message Store usage is controlled by a combination of trading profile and control card options. The calculation shown below assumes that all inbound application data received is being placed on the Inbound Application Databank Message Store.

Approximate space requirements can be calculated based on the following formula:

**# of Records = # of inbound application records**

**Note:** In calculating the number of records, there are several equivalent methods of accounting for system activity, retention period, and cycle. For simplicity, the following calculations do not include these factors. System activity, retention period and cycle must be factored into these calculations. The following example shows one possible method.

### Example

An average of 150 Purchase Orders are received each day.

The Inbound Application Databank Maintenance program is run twice per week, and it is run with a Retention-Days-Loaded control card specifying 4 days.

- Calculate the average daily volumes.
  - # of documents = 150 purchase orders
  - Average records generated by each purchase order = 75
  - Average # of records per day = 11,250 (# of documents \* average records/document)
- Determine the number of records on file, given the retention options chosen. To simplify the calculation assume that all documents are successfully processed within one day.
  - Records created over retention period = 45,000 (11,250 \* 4 days)
  - Records created between maintenance runs = 45,000 (11,250 \* 4 days)
  - Records on file = 90,000 (45,000 + 45,000)

### VSAM Data Set Definition

```

DEFINE          CLUSTER(NAME(@@@@@.EDI.EDIAS)          -
                VOLUME(VVVVVV)                        -
                RECORDS(45000 11250)                   -
                FREESPACE(00 00)                       -
                KEYS(16 0)                              -
                RECORDSIZE(250 5076)                   -
                SHR(2 3))                               -
                DATA(NAME(@@@@@.EDI.EDIAS.DATA)       -
                CISZ(8192))                             -
                INDEX(NAME(@@@@@.EDI.EDIAS.INDEX))
  
```

---

## Inbound Application Databank Pending File

**EDIAP**

### Description

The Inbound Application Databank Pending (EDIAP) file is used to present the most current view of the Inbound Application Databank based on CICS Databank Maintenance activity that has been performed.

### Activity

This is a low activity operational file.

### Program References

The following program references the Inbound Application Databank Pending file:

Program	Purpose
CICS Databank Maintenance	Updates the file

### Data Set Recovery

You can re-load this file from the Gentran:Basic system tape. If this file is recovered, Databank Maintenance does not reflect any online changes which were pending against the Inbound Application Databank until they have been applied in batch.

### Disk Space Requirements

The actual space requirements for the Inbound Application Databank Pending file vary. The requirements depend on the number of databank maintenance updates that are made to the Inbound Application Databank and how often the inbound Databank Extract program is run.

### Example

The Inbound Application Databank Extract program (EDID405) is run at midnight with the APPLY-UPDATES-ONLY parameter set to Yes. On a daily basis, two to five inbound application purchase orders are edited.

In this case, the Pending file holds the records until the Extract program is run at midnight. When the Extract program completes its processing, the pending records are removed.

**Note:** Twenty records have been allocated for the Pending file. This allocation should be sufficient to handle all situations, since on the average day only two records would be required.

**VSAM Data Set Definition**

```
DEFINE          CLUSTER(NAME(@@@@@.EDI.EDIAP)          -  
                VOLUME(VVVVVV)                        -  
                RECORDS(20 10)                        -  
                FREESPACE(20 10)                      -  
                KEYS(118 0)                            -  
                RECORDSIZE(2236 5400)                 -  
                SHR(2 3))                              -  
                DATA(NAME(@@@@@.EDI.EDIAP.DATA)      -  
                CISZ(8192))                            -  
                INDEX(NAME(@@@@@.EDI.EDIAP.INDEX)
```



## Inbound Application Databank Transaction Queue

**EDIQ095**

### Description

The Inbound Application Databank Transaction Queue (EDIQ095) contains change transactions for the Inbound Application Databank that users have created through databank maintenance. It is used to move change transactions between the CICS and batch environments. This file is initialized by running the Transaction Queue Initialize program.

### Activity

This is a low activity transaction file.

### Program References

The following programs reference the Inbound Application Databank Transaction Queue:

Program	Purpose
Transaction Queue Initialize	Updates the file
Inbound Application Databank Extract	Updates the file
CICS Databank Maintenance	Updates the file

### Data Set Recovery

Before recovering this file, review the online Change Audit and not any outstanding online updates for the Inbound Application Databank (outstanding updates have an applied date and a time of zeros).

The Inbound Application Databank Transaction Queue can be re-defined and initialized using the Transaction Queue Initialize program. Any online updates which were pending should be re-applied.

The Transaction Queue must not be recovered from a backup.

### Disk Space Requirements

The actual space requirements for the Inbound Application Databank Transaction Queue file vary. The requirements depend on the number of databank maintenance updates that are made to the Inbound Application Databank and how often the Inbound Application Databank Extract program is run. Each update requires one record.

### Example

The Inbound Application Databank Extract program is run each night at midnight. On Monday morning two inbound application purchase orders were edited and an inbound application invoice was deleted.

In this case, the Transaction Queue file would require three records until the Inbound Application Databank Extract program was run at midnight. When the Inbound Application Databank Extract program completes the Transaction Queue records are removed.

**VSAM Data Set Definition**

```
DEFINE          CLUSTER(NAME(@@@@@.EDI.EDIQ095)      -  
                VOLUME(VVVVVVV)                    -  
                RECORDS(100)                         -  
                NUMBERED                             -  
                RECORDSIZE(4088 4088)                -  
                SHR(4 3))                             -  
                DATA(NAME(@@@@@.EDI.EDIQ095.DATA)   -  
                CISZ(4096))                           -  
                INDEX(NAME(@@@@@.EDI.EDIQ095.INDEX))
```

**Note:** No secondary allocation can be defined because the file is formatted by a maintenance program that calculates the exact number of records in the file. This number is required for accurate processing. The maximum number of records that can be allocated when defining this transaction queue file is 32,700.

---

**Inbound EDI Databank Archive File****EDIIEAR****Description**

The Inbound EDI Databank Archive (EDIIEAR) file is a standard sequential data set that contains documents removed from the databank using the Databank Maintenance programs with the Archive Control option.

The Archive file is a standard sequential file with a maximum record length of 5082 (DSORG=PS,LRECL=5082,RECFM=VB).

**Activity**

The file volume depends on the number of documents being removed from the databank.

**Reference**

The following program references the Archive file:

<b>Program</b>	<b>Purpose</b>
Inbound EDI Databank Maintenance	Creates the data set (Purge or Housekeeping mode)
	Reads the data set (Restore mode)

**Disk Space Requirements**

The space requirements depend on databank activity. The Archive file contains images of the Inbound EDI Databank Directory and Message Store records deleted by databank maintenance.

## Inbound EDI Databank Change Audit File

**EDIIECA**

### Description

The Inbound EDI Databank Change Audit (EDIIECA) file contains detailed audit information for online updates which are applied, through databank maintenance, to the Inbound EDI Databank. This information is available for online or batch inquiry.

### Activity

This is a low activity operational file.

### Program References

The following programs reference the Inbound EDI Databank Change Audit file:

Program	Purpose
Inbound Editor	Updates the file
Databank Change Audit Inquiry	Reads the file
Databank Change Audit Maintenance	Updates the file
CICS Databank Maintenance	Reads the file

### Data Set Recovery

Before recovering files, you should review the online Change Audit to identify all outstanding updates (outstanding updates have an applied date and a time of zeros). These pending online changes must be re-applied after file recovery has been completed.

The Inbound EDI Databank Directory, Message Store, Change Audit, Pending and Transaction Queue files are all closely related. If you need to recover the Inbound EDI Databank Change Audit file, you must recover the Inbound EDI Databank Directory and Inbound EDI Databank Message Store files at the same time, using the same method.

**Note:** If you are **re-initializing** the Change Audit file, it is not necessary to re-initialize your Directory, Message, Store and Link files.

The Inbound EDI Databank Pending and Inbound EDI Databank Transaction Queue files must be re-initialized and any pending online changes re-applied.

The Inbound EDI Databank Directory, Message Store, and Change Audit files can be recreated from either a recent backup or re-loaded from the Gentran:Basic system tape.

## Disk Space Requirements

One Change Audit record is created for each update performed against the Inbound EDI Databank. Space utilization for this file is maintained with retention periods.

**Note:** In calculating the number of records, there are several equivalent methods of accounting for system activity, retention period, and cycle. For simplicity, the following calculations do not include these factors. System activity, retention period, and cycle must be factored into these calculations. The example shows one possible method.

### Example

An average of 2 updates are performed each day against the Inbound EDI Databank.

The Databank Change Audit Maintenance program is run once a month with a retention period of 30 days specified.

1. Calculate the average monthly volume.  
Updates per month = 60 (30 days \* 2 updates per day)
2. Determine the number of records on file, given the retention options chosen. To simplify the calculation assume that all updates are successfully processed within one day.  
Records created over retention period = 60  
Records created between maintenance runs = 60  
Records on file = 120 (60 + 60)

## VSAM Data Set Definition

```

DEFINE      CLUSTER(NAME(@@@@@.EDI.EDIIECA)      -
            VOLUME(VVVVVV)                      -
            RECORDS(120 30)                      -
            FREESPACE(30 10)                     -
            KEYS(116 0)                          -
            RECORDSIZE(2100 2100)                -
            SHR(2 3))                             -
            DATA(NAME(@@@@@.EDI.EDIIECA.DATA)  -
            CISZ(8192))                           -
            INDEX(NAME(@@@@@.EDI.EDIIECA.INDEX))

```

## Inbound EDI Databank Directory File

**EDIIEA**

### Description

The Inbound EDI Databank Directory (EDIIEA) contains status and control information for inbound EDI data. The Inbound EDI Databank Directory also maintains records for controlling data recovery.

### Activity

This is a high activity operational file.

### Program References

The following programs reference the Inbound EDI Databank Directory:

Program	Purpose
Inbound Editor	Updates the directory
Inbound EDI Databank Maintenance	Updates the directory
EDI Databank Inquiry	Reads, and optionally updates the directory
CICS Databank Maintenance	Reads the directory

### Data Set Recovery

Before recovering files, you should review the online Change Audit to identify all outstanding updates (outstanding updates have an applied date and a time of zeros). These pending online changes must be re-applied after file recovery has been completed.

The Inbound EDI Databank Directory, Message Store, Change Audit, Pending and Transaction Queue files are all closely related. If you need to recover the Inbound EDI Databank Directory, you must recover the Inbound EDI Databank Message Store and Inbound EDI Databank Change Audit files at the same time, using the same method. The Inbound EDI Databank Pending and Inbound EDI Databank Transaction Queue files must be re-initialized and any pending online changes re-applied.

The Inbound EDI Databank Directory, Message Store, and Change Audit files can be recreated from either a recent backup or re-loaded from the Gentran:Basic system tape.

## Disk Space Requirements

The exact space required depends on the level of system activity, the retention period specified, and the frequency of databank maintenance.

**Note:** Maintaining Directory data on the Inbound EDI Databank is optional. Directory use is controlled by a combination of trading profile and control card options. The calculation shown below assumes that all inbound EDI data received is being placed on the Inbound EDI Databank Directory.

Approximate space requirements can be calculated based on the following formula:

**# of Records = # interchanges + 2(# of functional groups) + # of transactions**

**Note:** In calculating the number of records, there are several equivalent methods of accounting for system activity, retention period, and cycle. For simplicity, the following calculations do not include these factors. System activity, retention period, and cycle must be factored into these calculations. The example shows one possible method.

### Example

Orders are received from 10 trading partners each day. An average of 150 purchase order transactions are received each day. An average of 10 purchase order change transactions are received each day. The trading partners send an average of 1 interchange a day, containing 15 purchase orders and 1 purchase order change.

The Inbound EDI Databank Maintenance program is run twice per week with a Retention-Days-Loaded control card specifying 4 days.

1. Calculate the average daily volumes.

Inbound interchanges = 10

Inbound functional groups = 20 (10 trading partners \* 2 functional groups/interchanges)

Inbound transactions = 160 (150 purchase orders + 10 purchase orders)

Average # of records created per day = 1 + (2 \* 20) + 160 = 210

2. Determine the number of records on file, given the retention options chosen. To simplify the calculation, assume that all documents are successfully processed within one day.

Records created over retention period = 840 (210 \* 4 days)

Records created between maintenance runs = 840 (210 \* 4 days)

Records on file = 1680 (840 + 840)

## VSAM Data Set Definition

```

DEFINE      CLUSTER(NAME(@@@@@.EDI.EDIIEA)          -
            VOLUME(VVVVVV)                          -
            RECORDS(1680 420)                        -
            FREESPACE(40 20)                         -
            KEYS(125 0)                              -
            RECORDSIZE(800 800)                      -
            SHR(2 3))                                -
            DATA(NAME(@@@@@.EDI.EDIIEA.DATA)       -
            CISZ(4096))                              -
            INDEX(NAME(@@@@@.EDI.EDIIEA.INDEX))

```

## Inbound EDI Databank Link File

**EDIIEI**

### Description

The Inbound EDI Databank Link file (EDIIEI) contains control information used to connect each EDI transaction on the Inbound EDI Databank to the same document on the Inbound Application Databank.

### Activity

This file is relatively stable. On-going access is for connection purposes only.

### Program References

The following programs reference the Inbound EDI Databank Link file:

Program	Purpose
Inbound Mapper	Updates the file
Inbound Application Databank Maintenance	Updates the file
CICS Databank Maintenance	Reads the directory

### Data Set Recovery

Before recovering files, you should review the online Change Audit to identify all outstanding updates (outstanding updates have an applied date and a time of zeros). These pending online changes must be re-applied after file recovery has been completed.

The Inbound EDI Databank Link file is closely related to the Inbound Application Databank Directory, Message Store, Change Audit, Pending and Transaction Queue files. If you need to recover the Inbound EDI Databank Link file, you must recover the Inbound Application Databank Directory and the Inbound Application Databank Message Store and Inbound Application Databank Change Audit files at the same time, using the same method. Also, the Inbound Application Databank Pending and Inbound Application Databank Transaction Queue files must be re-initialized and any pending online changes re-applied once the files have been recovered.

The Inbound EDI Databank Link file, and the Inbound Application Databank Directory, Message Store, and Change Audit files can be recreated from either a recent backup or re-loaded from the Gentran:Basic system tape.

### Disk Space Requirements

The disk space requirements for this file are closely related to the Inbound Application Databank Directory. One record needs to be allocated for each document on the Inbound Application Databank for which the same transaction resides on the Inbound EDI Databank.



---

**VSAM Data Set Definition**

```
DEFINE      CLUSTER(NAME(@@@@@.EDI.EDIHEL)          -  
            VOLUME(VVVVVV)                          -  
            RECORDS(1200 300)                        -  
            FREESPACE(40 10)                         -  
            KEYS(126 0)                              -  
            RECORDSIZE(237 237)                      -  
            SHR(2 3))                                 -  
            DATA(NAME(@@@@@.EDI.EDIHEL.DATA)        -  
            CISZ(4096))                              -  
            INDEX(NAME(@@@@@.EDI.EDIHEL.INDEX))      -  
            CISZ(10240))
```

---

## Inbound EDI Databank Message Store File

**EDIIES**

### Description

The Inbound EDI Databank Message Store (EDIIES) file contains the actual EDI data in its entirety as it was received from your trading partner. The EDI data on the Inbound EDI Databank Message Store file can be viewed using CICS Databank Maintenance.

### Activity

This is a high activity operational file, if Message Store level data is maintained by your installation.

### Program References

The following programs reference the Inbound EDI Databank Message Store:

Program	Purpose
Inbound Editor	Updates the file
Inbound EDI Databank Maintenance	Updates the file
CICS Databank Inquiry	Reads the file

### Data Set Recovery

Before recovering files, you should review the online Change Audit to identify all outstanding updates (outstanding updates have an applied date and a time of zeros). These pending online changes must be re-applied after file recovery has been completed.

The Inbound EDI Databank Directory, Message Store, Change Audit, Pending and Transaction Queue files are all closely related. If you need to recover the Inbound EDI Databank Message Store file, you must recover the Inbound EDI Databank Directory and Inbound EDI Databank Change Audit files at the same time, using the same method. The Inbound EDI Databank Pending and Inbound EDI Databank Transaction Queue files must be re-initialized and any pending online changes re-applied.

The Inbound EDI Databank Directory, Message Store, and Change Audit files can be recreated from either a recent backup or re-loaded from the Gentran:Basic system tape.

---

## Disk Space Requirements

The exact space required depends on the level of system activity, the retention period specified, and the frequency of databank maintenance.

**Note:** Maintaining Message Store data on the Inbound EDI Databank is optional. Message Store usage is controlled by a combination of trading profile and control card options. The calculation shown below assumes that all inbound EDI data received is being placed on the Inbound EDI Databank Message Store.

Approximate space requirements can be calculated based on the following formula:

**# of Records = # of EDI transactions**

**Note:** In calculating the number of records, there are several equivalent methods of accounting for system activity, retention period, and cycle. For simplicity, the following calculations do not include these factors. System activity, retention period, and cycle must be factored into these calculations. The example shows one possible method.

### Example

An average of 150 Purchase Orders are received each day, on an average of 50 orders per run. The average size of a purchase order is 2700 characters.

The Inbound EDI Databank Maintenance program is run twice per week, and it is run with a Retention-Days-Loaded control card specifying 4 days.

1. Calculate the average daily volumes:  
# of transactions = 150 purchase orders  
average char./transaction = 2700  
average total characters/day = 405,000 (# of transactions \* average char./transaction)
2. Determine the average number of records created per day, based on the maximum record length of 4000:  
average # of records created per day = 101 (average total characters / 4000)
3. Determine the number of records on file, given the retention options chosen. To simplify the calculation assume that all documents are successfully processed within one day.  
Records created over retention period = 404 (101 \* 4 days)  
Records created between maintenance runs = 404 (101 \* 4 days)  
Records on file = 808 (404 + 404)

**VSAM Data Set Definition**

```
DEFINE          CLUSTER(NAME(@@@@@.EDI.EDIIES)      -
                VOLUME(VVVVVV)                    -
                RECORDS(808 202)                    -
                FREESPACE(00 00)                    -
                KEYS(16 0)                           -
                RECORDSIZE(2050 4050)                -
                SHR(2 3))                             -
                DATA(NAME(@@@@@.EDI.EDIIES.DATA)    -
                CISZ(4096))                           -
                INDEX(NAME(@@@@@.EDI.EDIIES.INDEX))
```

**Inbound EDI Databank Pending File****EDIIEP****Description**

The Inbound EDI Databank Pending (EDIIEP) file presents the most current view of the Inbound EDI Databank, based on CICS Databank Maintenance activity that has been performed.

**Activity**

This is a low activity operational file.

**Program References**

The following program references the Inbound EDI Databank Pending file:

Program	Purpose
CICS Databank Maintenance	Updates the file

**Data Set Recovery**

This file can be re-loaded from the Gentran:Basic system tape. If this file is recovered Databank Maintenance inquiries do not reflect Databank Maintenance changes that were made between the time the recovery backup was created and the time the file was recreated. However, these changes are still applied to the Inbound EDI Databank.

**Disk Space Requirements**

The actual space requirements for the Inbound EDI Databank Pending file vary. The requirements depend on the number of Databank Maintenance updates that are made to the Inbound EDI Databank and how often the Inbound Editor is run with a control card of INBOUND EDI DATABANK REPROCESS specified. Each update requires one record.

**Example**

The Inbound Editor is run each night at midnight with the global parameter INBOUND EDI DATABANK REPROCESS specified. On Monday morning one inbound EDI interchange was deleted from the databank and two inbound EDI purchase orders were edited.

In this case, the Pending file would require three records until the Inbound Editor was run at midnight. When the Inbound Editor completes the pending records are removed.

**VSAM Data Set Definition**

```

DEFINE          CLUSTER(NAME(@@@@@.EDI.EDIIEP)          -
                VOLUME(VVVVVV)                        -
                RECORDS(20 10)                         -
                FREESPACE(20 10)                       -
                KEYS(127 0)                             -
                RECORDSIZE(527 2500)                   -
                SHR(2 3))                               -
                DATA(NAME(@@@@@.EDI.EDIIEP.DATA)     -
                CISZ(4096))                             -
                INDEX(NAME(@@@@@.EDI.EDIIEP.INDEX))

```

## Inbound EDI Databank Transaction Queue

EDIQ097

### Description

The Inbound EDI Databank Transaction Queue (EDIQ097) contains change transactions for the Inbound EDI Databank that users have created through databank maintenance. It is used to move change transactions between the CICS and batch environments. This file is initialized by running the Transaction Queue Initialize program.

### Activity

This is a low activity transaction file.

### Program References

The following programs reference the Inbound EDI Databank Transaction Queue:

Program	Purpose
Transaction Queue Initialize	Updates the file
Inbound Editor	Updates the file
CICS Databank Maintenance	Updates the file

### Data Set Recovery

Before recovering this file, the online Change Audit should be reviewed and any outstanding online updates for the Inbound EDI Databank noted (outstanding updates have an applied date and a time of zeros).

The Inbound EDI Databank Transaction Queue can be re-defined and initialized using the Transaction Queue Initialize program. Any online updates which were pending should be re-applied.

**The Transaction Queue must not be recovered from a backup.**

### Disk Space Requirements

The actual space requirements for the Inbound EDI Databank Transaction Queue file vary. The requirements depend on the number of Databank Maintenance updates that are made to the Inbound EDI Databank, and how often the Inbound Editor is run with a global parameter of INBOUND EDI DATABANK REPROCESS specified. Each update requires one record.

### Example

The Inbound Editor is run each night at midnight with a global parameter of INBOUND EDI DATABANK REPROCESS. On Monday morning one inbound EDI interchange was deleted from the databank and two inbound EDI purchase orders were edited.

In this case, the Transaction Queue file would require three records until the Inbound Editor was run at midnight. When the Inbound Editor completes, the Transaction Queue records are removed.

---

**VSAM Data Set Definition**

```
DEFINE          CLUSTER(NAME(@@@@@.EDI.EDIQ097)      -  
                VOLUME(VVVVVVV)                    -  
                RECORDS(100)                         -  
                NUMBERED                             -  
                RECORDSIZE(4088 4088)                -  
                SHR(4 3))                             -  
                DATA(NAME(@@@@@.EDI.EDIQ097.DATA)  -  
                CISZ(4096))                           -
```

**Note:** No secondary allocation can be defined because the file is formatted by a maintenance program that calculates the exact number of records in the file. This number is required for accurate processing. The maximum number of records that can be allocated when defining this transaction queue file is 32,700.

## Network Reconciliation File

EDINRC

### Description

The Network Reconciliation (EDINRC) file stores expected network status information records.

### Activity

This is an operational file which normally has low activity.

### Program References

The following programs reference the Network Reconciliation file:

Program	Purpose
Network Reconciliation	Updates the file
Network Reconciliation Maintenance	Updates the file
Outbound Editor	Updates the file
Outbound EDI Databank Extract	Updates the file

### Data Set Recovery

The Network Reconciliation file can be re-created by restoring from the most recent backup.

### Recovery/Restart Considerations

For file **recovery** purposes, a backup of the Network Reconciliation file must be completed before running the above programs: Network Reconciliation, Network Reconciliation Maintenance, Outbound Editor, and Outbound EDI Databank Extract. **Restart** of these programs requires the user to restore the Network Reconciliation file using the backup created prior to the run and rerun the program.

### Disk Space Requirements

The space required for the Network Reconciliation file depends on the volume of outbound interchanges for which network tracking has been specified, the retention period selected, and the frequency with which the Network Reconciliation Maintenance program is run.

A guideline to use is one record for each outbound interchange that is being tracked.

### VSAM Data Set Definition

```

DEFINE          CLUSTER(NAME(@@@@@.EDI.EDINRC)          -
                VOLUME(VVVVVV)                          -
                RECORDS(100 100)                         -
                FREESPACE (10 10)                        -
                KEYS(130 0)                               -
                RECORDSIZE(300 300)                       -
                SHR(2 3))                                 -
                DATA(NAME(@@@@@.EDI.EDINRC.DATA)        -
                CISZ(4096))                               -
                INDEX(NAME(@@@@@.EDI.EDINRC.INDEX))

```



## Network Response File

**EDINR**

### Description

The Network Response (EDINR) file is used as a temporary sequential file to collect network response/status information for input to the specific network's Response Processor (such as the Commerce:Network reports, GEIS Electronic Format reports, IBM IN Audit reports) or the generic Universal Response Processor (EDID825).

The Network Response file is a standard sequential file (DSORG=PS) with a record length that varies with the originating network as follows:

<b>Network</b>	<b>Record Length (LRECL)</b>
Commerce:Network	80
GEIS	80
IBM IN	254
Universal (generic)	350

### Activity

This is an operational file which normally has low activity.

### Program References

The following programs reference the Network Response file:

<b>Program</b>	<b>Purpose</b>
Commerce:Network Response Processor Program	Reads the file
GEIS Response Processor Program	Reads the file
IBM IN Response Processor Program	Reads the file
Universal Response Processor Program	Reads the file

### Data Set Recovery

The Network Response file can be re-created by either running the appropriate network report or by re-retrieving the report/data file from the specific network.

### Disk Space Requirements

The space requirements for the Network Response file depends on the anticipated network activity and the frequency of retrieving network reports.

TRACKS(10,5) addresses most requirements.

**Outbound Application Databank Archive File****EDIOAAR****Description**

The Outbound Application Databank Archive (EDIOAAR) file is a standard sequential data set that contains documents removed from the Databank using the Outbound Application Databank Maintenance program with the Archive control option.

The Archive file is a standard sequential file with a maximum record length of 5082 (DSORG=PS,LRECL=5082,RECFM=VB).

**Activity**

The Archive file is a transaction file. Its volume depends on the number of documents being removed from the Databank.

**Program Reference**

The following program references the Archive file:

<b>Program</b>	<b>Purpose</b>
Outbound Application Databank Maintenance	Creates the data set (Purge or Housekeeping mode) Reads the data set (Restore mode)

**Disk Space Requirements**

The space requirements depend on databank activity. The Archive file contains images of the Databank Directory and Databank Message Store records deleted by databank maintenance.

**Outbound Application Databank Change Audit File****EDIOACA****Description**

The Outbound Application Databank Change Audit (EDIOACA) file contains detailed audit information of all online updates which are applied to the Outbound Application Databank. This information is available for online or batch inquiry.

**Activity**

This is a low activity file.

**Program References**

The following programs reference the Outbound Application Databank Change Audit file:

<b>Program</b>	<b>Purpose</b>
Outbound Mapper	Updates the file
Databank Change Audit Inquiry	Reads the file
Databank Change Audit Maintenance	Updates the file
CICS Databank Maintenance	Reads the file

**Data Set Recovery**

Before any files are recovered, the online Change Audit should be reviewed and any outstanding online updates for the Outbound Application Databank noted (outstanding updates have an applied date and a time of zeros).

The Outbound Application Databank Directory, Message Store, Change Audit, Pending and Transaction Queue files are all closely related. If you need to recover the Outbound Application Databank Directory, you must recover the Outbound Application Databank Message Store and Outbound Application Databank Change Audit files at the same time, using the same method.

**Note:** If you are **re-initializing** the Change Audit file, it is not necessary to re-initialize your Directory, Message, Store and Link files.

The Outbound Application Databank Pending and Outbound Application Databank Transaction Queue files must be re-initialized and any pending online changes re-applied.

The Outbound Application Databank Directory, Message Store, and Change Audit files can be recreated from either a recent backup or re-loaded from the Gentran:Basic system tape.

## Disk Space Requirements

One Change Audit record is created for each update performed against the Outbound Application Databank. Space utilization for this file is maintained with retention periods.

**Note:** In calculating the number of records, there are several equivalent methods of accounting for system activity, retention period, and cycle. For simplicity, the following calculations do not include these factors. System activity, retention period, and cycle must be factored into these calculations. The example shows one possible method.

### Example

An average of two updates are performed each day against the Outbound Application Databank.

The Databank Change Audit Maintenance program is run once a month with a retention period of 30 days specified.

1. Calculate the average monthly volume.  
Updates per month = 60 (30 days \* 2 updates per day)
2. Determine the number of records on file, given the retention options chosen. To simplify the calculation assume that all updates are successfully processed within one day.  
Records created over retention period = 60  
Records created between maintenance runs = 60  
Records on file = 120 (60 + 60)

## VSAM Data Set Definition

```

DEFINE          CLUSTER(NAME(@@@@@.EDI.EDIOACA)      -
                VOLUME(VVVVVV)                      -
                RECORDS(120 30)                      -
                FREESPACE(30 10)                     -
                KEYS(116 0)                           -
                RECORDSIZE(1700 2100)                 -
                SHR(2 3))                             -
                DATA(NAME(@@@@@.EDI.EDIOACA.DATA)   -
                CISZ(4096))                           -
                INDEX(NAME(@@@@@.EDI.EDIOACA.INDEX))

```

## Outbound Application Databank Directory

EDIOAA

### Description

The Outbound Application Databank Directory (EDIOAA) contains status and control information for outbound application data. The Outbound Application Databank Directory also maintains records for controlling data recovery.

### Activity

This is a high activity operational file.

### Program References

The following programs reference the Outbound Application Databank Directory:

Program	Purpose
Outbound Mapper	Updates the directory
Outbound Application Databank Maintenance	Updates the directory
Application Databank Inquiry	Reads, and optionally updates the directory
CICS Databank Maintenance	Reads the directory

### Data Set Recovery

Before any files are recovered, the online Change Audit should be reviewed and any outstanding online updates for the Outbound Application Databank noted (outstanding updates have an applied date and a time of zeros).

The Outbound Application Databank Directory, Message Store, Change Audit, Pending and Transaction Queue files are all closely related. If you need to recover the Outbound Application Databank Directory, you must recover the Outbound Application Databank Message Store and Outbound Application Databank Change Audit files at the same time, using the same method. The Outbound Application Databank Pending and Outbound Application Databank Transaction Queue files must be re-initialized and any pending online changes re-applied.

The Outbound Application Databank Directory, Message Store, and Change Audit files can be recreated from either a recent backup or re-loaded from the Gentran:Basic system tape.

## Disk Space Requirements

The exact space required depends on the level of system activity, the retention period specified, and the frequency of databank maintenance. One record is required for each document on the Outbound Application Databank.

**Note:** Maintaining Directory data on the Outbound Application Databank is optional. Directory usage is controlled by a combination of trading profile and control card options. The calculation shown below assumes that all outbound application data being sent is being placed on the Outbound Application Databank Directory.

Approximate space requirements can be calculated based on the following formula:

**# of Records = # outbound application documents mapped and placed on the Outbound Application Databank**

**Note:** In calculating the number of records, there are several equivalent methods of accounting for system activity, retention period, and cycle. For simplicity, the following calculations do not include these factors. System activity, retention period, and cycle must be factored into these calculations. The example shows one possible method.

### Example

An average of 150 Purchase Orders and 20 Invoices are sent each day.

The Outbound Application Databank Maintenance program is run twice per week, and it is run with a Retention-Days-Loaded control card specifying 4 days.

1. Calculate the average daily volumes.  
Average outbound documents = 170 (150 + 20)  
Average # of records created per day = 170
2. Determine the number of records on file, given the retention options chosen. To simplify the calculation assume that all documents are successfully processed within one day.  
Records created over retention period = 680 (170 \* 4 days)  
Records created between maintenance runs = 680 (170 \* 4 days)  
Records on file = 1360 (680 + 680)

### VSAM Data Set Definition

```

DEFINE          CLUSTER(NAME(@@@@@.EDI.EDIOAA)          -
                VOLUME(VVVVVV)                          -
                RECORDS(1360 340)                        -
                FREESPACE(40 20)                         -
                KEYS(110 0)                               -
                RECORDSIZE(800 800)                     -
                SHR(2 3))                                 -
                DATA(NAME(@@@@@.EDI.EDIOAA.DATA)        -
                CISZ(4096))                              -
                INDEX(NAME(@@@@@.EDI.EDIOAA.INDEX))

```

## Outbound Application Databank Link File

EDIOAL

### Description

The Outbound Application Databank Link (EDIOAL) file contains control information used to connect each application document on the Outbound Application Databank to the same transaction on the Outbound EDI Databank.

### Activity

This file is relatively stable. On-going accessibility are for connection purposes only.

### Program References

The following programs reference the Outbound Application Databank Link file:

Program	Purpose
Outbound Editor	Updates the file
Outbound EDI Databank Extract	Updates the file
Outbound EDI Databank Maintenance	Updates the file
CICS Databank Maintenance	Reads the file

### Data Set Recovery

Before any files are recovered, the online Change Audit should be reviewed and any outstanding online updates noted (outstanding updates have an applied date and a time of zeros).

The Outbound Application Databank Link file is closely related to the Outbound EDI Databank Directory, Message Store, Change Audit, Pending and Transaction Queue files. If you need to recover the Outbound Application Databank Link file, you must recover the Outbound EDI Databank Directory, the Outbound EDI Databank Message Store and Outbound EDI Databank Change Audit files at the same time, using the same method. The Outbound EDI Databank Pending and Outbound EDI Databank Transaction Queue files must be re-initialized and any pending online changes re-applied.

The Outbound Application Databank Link file, and the Outbound EDI Databank Directory, Message Store, and Change Audit files can be recreated from either a recent backup or re-loaded from the Gentran:Basic system tape.

### Disk Space Requirements

The disk space requirements for this file are closely related to the Outbound EDI Databank Directory. One record needs to be allocated for each transaction on the Outbound EDI Databank for which the same document resides on the Outbound Application Databank.

**VSAM Data Set Definition**

```
DEFINE          CLUSTER(NAME(@@@@@.EDI.EDIOAL)      -  
                VOLUME(VVVVVV)                    -  
                RECORDS(1000 100)                  -  
                FREESPACE(30 10)                   -  
                KEYS(111 0)                         -  
                RECORDSIZE(237 237)                -  
                SHR(2 3))                           -  
DATA(NAME(@@@@@.EDI.EDIOAL.DATA)                  -  
      CISZ(4096))                                   -  
INDEX(NAME(@@@@@.EDI.EDIOAL.INDEX))               -  
      CISZ(6144))                                   -
```



## Outbound Application Databank Message Store

EDIOAS

### Description

The Outbound Application Databank Message Store (EDIOAL) contains the application data in the same format as it was received from the application. The application data on the Outbound Application Databank Message Store file can be viewed and optionally edited using CICS Databank Maintenance.

### Activity

This file is optional. If your installation maintains outbound application data at the message store level this file must be defined.

If used, this is a high activity operational file.

### Program References

The following programs reference the Outbound Application Databank Message Store:

Program	Purpose
Outbound Mapper	Updates the file
Outbound Application Databank Maintenance	Updates the file
CICS Databank Maintenance	Updates the file

### Data Set Recovery

Before any files are recovered, the online Change Audit should be reviewed and any outstanding online updates for the Outbound Application Databank noted (outstanding updates have an applied date and a time of zeros).

The Outbound Application Databank Directory, Message Store, Change Audit, Pending and Transaction Queue files are all closely related. If you need to recover the Outbound Application Databank Message, you must recover the Outbound Application Databank Directory and Outbound Application Databank Change Audit files at the same time, using the same method. The Outbound Application Databank Pending and Outbound Application Databank Transaction Queue files must be re-initialized and any pending online changes re-applied.

The Outbound Application Databank Directory, Message Store, and Change Audit files can be recreated from either a recent backup or re-loaded from the Gentran:Basic system tape.

## Disk Space Requirements

The exact space required depends on the level of system activity, the retention period specified, and the frequency of databank maintenance.

**Note:** Maintaining Message Store data on the Outbound Application Databank is optional. Message Store usage is controlled by a combination of trading profile and control card options. The calculation shown below assumes that all outbound application data being sent is being placed on the Outbound Application Databank Message Store.

Approximate space requirements can be calculated based on the following formula:

**# of Records = # of outbound application document records**

**Note:** In calculating the number of records, there are several equivalent methods of accounting for system activity, retention period, and cycle. For simplicity, the following calculations do not include these factors. System activity, retention period, and cycle must be factored into these calculations. The example shows one possible method.

### Example

An average of 150 Purchase Orders are sent each day.

The Outbound Application Databank Maintenance program is run twice per week, and it is run with a Retention-Days-Loaded control card specifying 4 days.

1. Calculate the average daily volumes.  
 # of documents = 150 purchase orders  
 Average records/documents = 75  
 Average # of records per day = 11,250 (# of documents \* average records/document)
2. Determine the number of records on file, given the retention options chosen. To simplify the calculation assume that all documents are successfully processed within one day.  
 Records created over retention period = 45,000 (11,250 \* 4 days)  
 Records created between maintenance runs = 45,000 (11,250 \* 4 days)  
 Records on file = 90,000 (45,000 + 45,000)

## VSAM Data Set Definition

```

DEFINE                                CLUSTER(NAME(@@@@@.EDI.EDIOAS)      -
                                       VOLUME(VVVVVV)                -
                                       RECORDS(90000 22500)           -
                                       FREESPACE(00 00)                -
                                       KEYS(16 0)                      -
                                       RECORDSIZE(250 5076)           -
                                       SHR(2 3))                      -
                                       DATA(NAME(@@@@@.EDI.EDIOAS.DATA) -
                                       CISZ(8192))                    -
                                       INDEX(NAME(@@@@@.EDI.EDIOAS.INDEX))

```

**Outbound Application Databank Pending File****EDIOAP****Description**

The Outbound Application Databank Pending (EDIOAP) file presents the most current view of the Outbound Application Databank based on CICS Databank Maintenance activity that has been performed.

**Activity**

This is a low activity operational file.

**Program References**

The following programs reference the Outbound Application Databank Pending file:

Program	Purpose
CICS Databank Maintenance	Updates the file

**Data Set Recovery**

This file can be re-loaded from the Gentran:Basic system tape. If this file is recovered databank maintenance does not reflect any online changes which were pending against the Outbound Application Databank until they have been applied in batch.

**Disk Space Requirements**

The actual space requirements for the Outbound Application Databank Pending file vary. The requirements depend on the number of databank maintenance updates that are made to the Outbound Application Databank and how often the Outbound Mapper is run with a reprocess parameter of Y specified. Each update requires one record.

**Example**

The Outbound Mapper is run each night at midnight with a reprocess parameter of Y specified. On Monday morning two outbound application invoices were deleted.

In this case, the Pending file would require two records until the Outbound Mapper was run at midnight. When the Outbound Mapper completes the pending records are removed.

**VSAM Data Set Definition**

```

DEFINE          CLUSTER(NAME(@@@@@.EDI.EDIOAP)      -
                VOLUME(VVVVVVV)                   -
                RECORDS(20 10)                       -
                FREESPACE(20 10)                     -
                KEYS(118 0)                           -
                RECORDSIZE(2236 2500)                 -
                SHR(2 3))                             -
                DATA(NAME(@@@@@.EDI.EDIOAP.DATA)    -
                CISZ(4096))                           -
                INDEX(NAME(@@@@@.EDI.EDIOAP.INDEX))

```

## Outbound Application Databank Transaction Queue

EDIQ093

### Description

The Outbound Application Databank Transaction Queue (EDIQ093) contains change transactions for the Outbound Application Databank that users have created through databank maintenance. It is used to move change transactions between the CICS and batch environments. This file is initialized by running the Transaction Queue Initialize program.

### Activity

This is a low activity transaction file.

### Program References

The following programs reference the Outbound Application Databank Transaction Queue:

Program	Purpose
Transaction Queue Initialize	Updates the file
Outbound Mapper	Updates the file
CICS Databank Maintenance	Updates the file

### Data Set Recovery

Before recovering this file, the online Change Audit should be reviewed and any outstanding online updates for the Outbound Application Databank noted (outstanding updates have an applied date and a time of zeros).

The Outbound Application Databank Transaction Queue can be re-defined and initialized using the Transaction Queue Initialize program. Any online updates which were pending should be re-applied.

**Do not recover the Outbound Application Transaction Queue from a backup.**

### Disk Space Requirements

The actual space requirements for the Outbound Application Databank Transaction Queue file vary. The requirements depend on the number of Databank Maintenance updates that are made to the Outbound Application Databank and how often the Outbound Mapper is run with a reprocess parameter of Y specified. Each update requires one record.

### Example

The Outbound Mapper is run each night at midnight with a reprocess parameter of Y specified. On Monday morning two outbound application invoices were edited and an outbound application invoice was deleted.

In this case, the Transaction Queue file requires three records until the Outbound Mapper was run at midnight. When the Outbound Mapper completes the Transaction Queue records are removed.

---

**VSAM Data Set Definition**

```
DEFINE          CLUSTER(NAME(@@@@@.EDI.EDIQ093)      -  
                VOLUME(VVVVVVV)                   -  
                RECORDS(100)                         -  
                NUMBERED                             -  
                RECORDSIZE(4088 4088)                -  
                SHR(4 3))                            -  
                DATA(NAME(@@@@@.EDI.EDIQ093.DATA)  -  
                CISZ(4096))                           -
```

**Note:** No secondary allocation can be defined because the file is formatted by a maintenance program that calculates the exact number of records in the file. This number is required for accurate processing. The maximum number of records that can be allocated when defining this transaction queue file is 32,700.

## Outbound EDI Databank Archive File

**EDIOEAR**

### Description

The Outbound EDI Databank Archive (EDIOEAR) file is a standard sequential data set that contains documents removed from the databank using the Outbound EDI Databank Maintenance program with the Archive Control option.

The Archive file is a standard sequential file with a maximum record length of 5082 (DSORG=PS,LRECL=5082,RECFM=VB).

### Activity

The Archive file is a transaction file. Its volume depends on the number of documents being removed from the databank.

### Program Reference

The following programs reference the Archive file:

Program	Purpose
Outbound EDI Databank Maintenance	Creates the data set (Purge or Housekeeping mode)
	Reads the data set (Restore mode)

### Disk Space Requirements

The space requirements depend on databank activity. The Archive file contains images of the Databank Directory and Databank Message Store records deleted by databank maintenance.

## Outbound EDI Databank Change Audit File

EDIOECA

### Description

The Outbound EDI Databank Change Audit (EDIOECA) file contains detailed audit information of all online updates which are applied to the Outbound EDI Databank. This information is available for online or batch inquiry.

### Activity

This is a low activity operational file.

### Program References

The following programs reference the Outbound EDI Databank Change Audit file:

Program	Purpose
Outbound EDI Databank Extract	Updates the file
Databank Change Audit Maintenance	Updates the file
Databank Change Audit Inquiry	Reads the file
CICS Databank Maintenance	Reads the file

### Data Set Recovery

Before any files are recovered, the online Change Audit should be reviewed and any outstanding online Updates for the Outbound EDI Databank noted (outstanding updates have an applied date and a time of zeros).

The Outbound EDI Databank Directory, Message Store, Change Audit, Pending and Transaction Queue files, as well as the Outbound Application Databank Link file are all closely related. If you need to recover the Outbound EDI Databank Change Audit, you must recover the Outbound EDI Databank Directory and Outbound Application Databank Message Store files, as well as the Outbound Application Databank Link file at the same time, using the same method.

**Note:** If you are **re-initializing** the Change Audit file, it is not necessary to re-initialize your Directory, Message, Store and Link files.

The Outbound EDI Databank Pending and Outbound EDI Databank Transaction Queue files must be re-initialized and any pending online changes re-applied.

The Outbound EDI Databank Directory, Message Store, and Change Audit files, as well as the Outbound Application Databank Link file can be recreated from either a recent backup or re-loaded from the Gentran:Basic system tape.

## Disk Space Requirements

One Change Audit record is created for each update performed against the Outbound EDI Databank. Space utilization for this file is maintained with retention periods.

**Note:** In calculating the number of records, there are several equivalent methods of accounting for system activity, retention period, and cycle. For simplicity, the following calculations do not include these factors. System activity, retention period, and cycle must be factored into these calculations. The example shows one possible method.

### Example

An average of two updates are performed each day against the Outbound EDI Databank.

The Databank Change Audit Maintenance program is run once a month with a retention period of 30 days specified.

1. Calculate the average monthly volume.  
Updates per month = 60 (30 days \* 2 updates per day)
2. Determine the number of records on file, given the retention options chosen. To simplify the calculation assume that all updates are successfully processed within one day.  
Records created over retention period = 60  
Records created between maintenance runs = 60  
Records on file = 120 (60 + 60)

## VSAM Data Set Definition

```

DEFINE          CLUSTER(NAME(@@@@@.EDI.EDIOECA)      -
                VOLUME(VVVVVVV)                    -
                RECORDS(120 30)                      -
                FREESPACE(30 10)                     -
                KEYS(116 0)                           -
                RECORDSIZE(2100 2100)                -
                SHR(2 3))                             -
                DATA(NAME(@@@@@.EDI.EDIOECA.DATA)   -
                CISZ(4096))                           -
                INDEX(NAME(@@@@@.EDI.EDIOECA.INDEX))

```



## Outbound EDI Databank Directory

EDIOEA

### Description

The Outbound EDI Databank Directory (EDIOEA) contains status and control information for outbound EDI data. The Outbound EDI Databank Directory also maintains records for controlling data recovery.

### Activity

This is a high activity file.

### Program References

The following programs reference the Outbound EDI Databank Directory:

Program	Purpose
Outbound Editor	Updates the file
Outbound EDI Databank Extract	Updates the file
Outbound EDI Databank Maintenance	Updates the file
Acknowledgment Reconciliation/Monitor	Updates the file.
EDI Databank Inquiry	Reads and optionally updates the file
Network Reconciliation	Updates the file
CICS Databank Maintenance	Reads the file

### Data Set Recovery

Before any files are recovered, the online Change Audit should be reviewed and any outstanding online updates for the Outbound EDI Databank noted (outstanding updates have an applied date and a time of zeros).

The Outbound EDI Databank Directory, Message Store, Change Audit, Pending and Transaction Queue files, as well as the Outbound Application Databank Link file are all closely related. If you need to recover the Outbound EDI Databank Directory, you must recover the Outbound EDI Databank Message Store and Outbound EDI Databank Change Audit files, as well as the Outbound Application Databank Link file at the same time, using the same method. The Outbound EDI Databank Pending and Outbound EDI Databank Transaction Queue files must be re-initialized and any pending online changes re-applied.

The Outbound EDI Databank Directory, Message Store, and Change Audit files, and Outbound Application Databank Link file can be recreated from either a recent backup or re-loaded from the Gentran:Basic system tape.

## Disk Space Requirements

The exact space required depends on the level of system activity, the retention period specified, and the frequency of databank maintenance.

**Note:** Maintaining Directory data on the Outbound EDI Databank is optional. Directory usage is controlled by a combination of trading profile and control card options. The calculation shown below assumes that all outbound EDI data sent is being placed on the Outbound EDI Databank Directory.

Space requirements for the Outbound EDI Databank Directory depend on whether deferred enveloping and/or nondeferred enveloping is implemented. Approximate space requirements can be calculated based on the following formula:

$$\begin{aligned}
 \text{Total \# of records} &= \text{\# of records from nondeferred processing} \\
 &\quad + \text{\# of records from deferred processing} \\
 \\
 \text{\# of records from nondeferred processing} &= \text{\# of interchanges} \\
 &\quad + 2 (\text{\# of functional groups}) \\
 &\quad + \text{\# of transactions} \\
 \\
 \text{\# of records from deferred processing} &= \text{\# of interchanges} \\
 &\quad + 2(\text{\# of functional groups}) \\
 &\quad + 2(\text{\# of transactions}) \\
 &\quad + 3(\text{\# of deferred partners})
 \end{aligned}$$

**Note:** In calculating the number of records, there are several equivalent methods of accounting for system activity, retention period, and cycle. For simplicity, the following calculations do not include these factors. System activity, retention period, and cycle must be factored into these calculations. The example shows one possible method.

### Example

An average of 100 purchase orders are sent each day. These orders are sent to five different trading partners, four of which are deferred. Each deferred partner receives one interchange containing one functional group which contains 20 purchase orders. The nondeferred partner receives 5 interchanges containing one functional group and 4 purchase orders each.

1. Calculate the daily volumes of deferred and nondeferred processing.

#### **Deferred requirements:**

Outbound transactions = 160 (2 \* 80 transactions)  
 Outbound functional groups = 8 (2 \* 4 functional groups)  
 Outbound interchanges = 4  
 Outbound deferred records = 12 (3 \* 4 deferred partners)  
 Average # of deferred records = 184

**Note:** The number of deferred records may be reduced by running the Outbound EDI Databank Maintenance program to clean up obsolete deferred records. If the Outbound EDI Databank Maintenance clean feature is used more frequently than the maintenance retention period, reduce the number of records by the number of transactions selected to be enveloped and transmitted.

**Nondeferred requirements:**

Outbound transactions = 20

Outbound functional groups = 10 (2 \* 5 functional groups)

Outbound interchanges = 5

Average # of nondeferred records = 35

Total # of records per day = 219

(# of nondeferred records per day + # of deferred records per day = 184 + 35)

2. Determine the number of records on file, given the retention options chosen. To simplify the calculation, assume that all documents are successfully processed within one day.

Records created over retention period = 876 (219 \* 4)

Records created between maintenance runs = 876 (219 \* 4)

Records on file = 1752 (876 + 876)

**VSAM Data Set Definition**

```

DEFINE          CLUSTER(NAME(@@@@@.EDI.EDIOEA)          -
                VOLUME(VVVVVV)                          -
                RECORDS(1752 876)                        -
                FREESPACE(40 20)                         -
                KEYS(125 0)                               -
                RECORDSIZE(800 800)                      -
                SHR(2 3))                                 -
                DATA(NAME(@@@@@.EDI.EDIOEA.DATA)        -
                CISZ(4096))                              -
                INDEX(NAME(@@@@@.EDI.EDIOEA.INDEX))

```

## Outbound EDI Databank Message Store

EDIOES

### Description

The Outbound EDI Databank Message Store (EDIOES) contains EDI data in the same format that is sent to your trading partners. The EDI data on the Outbound EDI Databank Message Store file can be viewed using CICS Databank Maintenance.

### Activity

This file is optional. If your installation maintains outbound EDI data at the message store-level this file must be defined.

If used, this is a high activity operational file.

### Program References

The following programs reference the Outbound EDI Databank Message Store:

Program	Purpose
Outbound Editor	Updates the file
Outbound EDI Databank Extract	Updates the file
Outbound EDI Databank Maintenance	Updates the file
CICS Databank Maintenance	Reads the file

### Data Set Recovery

Before any files are recovered, the online Change Audit should be reviewed and any outstanding online updates for the Outbound EDI Databank noted (outstanding updates have an applied date and a time of zeros).

The Outbound EDI Databank Directory, Message Store, Change Audit, Pending and Transaction Queue files, as well as the Outbound Application Databank Link file are all closely related. If you need to recover the Outbound EDI Databank Message Store, you must recover the Outbound EDI Databank Directory and Inbound Application Databank Change Audit files, as well as the Outbound Application Databank Link file at the same time, using the same method. The Outbound EDI Databank Pending and Outbound EDI Databank Transaction Queue files must be re-initialized and any pending online changes re-applied.

The Outbound EDI Databank Directory, Message Store, and Change Audit files, and Outbound Application Databank Link file can be recreated from either a recent backup or re-loaded from the Gentran:Basic system tape.

## Disk Space Requirements

Space requirements for the Outbound EDI Databank Message Store depends on whether deferred enveloping and/or nondeferred enveloping is implemented. Approximate space requirements can be calculated based on the following formula:

$$\text{Total \# of records} = \text{\# of nondeferred EDI interchanges} + \text{\# of deferred EDI transactions}$$

Maintaining Message Store data on the Outbound EDI Databank is optional. Message Store usage is controlled by a combination of trading profile and control card options. The calculation shown below assumes that all outbound EDI data sent is being placed on the Outbound EDI Databank Message Store.

### Example

An average of 100 purchase orders are sent each day using deferred enveloping. The average size of a purchase order is 2700 characters.

An average of 50 invoices are sent each day via nondeferred envelope processing. The average size of an invoice is 1500 characters. An average interchange contains 25 invoices; therefore, the average invoice interchange size is 37500 characters.

The Outbound EDI Databank Maintenance program is run twice per week with a Retention-Days-Loaded control card specifying four days.

1. Calculate daily volumes of deferred and nondeferred processing.

#### **Deferred requirements:**

# of deferred transactions per day = 100

average # of characters per deferred transaction = 2700

Average # of deferred Message Store records per transaction = 1

(average # of characters per deferred transaction / Max # of characters per Message Store record = 2700 / 4000)

Average # of Message Store records per day = 100

(average # of Message Store records per transaction \* # of deferred transactions per day = 1 \* 100)

#### **Nondeferred requirements:**

# of nondeferred interchanges per day = 2

average # of characters per interchange = 37500

Average # of nondeferred Message Store records per interchange = 10

(average # of characters per nondeferred interchange / Max # of characters per Message Store record = 37500 / 4000)

Average # of Message Store records per day = 20

(average # of Message Store records per interchange \* # of nondeferred interchanges per day = 10 \* 2)

Total # of records per day = 120

(# of nondeferred records per day + # of deferred records per day = 20 + 100)

2. Determine the number of records on file, given the retention options chosen. To simplify the calculation assume that all documents are successfully processed within one day.

Records created over retention period = 480 (120 \* 4)

Records created between maintenance runs = 480 (120 \* 4)

Records on file = 960 (480 + 480)

### VSAM Data Set Definition

```

DEFINE          CLUSTER(NAME(@@@@@.EDI.EDIOES)          -
                VOLUME(VVVVVVV)                       -
                RECORDS(960 480)                       -
                FREESPACE(00 00)                       -
                KEYS(16 0)                              -
                RECORDSIZE(2050 4050)                  -
                SHR(2 3))                              -
                DATA(NAME(@@@@@.EDI.EDIOES.DATA)     -
                CISZ(4096))                            -
                INDEX(NAME(@@@@@.EDI.EDIOES.INDEX))

```

**Outbound EDI Databank Pending File****EDIOEP****Description**

The Outbound EDI Databank Pending (EDIOEP) file presents the most current view of the Outbound EDI Databank based on CICS Databank Maintenance activity that has been performed.

**Activity**

This is a low activity file.

**Program References**

The following program references the Outbound EDI Databank Pending file:

Program	Purpose
CICS Databank Maintenance	Updates the file.

**Data Set Recovery**

This file can be re-loaded from the Gentran:Basic system tape. If this file is recovered, databank maintenance does not reflect any online changes which were pending against the Outbound EDI Databank until they have been applied in batch.

**Disk Space Requirements**

The actual space requirement for the Outbound EDI Databank Pending file varies depending on the number of databank maintenance updates that are made to the outbound EDI databank, and how often the Outbound EDI Databank Extract program is run. Each update requires one record.

**Example**

The Outbound EDI Databank Extract program is run each night at midnight. On Monday morning, one outbound EDI interchange was deleted from the databank and two outbound EDI purchase orders were edited.

In this case, the Pending file would require three records until the Outbound EDI Databank Extract program was run at midnight. When the Outbound EDI Databank Extract program completes, the pending records are removed.

**VSAM Data Set Definition**

```

DEFINE          CLUSTER(NAME(@@@@@.EDI.EDIOEP)          -
                VOLUME(VVVVVV)                        -
                RECORDS(20 10)                         -
                FREESPACE(20 10)                       -
                KEYS(127 0)                             -
                RECORDSIZE(527 2500)                   -
                SHR(2 3))                              -
                DATA(NAME(@@@@@.EDI.EDIOEP.DATA)      -
                CISZ(4096))                            -
                INDEX(NAME(@@@@@.EDI.EDIOEP.INDEX))

```

## Outbound EDI Databank Transaction Queue

EDIQ091

### Description

The Outbound EDI Databank Transaction Queue (EDIQ091) contains change transactions for the Outbound EDI Databank that users have created through databank maintenance. It is used to move change transactions between the CICS and batch environments. This file is initialized by running the Transaction Queue Initialize program.

### Activity

This is a low activity transaction file.

### Program References

The following programs reference the Outbound EDI Databank Transaction Queue:

Program	Purpose
Transaction Queue Initialize	Updates the file.
Outbound EDI Databank Extract	Updates the file.
CICS Databank Maintenance	Updates the file.

### Data Set Recovery

Before recovering this file, the online Change Audit should be reviewed and any outstanding online updates for the Outbound EDI Databank noted (outstanding updates have an applied date and a time of zeros).

The Outbound EDI Databank Transaction Queue can be re-defined and initialized using the Transaction Queue Initialize program. Any online updates which were pending should be re-applied.

The Transaction Queue must not be recovered from a backup.

### Disk Space Requirements

The actual space requirements for the Outbound EDI Databank Transaction Queue file vary. The requirements depend on the number of Databank Maintenance updates that are made to the Outbound EDI Databank and how often the Outbound EDI Databank Extract program is run. Each update requires one record.

### Example

The Outbound EDI Databank Extract program is run each night at midnight. On Monday morning one outbound EDI interchange was deleted from the databank and two outbound EDI purchase orders were edited.

In this case, the Transaction Queue file would require three records until the Outbound EDI Databank Extract program was run at midnight. When the Outbound EDI Databank Extract program completes the Transaction Queue records are removed.



---

**VSAM Data Set Definition**

```
DEFINE          CLUSTER(NAME(@@@@@.EDI.EDIQ091)      -  
                VOLUME(VVVVVVV)                    -  
                RECORDS(100)                         -  
                NUMBERED                             -  
                RECORDSIZE(4088 4088)                -  
                SHR(4 3))                             -  
                DATA(NAME(@@@@@.EDI.EDIQ091.DATA)  -  
                CISZ(4096))                           -
```

**Note:** No secondary allocation can be defined because the file is formatted by a maintenance program that calculates the exact number of records in the file. This number is required for accurate processing. The maximum number of records that can be allocated when defining this transaction queue file is 32,700.

## Reprocess Work File

**EDIRW**

### Description

The Outbound Mapper and the Inbound Editor uses the Reprocess Work (EDIRW) file for processing Databank Maintenance “Reset” action requests. These are temporary files which are only required during the job step.

The Reprocess Work files are standard sequential files with a record length of 125 (DSORG=PS, LRECL=125, RECFM=FB).

### Activity

This is a low activity file.

### Program References

The following programs reference the Reprocess Work file:

Program	Purpose
Outbound Mapper	Creates the data set
Inbound Editor	Creates the data set

### Disk Space Requirements

The space requirements vary according to the online activity. This is a low activity file with minimal space requirements (1 record per online Reset applied).

TRACKS(1,1) should accommodate most situations.

# Batch Partner Profile Maintenance

## Overview

This chapter contains a high-level explanation of the records that make up the partner profile. Each section describes individual fields found on the record.

See Chapter 2 in the *Gentran:Basic for zSeries Release 6.4 User's Guide* for additional field descriptions.

**Note:** The online Partner subsystem performs the same functions as batch partner profile maintenance, but by using screens instead.

This chapter contains the following topics related to the Batch Partner Profile Maintenance Program (EBDI005):

Topic	Page
<b>Before You Begin</b> .....	<b>9-3</b>
What is the Partner Profile? .....	9-3
What is the Batch Partner Profile Maintenance Facility? .....	9-3
Batch Partner Cross-Reference Table .....	9-3
<b>Partner Profile Set Up</b> .....	<b>9-5</b>
Partner Profile Record Types .....	9-5
Partner Profile Processing .....	9-7
<b>Batch Partner Processing</b> .....	<b>9-10</b>
Partner Profile Batch Update Parameters .....	9-12
Files Used .....	9-14
<b>Header Record</b> .....	<b>9-16</b>
PARTHDR Record .....	9-16
<b>Control Records</b> .....	<b>9-18</b>
PARTMISC Record .....	9-18
PARTIN Record .....	9-23
PARTOUT Record .....	9-27
GENISA Record .....	9-30
GENICS Record .....	9-35
GENBG Record .....	9-39
GENUNA Record .....	9-42

GENUNB1 Record .....	9-46
GENUNB2 Record .....	9-49
GENUNB3 Record .....	9-51
GENUNB4 Record .....	9-54
GENSCH Record .....	9-57
GENSTX1 Record .....	9-60
GENSTX2 Record .....	9-63
GENSTX3 Record .....	9-65
<b>Group Records.....</b>	<b>9-67</b>
GRPMISC Record.....	9-67
GRPIN Record.....	9-72
GRPOUT Record.....	9-75
GENGS Record.....	9-79
GENUNG1 Record.....	9-82
GENUNG2 Record.....	9-85
<b>Transaction Records.....</b>	<b>9-88</b>
TRANMISC Record .....	9-88
TRANIN Record.....	9-92
TRANOUT Record.....	9-96
GENST Record.....	9-99
GENUNH Record.....	9-102
GENMHD Record.....	9-106
<b>Name and Address Records.....</b>	<b>9-109</b>
NAMEAD1 Record.....	9-109
NAMEAD2 Record.....	9-111
NAMEAD3 Record.....	9-113
NAMEAD4 Record.....	9-115
NAMEAD5 Record.....	9-117
<b>User Defined Data Records.....</b>	<b>9-119</b>
USER Record.....	9-119
<b>Data Split Record.....</b>	<b>9-121</b>
Data Split Record.....	9-121
<b>Error Rejection Record.....</b>	<b>9-124</b>
Error Rejection Record .....	9-7
<b>CROSSREF Record .....</b>	<b>9-127</b>
<b>RELPART Record .....</b>	<b>9-129</b>
<b>Processing Reports .....</b>	<b>9-131</b>
Transaction Totals Report .....	9-131
Errors Encountered Processing Partner Data Report.....	9-132

## Before You Begin

### What is the Partner Profile?

The partner profile is a user-entered data set. You can add, change, and delete trading partners by using either the online screens in the online Partner subsystem or the Batch Partner Profile Maintenance Program (EBDI005).

With the partner profile, you can tailor Gentran:Basic to meet the requirements of each specific trading partner for both inbound and outbound processing.

### What is the Batch Partner Profile Maintenance Facility?

The Batch Partner Profile Maintenance facility provides an alternative method for updating the Gentran:Basic partner profile when CICS is either not available or not feasible. You can either enter the input transactions or allow the application to create the input transactions. Processing then occurs against the partner profile, which will perform additions, changes, and deletions.

The Batch Partner Profile Maintenance facility can be helpful if many partner records need to be added to the partner profile, such as in a conversion or for routine maintenance of a separate customer file. The application that maintains the customer file can generate the input transactions to update the Gentran:Basic partner profile; thus, eliminating duplicate effort.

### Batch Partner Cross-Reference Table

The following table cross-references each Batch Partner Control record with a copybook. Use these copybooks in your own application programs. All copybooks are located in the product source utility library.

Batch Partner Control Records	Source Utility Copybook
<b>Partner Level of Profile</b>	
PARTHDR record	Copybook PRTIAHRC
<b>Control Level of Profile</b>	
PARTMISC record	Copybook PRTICMRC
PARTIN record	Copybook PRTICIRC
PARTOUT record	Copybook PRTICORC
GENISA record	Copybook PRTIARC
GENICS record	Copybook PRTISRC
GENBG record	Copybook PRTIBGRC
GENUNA record	Copybook PRTIGARC
GENUNB1 record	Copybook PRTIB1RC
GENUNB2 record	Copybook PRTIB2RC
GENUNB3 record	Copybook PRTIB3RC
GENUNB4 record	Copybook PRTIB4RC
GENSCH record	Copybook PRTIGBRC

GENSTX1 record	Copybook PRTIS1RC
GENSTX2 record	Copybook PRTIS2RC
GENSTX3 record	Copybook PRTIS3RC
<b>Group Level of Profile</b>	
GRPMISC record	Copybook PRTIGMRC
GRPIN record	Copybook PRTIGIRC
GRPOUT record	Copybook PRTIGORC
GENGS record	Copybook PRTIGSRC
GENUNG1 record	Copybook PRTIG1RC
GENUNG2 record	Copybook PRTIG2RC
<b>Transaction Level of Profile</b>	
TRANSMISC record	Copybook PRTITMRC
TRANIN record	Copybook PRTITIRC
TRANOUT record	Copybook PRTITORC
GENST record	Copybook PRTISTRC
GENUNH record	Copybook PRTIGHRC
GENMHD record	Copybook PRTIMHRC
<b>Name and Address Level of Profile</b>	
NAMEAD1 record	Copybook PRTIN1RC
NAMEAD2 record	Copybook PRTIN2RC
NAMEAD3 record	Copybook PRTIN3RC
NAMEAD4 record	Copybook PRTIN4RC
NAMEAD5 record	Copybook PRTIN5RC
<b>User Level of Profile</b>	
User record	Copybook PRTIUTRC
<b>Data Split Level of Profile</b>	
Data Split record	Copybook PRTIDSRC
<b>Error Rejection Level of Profile</b>	
Error Rejection record	Copybook PRTIERRC
<b>Cross-Reference Level of Profile</b>	
CROSSREF record	Copybook PRTCERRC
<b>Relationship Level of Profile</b>	
RELPART record	Copybook PRTPRLRC

## Partner Profile Set Up

This section explains the batch interface for maintaining the partner profile. This section also includes the transactions required to add, change, and delete records from the Partner file, inbound and outbound Control files, Partner X-Ref file, and/or the Partner Relationship file.

### Partner Profile Record Types

A set of eight maintenance record types make up the Gentran:Basic partner profile. You must create the two primary record types, *Header* and *Control*, before you can access the remaining records. The partner profile includes:

- Header record
- Control record
- Group record
- Transaction record
- Name and Address record
- User Defined Data record
- Data Split record
- Error Rejection record

**Note:** The User Defined Data records are not used for Gentran:Basic processing, unless TCR processing is set up. These records store pertinent trading partner information to maintain all information about a trading partner in one file.

#### *Header Record*

The Header record type is the first record that you need to create in the partner profile. This **mandatory** record defines the trading partner to the system. The Header record is added through the PARTHDR transaction.

#### *Control Record*

The Control record type is the second record that you need to create in a partner profile. This record is added through the PARTMISC transaction.

When you add a Control record to the partner profile, Gentran creates default Group and Transaction records for the new trading partner. The model Group and Transaction records, that Gentran permanently maintains in the partner profile, controls default record generation.

#### **Partner/Qualifier Mode**

Gentran adds all default records with the same Partner ID and Qualifier as the Control record, and with a Group or Transaction ID of !!!DFT. The Group and Transaction record models have a partner key of !!!GENTRAN-RESERVED-PARTNER-ID-1 and a Group or Transaction ID of !!!DFT. You can modify the model records, either online or in batch to customize default record generation.

**Relationship Mode**

Gentran adds all default records with the same User ID and Partner ID as the Control record, and with a Group or Transaction ID of !!!DFT. The Group or Transaction record models have a User ID of !!!GENTRAN-RU1, Partner ID of !!!GENTRAN-RP1, and a Group or Transaction ID of !!!DFT.

***Group Record***

The Group record has the following two purposes:

- For use by the VERIFY PARTNER GROUP global parameter  
See Appendix A for information about verification.
- For building the functional group header/trailer information when generating outbound envelopes

***Transaction Record***

The Transaction record has the following two purposes:

- For use by the VERIFY PARTNER TRANSACTION global parameter as defined in the Global Parameter file
- For building the transaction set header/trailer information when generating outbound envelopes  
See Appendix A for information about verification.

***Name and Address Record***

The Name and Address field enables you to print the trading partner's name on various reports, including the Editor Compliance reports. The fields on this record provide demographic information about each trading partner.

***User Defined Data Record***

The User Defined Data record stores customized user-defined data. Except for TCR processing, Gentran does not use these records.

See Chapter 3, "Outbound Processing Programs," for more information.

***Data Split Record***

The Data Split record contains information about how to split inbound and outbound data. The Inbound and Outbound Editors (programs EBDI001 and EBDI002, respectively) use the Data Split record when you use the Data Split feature of Gentran:Basic. You must also code the Directed Output Files record of the global parameters file if you are using the Data Split feature.

See the topic "Inbound Split" section in Chapter 2 or the topic "Outbound Split" in Chapter 3 for more information.



### *Error Rejection Record*

The Error Rejection record works in conjunction with the Error Message file to set indicators used by the Inbound and Outbound Editors (programs EBDI001 and EBDI002, respectively) to perform these two functions:

- Identify unacceptable errors
- Route input data to the Reject file

The Error Rejection feature is the only way to write records to the Reject file. When you use the Error Rejection feature, you must also code the Partner Error Control record of the global parameters file.

See the topic “Error Rejection Processing” in either Chapter 2 or Chapter 3 of this guide for further details.

## Partner Profile Processing

### *Partner X-Ref (Cross-Reference) File*

The Partner X-Ref file enables you to create one set of Partner records for a trading partner in the partner profile, and then point any number of identifiers (either inbound/outbound or sender/receiver) to that trading partner. This substitution feature applies to interchange and group levels for both Sender and Receiver IDs, and can be used for both inbound and outbound processing. The actual input and output from the Editors does not change. The substitution feature substitutes only the Partner record used in processing. This enables you to create a Partner ID with any name or number that is meaningful to you and cross-reference it to the Partner ID using the Partner X-Ref file.

Each Partner X-Ref record contains a Cross Reference ID and Partner ID with a Qualifier. The key to the file is the Cross Reference ID and Qualifier. You can add, update, and delete records by using either the online Partner subsystem or Batch Partner Profile Maintenance.

#### **Example**

In the following example, whenever the batch programs call for a Partner ID that corresponds to either the Network ID or DUNS number, the Sterling Partner ID is substituted.

ISA Record

Sender ID = Network ID

GS Record

Sender ID = DUNS #

Partner X-Ref File:

	<b>X-Ref ID</b>	<b>Partner ID</b>
<b>Rec #1</b>	Network ID	Sterling
<b>Rec #2</b>	DUNS #	Sterling

### Partner Relationship File

The Partner Relationship file enables you to create unique trading relationships on the partner profile to identify trading partner IDs that represent multiple entities within your organization. The relationship definition points any number of identifiers to that trading partner and is used for both inbound and outbound processing. The actual input and output from the Editors does not change. The Partner X-Ref file does not exist when processing in the Relationship mode.

The Sender and Receiver EDI IDs from the interchange, group, and transaction levels are converted to a partner relationship key, composed of both a user portion and partner portion, for the partner profile. For inbound processing, the interchange or group Receiver ID and Qualifier comprise the *user* portion of the key and the interchange or group Sender ID and Qualifier comprise the *partner* portion of the key. For outbound processing, user refers to the sender or originator of the application data and partner refers to the receiver of EDI data. You can add, update, and delete records by using either the online Partner subsystem or Batch Partner Profile Maintenance.

#### Example

In the following example, a unique relationship is built by combining the EDI IDs with Qualifiers and associating this combination to the User/Partner ID that exists on the partner profile.

User EDI ID	Partner EDI ID	User	Partner
Interchange or Group Receiver ID	Interchange or Group Sender ID	Receiver ID	Sender ID
Interchange or Group Receiver Qualifier ID	Interchange or Group Sender Qualifier ID		

**Note:** Whenever the batch processing programs call for a User/Partner ID that is associated with User EDI ID and Partner EDI ID, then User/Partner are substituted.

### Cross-Reference Maintenance

Cross-reference maintenance is available to make additions, changes, or deletions from the Gentran:Basic Partner Cross Reference file. Requirements for performing cross-reference maintenance are:

- To add records, the cross-reference trading partner cannot exist in the Gentran:Basic partner profile.
- To change or delete records, the cross-reference alias must exist in the partner profile.
- The edits follow online requirements.

The cross-reference alias uses the same partner profile division for security.

### *Partner Relationship Maintenance*

Partner relationship maintenance is available to make additions, changes, or deletions from the Gentran:Basic Partner Relationship file. Requirements for performing partner relationship maintenance are:

- To add records, the partner relationship IDs cannot exist in the Gentran:Basic partner profile.
- To change or delete records, the partner relationship IDs must exist in the partner profile.
- The edits follow online requirements.

The partner relationship uses the same partner profile division for security.

### **Partner Profile Processing Functionality**

#### *Cross-Reference Cleanup*

Cross-reference cleanup is a functionality that processes the cross-reference sequentially.

See the global parameter value “XREF CLEANUP” in the Global Parameter Values section later in this chapter.

#### *Relationship Cleanup*

Relationship cleanup is a functionality that processes the relationship file sequentially.

See the global parameter value ‘PREL CLEANUP’ in the Global Parameter Values section later in this chapter.

#### *INS Relationship Control (for Gentran:Plus Users)*

For action code C (Change) on transaction GENSTX1, changes to the global parameter PCTR-STX-NETWORK-ID either cancel or establish all transactions for old and new TUN numbers.

For action code C (Change) on transactions TRANIN and TRANOUT, additional processing is needed to \*CANCEL or \*ESTABLISH based on changes to PTR-TRANSACTION-ACCEPT-FLAG and PTR-TRANSACTION-SEND-FLAG.

## Batch Partner Processing

### Field Descriptions

In each section of input transaction records, the first fields are divided according to the Partner/Qualifier and Relationship modes, as applicable. The following table provides the constant fields found in most records, as well as the action to perform for the field.

Field	Description
Action Code	<i>Action Code</i> . Perform an action for file maintenance tasks, such as adding, updating, and deleting. See the sub-topic “Action Codes” in this section.
Record Type	<i>Transaction Type</i> . Enter the type of transaction record.
<b>Partner/Qualifier Mode</b>	
Partner ID	<i>Partner ID</i> . Enter the Partner ID for the trading partner.
Partner ID Qualifier	<i>Partner ID Qualifier</i> . Enter the Partner ID and Qualifier for the trading partner.
<b>Relationship Mode</b>	
User ID	<i>User ID</i> . Enter the User ID for the trading partner.
Partner ID	<i>Partner ID</i> . Enter the Partner ID for the trading partner.

Review the section for each input transaction record for the remaining fields and descriptions.

### Action Codes

The following table describes action codes for maintenance that you can perform to records.

Data Maintenance Type	Action Code	Description
Add	A	<p>Adds a record to the partner profile or adds fields to an existing record.</p> <p>You can add a field to an existing record if that field is currently blank. If you add a field to a record that does not yet exist, that record is created to accommodate the data.</p>
Change	C	<p>Changes the contents of a field in the file to the corresponding field contents on the change transaction.</p> <p>To change a field in the file to blanks, type an exclamation point (!) left-justified in the field on the change transaction.</p>
Delete	D	<p>Deletes a record in its entirety if no data is in any field following the transaction type.</p> <p>If any field contains a non-blank value, that field on the file record is deleted by changing it to blanks. This has the same effect as entering “!” on the change transaction.</p>

### Transactions

After creating the Header record, you can perform several maintenance operations. Consider the following before submitting transactions:

- To add records, a Partner Control record must exist or be added first.
- To change fields, the record that the fields are on must exist.
- To delete fields or records, those fields or records must exist.

Partner profile record types with multiple maintenance records can be deleted only through the first (PARTHDR) and second (PARTMISC) primary maintenance records. For the following partner profile record types, use the specified maintenance record/transaction type for deletions.

Record Type	Transaction
Header	PARTHDR
Partner (Control)	PARTMISC
Group	GRPMISC
Transaction	TRANMISC
Name & Address	NAMEAD1

**Caution:** The PARTHDR delete transaction will delete *all* records for a trading partner.

You can type the exclamation point (!) for any record except the Data Split record.

Following the description of some fields is a value enclosed in parentheses. This value represents the Envelope ID and the position in the data for EDI-formatted data.

#### Example

(ISA06)

This example represents the ISA segment, the sixth element, which is the Interchange Sender ID.

## Partner Profile Batch Update Parameters

### Parameter Record Layout

The following table describes the parameter record layout.

Position	Field Content	COBOL Picture	Format	Length	Description
1 – 22	ID	X(22)	AN	22	Global Parameter Value
23 – 80	Filler	X(58)	AN	58	Unused

### Parameter Values

The values in bold may be entered in the Parameter Value field (1 – 22).

#### PRINT ALL TRANSACTIONS

Displays images of incoming transactions including any preceding errors.

**INS SUPPORT ON**

Enables INS network relationship processing. The default setting is INS SUPPORT OFF.

See the sub-topic “Files Used” for files SYS093, SYS094, and SYS095 in this guide.

INS relationship control is a functionality that allows INS relationship processing to execute. Originally, this INS relationship processing was updated for every GENSTX1, TRANIN, and TRANOUT. This provides run-time control over placing INS network commands in the following files:

- SYS093
- Network Header
- SYS094
- Network details

**INS SUPPORT OFF**

Disables INS network relationship processing. The files required with INS SUPPORT ON are not used.

**XREF CLEANUP**

Forces unused cross-reference entries to be deleted from Gentran:Basic. Use this option for periodic maintenance requiring cleanup of the Partner X-Ref file. XREF CLEANUP processing is performed after all maintenance transactions have been processed.

Cross-reference cleanup is a functionality that processes the cross-reference sequentially. For every cross-reference record, the cross-reference alias is validated as an existing partner profile. If the partner profile does not exist, the cross-reference entry is deleted. A count of *cleaned* cross-references is produced in a banner listing.

**PREL CLEANUP**

Forces unused relationship file entries to be deleted from Gentran:Basic. Use this option for periodic maintenance requiring cleanup of the Partner Relationship file. PREL CLEANUP processing is performed after all maintenance transactions have been processed. This option is valid only when using the Relationship or Mixed modes as indicated in the 0 Record of the Configuration file.

Relationship cleanup is a functionality that processes the relationship file sequentially. For every relationship record, the user/partner is validated as an existing partner profile. If the partner profile does not exist, the relationship entry is deleted. A count of *cleaned* relationships is produced in a banner listing.

## Files Used

The following files are used in batch partner profile maintenance processing.

Filename	Description
EDICFG	System Configuration File (Input)
EDISTRN	Gentran:Basic Standards Transaction File (Indexed Input) Validates transactions and provides logical document identification for INS support.
SYS040	Data Maintenance Transaction (Seq Input/250-byte Input) Contains all input sequential transactions (e.g., PARTHDR, PARMISC, PARTIN, PARTOUT, GRPMISC, GRPIN).
ERRCTL	Error Message and Control File (Indexed Input/Output) Contains descriptions of every error message in the Gentran:Basic system.
SYS010	Errors Encountered Processing Partner Data Report (Output) Contains all input sequential transactions (i.e., PARTHDR, PARTMISC, PARTIN, PARTOUT, GRPMISC, GRPIN).
SYS030	Input Parameters File This file contains input parameters to partner profile maintenance.
SYS089	Gentran:Basic Mapping Transaction Header (Indexed Input) Identifies the proper map header on a group.
SYS090	Gentran:Basic Partner Profile (Indexed Input/Output) Contains the primary destination of the majority of the profile information. Specifically, this file contains all Control, Group, Transaction, Data Separation, Name and Address, User Defined, and Error Rejection profile information.
SYS091	Gentran:Basic Partner Profile Outbound Control (Indexed Input/Output) Is used in the background maintenance of all outbound control numbers for use in envelope generation.
SYS092	Gentran:Basic Partner Profile Inbound Control (Indexed Input/Output) Is used in the background maintenance of all inbound control numbers for use in sequencing.
SYS093	Gentran:Basic INS Network Header (Indexed Input) Is used in INS relationship support.  <b>Note:</b> This file is required only if the global parameter INS SUPPORT ON is used.



---

Filename	Description
SYS094	Gentran:Basic INS Network Detail (Indexed Output) Is used in INS relationship support. <b>Note:</b> This file is required only if the global parameter INS SUPPORT ON is used.
SYS095	Gentran:Basic Partner Profile Cross-Reference (Indexed Input/Output) Cross-reference maintenance with the partner profile, and in INS relationship support.
EDIPREL	Gentran:Basic Partner Relationship (Indexed Input/Output) Relationship maintenance with the partner profile.

## Header Record

### PARTHDR Record

#### Description

A PARTHDR transaction defines the Header record of the partner profile. This record is used to add, modify, and delete partner profiles. This record specifies:

#### Record Layout

Name	Data Type	Position
01 IAHT-INPUT-PARTNER-HEADER-TRAN.		
05 IAHT-ACTION-CODE	X(01)	001 - 001
05 IAHT-PARTNER-KEY.		
10 IAHT-PARTNER-ID	X(35)	002 - 036
10 IAHT-PARTNER-ID-QUAL	X(04)	037 - 040
10 IAHT-RECORD-TYPE	X(16)	041 - 056
05 IAHT-USRPAR-KEY		
REDEFINES IAHT-PARTNER-KEY.		
10 IAHT-USRPAR-USER-ID	X(15)	002 - 016
10 IAHT-USRPAR-PARTNER-ID	X(15)	017 - 031
10 FILLER	X(09)	032 - 040
10 IAHT-USRPAR-RECORD-TYPE	X(16)	041 - 056
05 IAHT-DESCRIPTION	X(80)	057 - 136
05 IAHT-UNDERSCORE-CHAR	X(01)	137 - 137
05 FILLER	X(01)	138 - 138
05 IAHT-LOCKED	X(01)	139 - 139
05 IAHT-DIVISION	X(03)	140 - 142
05 FILLER	X(19)	143 - 161

#### PARTHDR Input Transaction Fields

##### IAHT-ACTION-CODE (Required)

Description: *Action Code*

Valid Value(s):

A	=	Add
C	=	Change
D	=	Delete

##### Partner/Qualifier Mode

##### IAHT-PARTNER-ID (Required)

Description: *Partner ID*

##### IAHT-PARTNER-ID-QUAL (Required)

Description: *Partner ID Qualifier*

##### IAHT-RECORD-TYPE (Required)

Description: *Transaction Type*

Valid Value(s): Constant PARTHDR (left-justified and space-filled).

**Relationship Mode**

IAHT-USRPAR-USER-ID (Required)

Description: *User ID*

IAHT-USRPAR-PARTNER-ID (Required)

Description: *Partner ID*

IAHT-USRPAR-RECORD-TYPE (Required)

Description: *Transaction Type*

Valid Value(s): Constant PARTHDR (left-justified and space-filled).

IAHT-DESCRIPTION

Description: *Partner Description*. Enter trading partner information.

IAHT-UNDERSCORE-CHAR

Description: *Underscore character*. Indicate substitution character to be used when displaying fields that contain an underscore ( *\_* ) on the online partner screens.

Valid Value(s):

~, @, #, \$, %, -, &amp;, \*, (, ), -, =, +, {, }, :, ;, &lt;, &gt;, /, ?, |, \, and space.

IAHT-LOCKED

Description: *Locked*. Specify whether trading partner information can be updated.

Valid Value(s):

Y	=	Update is allowed
N	=	No update is allowed

IAHT-DIVISION

Description: *Division Code*. Enter a user-specified code identifying the division.

## Control Records

### PARTMISC Record

#### Description

PARTMISC record defines the partner profile. This record is used to add, modify, and delete Control records. This record specifies:

- The header type used for outbound transmission header generation
- Whether to acknowledge interchanges, groups, transactions, or errors
- Whether the partner acknowledges received interchanges

#### Record Layout

Name	Data Type	Position
01 ICMT-INPUT-PARTNER-MISC-TRAN.		
05 ICMT-ACTION-CODE	X (01)	001 - 001
05 ICMT-PARTNER-KEY.		
10 ICMT-PARTNER-ID	X (35)	002 - 036
10 ICMT-PARTNER-ID-QUAL	X (04)	037 - 040
10 ICMT-RECORD-TYPE	X (16)	041 - 056
05 ICMT-USRPAR-KEY		
REDEFINES ICMT-PARTNER-KEY.		
10 ICMT-USRPAR-USER-ID	X (15)	002 - 016
10 ICMT-USRPAR-PARTNER-ID	X (15)	017 - 031
10 FILLER	X (09)	032 - 040
10 ICMT-USRPAR-RECORD-TYPE	X (16)	041 - 056
05 ICMT-INTERCHANGE-HEADER-OPTION	X (03)	057 - 059
05 ICMT-INTERCHANGE-VERSION	X (12)	060 - 071
05 ICMT-ACKNOWLEDGE-INTERCHANGE	X (01)	072 - 072
05 ICMT-NETWORK-TRACKING	X (01)	073 - 073
05 ICMT-ACKNOWLEDGE-ERRORS	X (01)	074 - 074
05 ICMT-EXPECT-INTERCHG-ACK	X (01)	075 - 075
05 ICMT-OUTBOUND-COMPRESS-LEVEL	X (01)	076 - 076
05 ICMT-DUNS-NUMBER	X (15)	077 - 091
05 FILLER	X (04)	092 - 095
05 ICMT-PARTNER-TYPE	X (01)	096 - 096
05 ICMT-PARTNER-ID-TYPE	X (01)	097 - 097
05 ICMT-ALT-ACK-PARTNER-ID	X (35)	098 - 132
05 ICMT-ALT-ACK-PARTNER-ID-QUAL	X (04)	133 - 136
05 ICMT-PLUS-REMOTE-ID	X (08)	137 - 144
05 ICMT-PLUS-NETWORK-ID	X (15)	145 - 159
05 ICMT-VIEWPOINT	X (01)	160 - 160
05 ICMT-VIEWPOINT-TM	X (01)	161 - 161
05 ICMT-MULTIPLE-ENVELOPE-ID	X (03)	162 - 164
05 ICMT-PARTNER-INT-VERSION	X (12)	165 - 176

**PARTMISC Input Transaction Fields**

ICMT-ACTION-CODE (Required)

Description: *Action Code*

Valid Value(s):

A	=	Add
C	=	Change
D	=	Delete

**Partner/Qualifier Mode**

ICMT-PARTNER-ID (Required)

Description: *Partner ID*

ICMT-PARTNER-ID-QUAL (Required)

Description: *Partner ID Qualifier*

ICMT-RECORD-TYPE (Required)

Description: *Transaction Type*

Valid Value(s): Constant PARTMISC (left-justified and space-filled).

**Relationship Mode**

ICMT-USRPAR-USER-ID (Required)

Description: *User ID*

ICMT-USRPAR-PARTNER-ID (Required)

Description: *Partner ID*

ICMT-USRPAR-RECORD-TYPE (Required)

Description: *Transaction Type*

Valid Value(s): Constant PARTMISC (left-justified and space-filled).

ICMT-INTERCHANGE-HEADER-OPTION (Required)

Description: *Interchange Header Option*. Specify which interchange header to generate for outbound transmissions.

Valid Value(s):

ISA	=	Generate ISA, IEA control envelopes
ICS	=	Generate ICS, ICE control envelopes
BG	=	Generate BG, EG control envelopes
GS	=	Do not generate transmission control envelope; use GS, GE instead
SCH	=	Generate SCH, END
STX	=	Generate STX, END
UNA	=	Generate UNA, UNZ control envelope
UNB	=	Generate UNB, UNZ control envelopes

ICMT-INTERCHANGE-VERSION

Description: *Interchange Version*. Enter the version of the EDI standards to use when generating the interchange header. This will also load into the ICS04 and ISA12, when applicable.

## ICMT-ACKNOWLEDGE-INTERCHANGE

Description: *Acknowledge Interchange*. Indicate to the inbound editor whether to generate a TA1 segment for every ISA interchange received, an AC1 segment for every ICS interchange received, or a UCI segment for every UNB interchange received.

Valid Value(s):

Y	=	Yes, always generate a TA1, AC1, or UCI.
N	=	No, do not generate a TA1, AC1, or UCI.
E	=	Evaluate the interchange envelope (ISA and UNB only) If the ISA14 is 1, generate a TA1. If the ISA14 is 0, do not generate a TA1. If the UNB09 is 1, generate a UCI. If the UNB09 is 0, do not generate a UCI.

Only applicable if used with the PARTNER ACKNOWLEDGEMENT global parameter.

## ICMT-NETWORK-TRACKING

Description: *Network Tracking*. Specify whether network tracking is valid for outbound databanking.

Valid Value(s):

Y	=	Network tracking is valid for outbound databanking
N	=	Network tracking is <i>not</i> valid for outbound databanking

## ICMT-ACKNOWLEDGE-ERRORS

Description: *Acknowledge Errors*. Specify whether to create acknowledgements when an errors occurs.

Valid Value(s):

Y	=	Generate error acknowledgments
N	=	Do not generate error acknowledgments
Blank	=	Do not generate error acknowledgments

## ICMT-EXPECT-INTERCHG-ACK

Description: *Expect Interchange Acknowledgment*. Set up the trading partner to send interchange acknowledgments.

Valid Value(s):

Y	=	Acknowledgment should be expected
N	=	Acknowledgment is not used
Blank	=	Acknowledgment is not used

## ICMT-OUTBOUND-COMPRESSSION-LEVEL

Description: *Outbound Compression Level*. Reserved for future use.

Valid Value(s):

I	=	Interchange
G	=	Group
T	=	Transaction

## ICMT-DUNS-NUMBER

Description: *DUNS Number*. Enter the partner DUNS number. (Not actively used by Gentran:Basic.)

## ICMT-PARTNER-TYPE

Description: *Partner Type*. Indicate which partner profile to use.

Valid Value(s):

U	=	User partner
T	=	Trading partner
N	=	Network partner

## ICMT-PARTNER-ID-TYPE

Description: *Partner ID Type*. Apply the Partner ID to use.

Valid Value(s):

E	=	EDI identification
A	=	Application identification
D	=	Descriptive name

## ICMT-ALT-ACK-PARTNER-ID

Description: *Alternate Acknowledgment Partner ID*. Enter an alias Partner ID to use for alternate acknowledgment partner. (The alias used in reconciliation against outbound EDI databank directory.)

## ICMT-ALT-ACK-PARTNER-ID-QUAL

Description: *Alternate Acknowledgment Partner ID Qualifier*. Enter a Partner ID and Qualifier to use as an alias for alternate acknowledgment partner. (The alias used in reconciliation against outbound EDI databank directory.)

## ICMT-PLUS-REMOTE-ID

Description: *Plus Remote ID*. In Gentran:Plus, associate a trading partner with the communications profile established by Remote ID.

## ICMT-PLUS-NETWORK-ID

Description: *Plus Network ID*. In Gentran:Plus, this field is used if ICMT-NETWORK-TRACKING is activated. Provide INS support to associate a trading partner with network tracking.

## ICMT-VIEWPOINT

Description: *Viewpoint*. In Gentran:Viewpoint, indicate whether exception management activity should be performed at the interchange level.

Valid Value(s):

Y	=	Yes, Gentran:Viewpoint Exception Management activity should be performed at the interchange level.
N	=	No, Gentran:Viewpoint Exception Management activity should <i>not</i> be performed at the interchange level.
Blank	=	This field must be blank if Viewpoint is not installed.

## ICMT-VIEWPOINT-TM

Description: *Viewpoint Tracking Management*. In Gentran:Viewpoint, indicate whether tracking management activity should be performed for this interchange level.

Valid Value(s):

Y	=	Yes, Gentran:Viewpoint Tracking Management activity should be performed for this interchange level.
N	=	No, Gentran:Viewpoint Tracking Management activity should <i>not</i> be performed for this interchange level.
Blank	=	This field must be blank if Viewpoint is not installed.

## ICMT-MULTIPLE-ENVELOPE-ID

Description: *Multiple Envelope ID*. Determine which interchange envelope to generate for outbound transmissions.

Valid Value(s):

ISA	=	Generate ISA, IEA control envelopes
ICS	=	Generate ICS, ICE control envelopes
BG	=	Generate BG, EG control envelopes
GS	=	Do not generate transmission control envelope; use GS, GE instead
SCH	=	Generate SCH, END
STX	=	Generate STX, END
UNA	=	Generate UNA, UNZ control envelopes
UNB	=	Generate UNB, UNZ control envelopes

## ICMT-PARTNER-INT-VERSION

Description: *Version*. Use the partner interchange version when more than 1 control record with the same interchange header option is needed. Setting up partners with the partner interchange version should only be used when multiple like interchange envelopes need to be generated with different information.



**PARTIN Record***Description*

The PARTIN transaction record specifies additional fields found on the partner profile Control record.

*Record Layout*

<b>Name</b>	<b>Data Type</b>	<b>Position</b>
01 ICIT-INPUT-PART-INCOMING-TRAN.		
05 ICIT-ACTION-CODE	X(01)	001 - 001
05 ICIT-PARTNER-KEY.		
10 ICIT-PARTNER-ID	X(35)	002 - 036
10 ICIT-PARTNER-ID-QUAL	X(04)	037 - 040
10 ICIT-RECORD-TYPE	X(16)	041 - 056
05 ICIT-USRPAR-KEY		
REDEFINES ICIT-PARTNER-KEY.		
10 ICIT-USRPAR-USER-ID	X(15)	002 - 016
10 ICIT-USRPAR-PARTNER-ID	X(15)	017 - 031
10 FILLER	X(09)	032 - 040
10 ICIT-USRPAR-RECORD-TYPE	X(16)	041 - 056
05 ICIT-US-PARAMETERS.		
10 ICIT-INTERCHANGE-VERSION-USE	X(01)	057 - 057
10 ICIT-INBD-INTERCHG-VERSION	X(05)	058 - 062
10 ICIT-INBD-BG-PASSWORD	X(10)	063 - 072
10 ICIT-BG-COMMUNICATION-ID	X(10)	073 - 082
10 ICIT-BG-COMMUNICATIONS-PASSWD	X(10)	083 - 092
10 ICIT-ACCEPT-ALL-TRANSACTIONS	X(01)	093 - 093
10 ICIT-LAST-INBD-INTERCHG	X(09)	094 - 102
10 ICIT-INBD-ISA-PASSWD-QUAL	X(02)	103 - 104
10 ICIT-INBD-ISA-PASSWORD	X(10)	105 - 114
05 ICIT-INTERNATIONAL-PARAMETERS.		
10 ICIT-INBD-INTL-PASSWD-QUAL	X(02)	115 - 116
10 ICIT-INBD-INTL-PASSWORD	X(10)	117 - 126
10 ICIT-INBD-INTL-VERSION-USE	X(01)	127 - 127
10 ICIT-INBD-INTL-VERSION	X(01)	128 - 128
10 ICIT-INBD-INTL-SYNTAX-ID	X(04)	129 - 132
10 ICIT-LAST-INBD-INTL-CNTL-NO	X(14)	133 - 146
05 ICIT-INTERCHG-INBD-DB-LEVEL	X(01)	147 - 147
05 ICIT-APPL-INBD-DB-LEVEL	X(01)	148 - 148
05 ICIT-MULTIPLE-ENVELOPE-ID	X(03)	149 - 151
05 ICIT-PARTNER-INT-VERSION	X(12)	152 - 163

***PARTIN FMS Transaction Layout***

ICIT-ACTION-CODE (Required)

Description: *Transaction Action Code*

Valid Value(s):

A	=	Add
C	=	Change
D	=	Delete

**Partner/Qualifier Mode**

ICIT-PARTNER-ID (Required)

Description: *Partner ID*

ICIT-PARTNER-ID-QUAL (Required)

Description: *Partner ID Qualifier*

ICIT-RECORD-TYPE (Required)

Description: *Transaction Type*

Valid Value(s): Constant PARTIN (left-justified and space-filled).

**Relationship Mode**

ICIT-USRPAR-USER-ID (Required)

Description: *User ID*

ICIT-USRPAR-PARTNER-ID (Required)

Description: *Partner ID*

ICIT-USRPAR-RECORD-TYPE (Required)

Description: *Transaction Type*

Valid Value(s): Constant PARTIN (left-justified and space-filled).

ICIT-INTERCHANGE-VERSION-USE

Description: *Interchange Version Use*. Define the version to use for this interchange.

Valid Value(s):

A	=	Always use this version for this partner
I	=	Use version from data
D	=	Use this version if cannot find version on input data

ICIT-INBD-INTERCHG-VERSION

Description: *Interchange Version Number*. Enter the version number for the interchange.

ICIT-INBD-BG-PASSWORD

Description: *Incoming BG Password*. Enter a password to use to verify transmission from the trading partner.

ICIT-BG-COMMUNICATION-ID

Description: *BG Communication ID*. Add the Communication ID to use when generating BG control segments (BG01) and ISA segments (ISA02) for this partner.

## ICIT-BG-COMMUNICATIONS-PASSWD

Description: *BG Communications Password*. Add a communications password to use when generating BG control segments (BG02) and ISA segments (ISA04) for this partner.

## ICIT-ACCEPT-ALL-TRANSACTIONS

Reserved for future use.

## ICIT-LAST-INBD-INTERCHG

Description: *Last Inbound Interchange*. Set up how Gentran checks the last control number received on an interchange.

Valid Value(s):

Zeros	=	Interchange control number checking starts at 1
Non-zero numerals	=	Interchange control number checking begins at n+1
Blank	=	Checking function is <i>not</i> used

## ICIT-INBD-ISA-PASSWD-QUAL

Reserved for future use.

## ICIT-INBD-ISA-PASSWORD

Reserved for future use.

## ICIT-INBD-INTL-PASSWD-QUAL

Reserved for future use.

## ICIT-INBD-INTL-PASSWORD

Reserved for future use.

## ICIT-INBD-INTL-VERSION-USE

Reserved for future use.

## ICIT-INBD-INTL-VERSION

Reserved for future use.

## ICIT-INBD-INTL-SYNTAX-ID

Reserved for future use.

## ICIT-INBD-INTL-CNTL-NO

Reserved for future use.

## ICIT-INTERCHG-INBD-DB-LEVEL

Description: *Interchange Inbound Databank Level*. Set the partner profile inbound EDI databanking level for directory maintenance.

Valid Value(s):

D	=	Directory
N	=	None

## ICIT-APPL-INBD-DB-LEVEL

Description: *Application Inbound Databank Level*. Not currently used.

## ICIT-MULTIPLE-ENVELOPE-ID

Description: *Multiple Envelope ID*. Determine which interchange envelope to generate for outbound transmissions.

Valid Value(s):

ISA	=	Generate ISA, IEA control envelopes
ICS	=	Generate ICS, ICE control envelopes
BG	=	Generate BG-EG control envelopes
GS	=	Do not generate transmission control envelopes; use GS, GE instead
SCH	=	Generate SCH, END
STX	=	Generate STX, END
UNA	=	Generate UNA, UNB, UNZ control envelopes
UNB	=	Generate UNB, UNZ control envelopes

## ICIT-PARTNER-INT-VERSION

Description: *Version*. Use the partner interchange version when more than one control record with the same interchange header option is needed. Setting up partners with the partner interchange version should only be used when multiple like interchange envelopes need to be generated with different information.

**PARTOUT Record***Description*

PARTOUT Transaction records allow you to edit fields found on the partner profile Control record.

*Record Layout*

Name	Data Type	Position
01 ICOT-INPUT-PART-OUTGOING-TRAN.		
05 ICOT-ACTION-CODE	X(01)	001 - 001
05 ICOT-PARTNER-KEY.		
10 ICOT-PARTNER-ID	X(35)	002 - 036
10 ICOT-PARTNER-ID-QUAL	X(04)	037 - 040
10 ICOT-RECORD-TYPE	X(16)	041 - 056
05 ICOT-USRPAR-KEY		
REDEFINES ICOT-PARTNER-KEY.		
10 ICOT-USRPAR-USER-ID	X(15)	002 - 016
10 ICOT-USRPAR-PARTNER-ID	X(15)	017 - 031
10 FILLER	X(09)	032 - 040
10 ICOT-USRPAR-RECORD-TYPE	X(16)	041 - 056
05 ICOT-US-PARAMETERS.		
10 ICOT-OUTBD-ICRG-VERSN-USE	X(01)	057 - 057
10 ICOT-OUTBD-INTERCHG-VERSION	X(05)	058 - 062
10 ICOT-LAST-OUTBD-ICRG-CNTL	X(09)	063 - 071
05 ICOT-INTERNATIONAL-PARAMETERS.		
10 ICOT-OUTBD-INTL-VERSION-USE	X(01)	072 - 072
10 ICOT-OUTBD-INTL-VERSION	X(01)	073 - 073
10 ICOT-OUTBD-INTL-SYNTAX-ID	X(04)	074 - 077
10 ICOT-LAST-OUTBD-INTL-CNTL-NO	X(14)	078 - 091
05 ICOT-INTERCHG-OUTBD-DB-LEVEL	X(01)	092 - 092
05 ICOT-APPL-OUTBD-DB-LEVEL	X(01)	093 - 093
05 ICOT-MULTIPLE-ENVELOPE-ID	X(03)	094 - 096
05 ICOT-PARTNER-INT-VERSION	X912)	097 - 108

*PARTOUT FMS Transaction Layout*

ICOT-ACTION-CODE (Required)

Description: *Transaction Action Code*

Valid Value(s):

A	=	Add
C	=	Change
D	=	Delete

**Partner/Qualifier Mode**

ICOT-PARTNER-ID (Required)

Description: *Partner ID*

ICOT-PARTNER-ID-QUAL (Required)

Description: *Partner ID Qualifier*

ICOT-RECORD-TYPE (Required)

Description: *Transaction Type*

Valid Value(s): Constant PARTOUT (left-justified and space-filled).

### Relationship Mode

ICOT-USRPAR-USER-ID (Required)

Description: *User ID*

ICOT-USRPAR-PARTNER-ID (Required)

Description: *Partner ID*

ICOT-USRPAR-RECORD-TYPE (Required)

Description: *Transaction Type*

Valid Value(s): Constant PARTOUT (left-justified and space-filled).

ICOT-OUTBD-ICHG-VERSN-USE

Description: *Outbound Interchange Version Use*. Define the version to use for this partner.

Valid Value(s):

A	=	Always use this version for this partner
I	=	Use version from data
D	=	Use this version if the version cannot be found on input data

ICOT-OUTBD-INTERCHG-VERSION

Description: *Outbound Interchange Version*. Add the version to use when generating interchange header. This will also load into the ICS04 and ISA12, when applicable.

ICOT-LAST-OUTBD-ICHG-CNTL

Description: *Last Outbound Interchange Control*. Set up how Gentran checks the outbound interchange control number (ISA13, ICS11, or BG07).

Valid Value(s):

Zeros	=	Interchange control number checking starts at 1
Non-zero numerals	=	Interchange control number checking begins at n+1
Blank	=	Checking function is <i>not</i> used

ICOT-OUTBD-INTL-VERSION-USE

Description: *Outbound International Version Use*. Define the international version to use for this partner.

Valid Value(s):

A	=	Always use this version for this partner
I	=	Use version from data
D	=	Use this version if the version cannot be found on input data

ICOT-OUTBD-INTL-VERSION

Description: *Outbound International Version*. Define the version to use when generating Edifact and TRADACOMS interchange headers (UNB01:02 and STX01:02).

ICOT-OUTBD-INTL-SYNTAX-ID

Description: *Syntax Identifier*. Define the syntax identifier to use when generating Edifact and TRADACOMS interchange headers (UNB01:01 and STX01:01).

## ICOT-LAST-OUTBD-INTL-CNTL-NO

Description: *Last Outbound Interchange Control Number.* Define how Gentran generates Edifact and TRADACOMS interchange headers (UNB05 and STX05).

## ICOT-INTERCHG-OUTBD-DB-LEVEL

Description: *Interchange Outbound Databank Level.* Set the partner profile inbound EDI databanking level for directory maintenance.

Valid Value(s):

F	=	Full
D	=	Directory
N	=	None

## ICOT-APPL-OUTBD-DB-LEVEL

Description: *Application Outbound Databank Level.* Not currently used.

## ICOT-MULTIPLE-ENVELOPE-ID

Description: *Multiple Envelope ID.* Determine which interchange envelope to generate for outbound transmissions.

Valid Value(s):

ISA	=	Generate ISA, IEA control envelopes
ICS	=	Generate ICS, ICE control envelopes
BG	=	Generate BG-EG control envelopes
GS	=	Do not generate transmission control envelopes; use GS, GE instead
SCH	=	Generate SCH, END
STX	=	Generate STX, END
UNA	=	Generate UNA, UNB, UNZ control envelopes
UNB	=	Generate UNB, UNZ control envelopes

## ICOT-PARTNER-INT-VERSION

Description: *Version.* Use the partner interchange version when more than one control record with the same interchange header option is needed. Setting up partners with the partner interchange version should only be used when multiple like interchange envelopes need to be generated with different information.

**GENISA Record***Description*

The GENISA record controls outbound envelope generation for ISA partner types.

*Record Layout*

<b>Name</b>	<b>Data Type</b>	<b>Position</b>
01 IIAT-INPUT-PART-GEN-ISA-TRAN.		
05 IIAT-ACTION-CODE	X(01)	001 - 001
05 IIAT-PARTNER-KEY.		
10 IIAT-PARTNER-ID	X(35)	002 - 036
10 IIAT-PARTNER-ID-QUAL	X(04)	037 - 040
10 IIAT-RECORD-TYPE	X(16)	041 - 056
05 IIAT-USRPAR-KEY		
REDEFINES IIAT-PARTNER-KEY.		
10 IIAT-USRPAR-USER-ID	X(15)	002 - 016
10 IIAT-USRPAR-PARTNER-ID	X(15)	017 - 031
10 FILLER	X(09)	032 - 040
10 IIAT-USRPAR-RECORD-TYPE	X(16)	041 - 056
05 IIAT-ELEMENT-SEPARATOR.		
10 IIAT-ELEMENT-SEPARATOR-CHAR	X(01)	057 - 057
10 IIAT-ELEMENT-SEPARATOR-PART2	X(02)	058 - 059
05 IIAT-AUTHORIZATION-CODE-QUAL	X(02)	060 - 061
05 IIAT-AUTHORIZATION-CODE	X(10)	062 - 071
05 IIAT-SECURITY-CODE-QUAL	X(02)	072 - 073
05 IIAT-SECURITY-CODE	X(10)	074 - 083
05 IIAT-INTRCHG-SENDER-ID-QUAL	X(02)	084 - 085
05 IIAT-INTRCHG-SENDER-ID	X(15)	086 - 100
05 IIAT-INTRCHG-RECEIVER-ID-QUAL	X(02)	101 - 102
05 IIAT-INTRCHG-RECEIVER-ID	X(15)	103 - 117
05 IIAT-CONTROL-STDS-IDENTIFIER	X(01)	118 - 118
05 IIAT-INTERCHANGE-VERSION	X(12)	119 - 130
05 IIAT-TRANSMISSION-CONTROL-NUM	X(09)	131 - 139
05 IIAT-INTERCHANGE-ACKNLDG-REQST	X(01)	140 - 140
05 IIAT-ISA-TEST-INDICATOR	X(01)	141 - 141
05 IIAT-SUBELEMENT-SEPARATOR.		
10 IIAT-SUBELT-SEPARATOR-CHAR	X(01)	142 - 142
10 IIAT-SUBELT-SEPARATOR-PART2	X(02)	143 - 144
05 IIAT-SEGMENT-TERMINATOR.		
10 IIAT-SEGMENT-TERMINATOR-CHAR	X(01)	145 - 145
10 IIAT-SEGMENT-TERMINATOR-PART2	X(02)	146 - 147
05 IIAT-MULTIPLE-ENVELOPE-ID	X(03)	148 - 150
05 IIAT-PARTNER-INT-VERSION	X(12)	151 - 162
05 IIAT-REPETITION-SEP.		
10 IIAT-REPETITION-SEP-CHAR	X(01)	163 - 163
10 IIAT-REPETITION-SEP-PART2	X(02)	164 - 165



**GENISA FMS Transaction Layout Definition**

IIAT-ACTION-CODE (Required)

Description: *Transaction Action Code*

Valid Value(s):

A	=	Add
C	=	Change
D	=	Delete

**Partner/Qualifier Mode**

IIAT-PARTNER-ID (Required)

Description: *Partner ID*

IIAT-PARTNER-ID-QUAL (Required)

Description: *Partner ID Qualifier*

IIAT-RECORD-TYPE (Required)

Description: *Transaction Type*

Valid Value(s): Constant GENISA (left-justified and space-filled).

**Relationship Mode**

IIAT-USRPAR-USER-ID (Required)

Description: *User ID*

IIAT-USRPAR-PARTNER-ID (Required)

Description: *Partner ID*

IIAT-USRPAR-RECORD-TYPE (Required)

Description: *Transaction Type*

Valid Value(s): Constant GENISA (left-justified and space-filled).

IIAT-ELEMENT-SEPARATOR-CHAR

Description: *Element Separator Character*. Specify the single character for this element separator, unless the next field is non-blank.

Valid Value(s) (if the next field is non-blank):

H	=	The hexadecimal value for this element separator
---	---	--

IIAT-ELEMENT-SEPARATOR-PART2

Description: If this field is not blank, enter a valid 2-digit hexadecimal or octal number for the element separator.

Valid Value(s) (If position 57, specify H):

Blank
Valid hexadecimal number

IIAT-AUTHORIZATION-CODE-QUAL

Description: *Authorization Code Qualification*. Enter a Qualifier that Gentran can use to associate with the authorization code for the ISA (ISA01).

IIAT-AUTHORIZATION-CODE

Description: *Authorization Code*. Enter a code to use for the ISA (ISA02) authorization.

## IIAT-SECURITY-CODE-QUAL

Description: *Security Code Qualifier*. Enter a Qualifier to use as a security code for the ISA (ISA03).

## IIAT-SECURITY-CODE

Description: *Security Code*. Enter a code to use as a security code for ISA the (ISA04).

## IIAT-INTRCHG-SENDER-ID-QUAL

Description: *Interchange Sender ID Qualifier*. Enter a Qualifier for the interchange Sender ID code for the ISA (ISA05).

## IIAT-INTRCHG-SENDER-ID

Description: *Interchange Sender ID*. Enter a code to use as the interchange Sender ID code for ISA the (ISA06).

**Note:** This field supports the use of underscore substitution characters (USCs). To use a USC, the substitution character must exist on the Partner Header record.

See the *Gentran:Basic for zSeries Release 6.4 User's Guide* for more information about this feature.

## IIAT-INTRCHG-RECEIVER-ID-QUAL

Description: *Interchange Receiver ID Qualifier*. Enter a Qualifier for interchange Receiver ID code for the ISA (ISA07).

## IIAT-INTRCHG-RECEIVER-ID

Description: *Interchange Receiver ID*. Enter a code to use as the interchange Receiver ID code for the ISA (ISA08).

**Note:** This field supports the use of underscore substitution characters (USCs). To use a USC, the substitution character must exist on the Partner Header record.

See the *Gentran:Basic for zSeries Release 6.4 User's Guide* for more information about this feature.

## IIAT-CONTROL-STDS-IDENTIFIER

Description: *Control Standards Identifier*. Enter the control standards identifier for this partner the ISA (ISA11). Use this field to populate the ISA11 for ASC X12 version earlier than 004020. Use the IIAT-REPETITION-SEP-CHAR field to populate the ISA11 for ASC X12 versions 004020 and later.

## IIAT-INTERCHANGE-VERSION

Description: *Interchange Version*. Enter the version used for generating the ISA (ISA12).

## IIAT-TRANSMISSION-CONTROL-NUM

Description: *Transmission Control Number*. Enter the transmission control number for the ISA (ISA13).

## IIAT-INTERCHANGE-ACKNLDG-REQST

Description: *Interchange Acknowledgment Request*. Specify whether to generate the acknowledgment request on the ISA (ISA14).

Valid Value(s):

Zero	=	No
N	=	No
1	=	Yes
Y	=	Yes

## IIAT-ISA-TEST-INDICATOR

Description: *ISA Test Indicator*. Specify whether to generate the test indicator on ISA (ISA15).

Valid Value(s):

T	=	Test
P	=	Production
Y	=	Test
N	=	Production
Blank	=	Production

Default Value(s):N

## IIAT-SUBELT-SEPARATOR-CHAR

Description: *Subelement Separator Character*. Enter the subelement character for this partner.

## IIAT-SUBELT-SEPARATOR-PART2

Description: If this field is not blank, enter a valid 2-digit hexadecimal or octal number for the element separator.

Valid Value(s) (If position 142 specify H):

Valid hexadecimal number (143)

## IIAT-SEGMENT-TERMINATOR-CHAR

Description: *Segment Terminator Character*. Specify the single character for the segment terminator, unless the next field is non-blank.

Valid Value(s) (If the next field is non-blank):

H	=	The hexadecimal value to be used
---	---	----------------------------------

## IIAT-SEGMENT-TERMINATOR-PART2

Description: If this field is not blank, specify a valid 2-digit hexadecimal or octal number for the element separator.

Valid Value(s) (If position 145 specify H):

Blank  
Valid hexadecimal number

Valid Value(s) (If position 145 specify O):

Blank  
Valid octal number

## IIAT-MULTIPLE-ENVELOPE-ID

Description: *Multiple Envelope ID*. Determine which interchange envelope to generate for outbound transmissions.

Valid Value(s):

ISA	=	Generate ISA, IEA control envelopes
ICS	=	Generate ICS, ICE control envelopes
BG	=	Generate BG-EG control envelopes
GS	=	Do not generate transmission control envelopes; use GS, GE instead
SCH	=	Generate SCH, END
STX	=	Generate STX, END
UNA	=	Generate UNA, UNB, UNZ control envelopes
UNB	=	Generate UNB, UNZ control envelopes

## IIAT-PARTNER-INT-VERSION

Description: *Version*. Use the partner interchange version when more than one control record with the same interchange header option is needed. Setting up partners with the partner interchange version should only be used when multiple like interchange envelopes need to be generated with different information.

## IIAT-REPETITION-SEP-CHAR

Description: *Repetition Separator Character*. Specify the single character for this repetition separator, unless the next field is non-blank.

Valid Value (if the next field is non-blank):

H	=	The hexadecimal value for this repetition separator.
---	---	--

## IIAT-REPETITION-SEP-PART2

Description: If this field is not blank, enter a valid 2-digit hexadecimal number for the repetition separator.

Valid Value (if position 163 specifies H):

A valid hexadecimal number

**Note:** The IIAT-REPETITION-SEP-CHAR/IIAT-REPETITION-SEP-PART2 fields should be used to populate the ISA11 for ASC X12 version 004020 and higher. The IIAT-CONTROL-STANDARDS-IDENTIFIER field should be used to populate the ISA11 for ASC X12 versions earlier than 004020.

**GENICS Record***Description*

The GENICS record controls outbound envelope generation.

*Record Layout*

<b>Name</b>	<b>Data Type</b>	<b>Position</b>
01 IIST-INPUT-PART-GEN-ICS-TRAN.		
05 IIST-ACTION-CODE	X(01)	001 - 001
05 IIST-PARTNER-KEY.		
10 IIST-PARTNER-ID	X(35)	002 - 036
10 IIST-PARTNER-ID-QUAL	X(04)	037 - 040
10 IIST-RECORD-TYPE	X(16)	041 - 056
05 IIST-USRPAR-KEY		
REDEFINES IIST-PARTNER-KEY.		
10 IIST-USRPAR-USER-ID	X(15)	002 - 016
10 IIST-USRPAR-PARTNER-ID	X(15)	017 - 031
10 FILLER	X(09)	032 - 040
10 IIST-USRPAR-RECORD-TYPE	X(16)	041 - 056
05 IIST-ELEMENT-SEPARATOR.		
10 IIST-ELEMENT-SEPARATOR-CHAR	X(01)	057 - 057
10 IIST-ELEMENT-SEPARATOR-PART2	X(02)	058 - 059
05 IIST-SUBELEMENT-SEPARATOR.		
10 IIST-SUBELT-SEPARATOR-CHAR	X(01)	060 - 060
10 IIST-SUBELT-SEPARATOR-PART2	X(02)	061 - 062
05 IIST-INTERCHANGE-VERSION	X(12)	063 - 074
05 IIST-INTRCHG-SENDER-ID-QUAL	X(02)	075 - 076
05 IIST-INTRCHG-SENDER-ID	X(15)	077 - 091
05 IIST-INTRCHG-RECEIVER-ID-QUAL	X(02)	092 - 093
05 IIST-INTRCHG-RECEIVER-ID	X(15)	094 - 108
05 IIST-CONTROL-STDS-IDENTIFIER	X(01)	109 - 109
05 IIST-CONTROL-STDS-IDENT-LONG	X(04)	110 - 113
05 IIST-TRANSMISSION-CONTROL-NUM	X(09)	114 - 122
05 IIST-SEGMENT-TERMINATOR.		
10 IIST-SEGMENT-TERMINATOR-CHAR	X(01)	123 - 123
10 IIST-SEGMENT-TERMINATOR-PART2	X(02)	124 - 125
05 IIST-MULTIPLE-ENVELOPE-ID	X(03)	126 - 128
05 IIST-PARTNER-INT-VERSION	X(12)	129 - 140

*GENICS FMS Transaction Definitions***IIST-ACTION-CODE (Required)**

Description: *Transaction Action Code*

Valid Value(s):

A	=	Add
C	=	Change
D	=	Delete

**Partner/Qualifier Mode**

IIST-PARTNER-ID (Required)

Description: *Partner ID*

IIST-PARTNER-ID-QUAL (Required)

Description: *Partner ID Qualifier*

IIST-RECORD-TYPE (Required)

Description: *Transaction Type*

Valid Value(s):

Constant GENICS (left-justified and space-filled).

**Relationship Mode**

IIST-USRPAR-USER-ID (Required)

Description: *User ID*

IIST-USRPAR-PARTNER-ID (Required)

Description: *Partner ID*

IIST-USRPAR-RECORD-TYPE (Required)

Description: *Transaction Type*

Valid Value(s):

Constant GENICS (left-justified and space-filled).

IIST-ELEMENT-SEPARATOR-CHAR

Description: *Element Separator Character*. Enter the single character for this element separator, unless the next field is non-blank.

Valid Value(s) (If the next field is non-blank):

H = The hexadecimal value for this element separator

O = The octal value for this element separator

IIST-ELEMENT-SEPARATOR-PART2

Description: If this field is not blank, enter a valid 2-digit hexadecimal or octal number for the element separator.

Valid Value(s) (if position 57 specify H):

Blank

Valid hexadecimal number

Valid Value(s) (if position 57 specify O)

Blank

Valid octal number

IIST-SUBELT-SEPARATOR-CHAR

Description: *Subelement Separator Character*. Enter the subelement character chosen for this partner.

## IIST-SUBELT-SEPARATOR-PART2

Description: If this field is not blank, enter a valid 2-digit hexadecimal or octal number for the subelement separator.

Valid Value(s) (If position 60 specify H):

Blank  
Valid hexadecimal number

Valid Value(s) (If position 60 specify O):

Blank  
Valid octal number

## IIST-INTERCHANGE-VERSION

Description: *Interchange Version*. Enter the version to use when generating the interchange header.

## IIST-INTRCHG-SENDER-ID-QUAL

Description: *Interchange Sender ID Qualifier*. Enter the Qualifier for the interchange the ICS (ICS05).

## IIST-INTRCHG-SENDER-ID

Description: *Interchange Sender ID*. Enter the interchange Sender ID code for the (ICS06).

## IIST-INTRCHG-RECEIVER-ID-QUAL

Description: *Interchange Receiver ID Qualifier*. Enter the Qualifier for interchange Receiver ID code for the ICS (ICS07).

## IIST-INTRCHG-RECEIVER-ID

Description: *Interchange Receiver ID*. Enter the Interchange Receiver ID code for the ICS (ICS08).

## IIST-CONTROL-STDS-IDENTIFIER

Description: *Control Standards Identifier*. Enter the interchange control standards identifier for the ICS (ICS03) for this partner. Must be blank if the next field is present.

## IIST-CONTROL-STDS-IDENT-LONG

Description: *Control Standards Identifier Long*. Specify whether to generate the control standards identifier for the ICS (ICS03) for this partner. Must be blank if the prior field is present.

## IIST-TRANSMISSION-CONTROL-NUM

Description: *Transmission Control Number*. Enter the transmission control number for generating the ICS segment (ICS11).

## IIST-SEGMENT-TERMINATOR-CHAR

Description: *Segment Terminator Character*. Enter the single character for the segment terminator, unless the next field is non-blank.

Valid Value(s) (if the next field is non-blank):

H = The hexadecimal value for the segment terminator  
O = The octal value for the segment terminator

## IIST-SEGMENT-TERMINATOR-PART2

Description: If this field is not blank, enter a valid 2-digit hexadecimal or octal number for the element separator.

Valid Value(s) (if position 123 specify H):

Blank  
Valid hexadecimal number

Valid Value(s) (if position 123 specify O):

Blank  
Valid octal number

## IIST-MULTIPLE-ENVELOPE-ID

Description: *Multiple Envelope ID*. Determine which interchange envelope to generate for outbound transmissions.

Valid Value(s):

ISA	=	Generate ISA, IEA control envelopes
ICS	=	Generate ICS, ICE control envelopes
BG	=	Generate BG-EG control envelopes
GS	=	Do not generate transmission control envelopes; use GS, GE instead
SCH	=	Generate SCH, END
STX	=	Generate STX, END
UNA	=	Generate UNA, UNB, UNZ control envelopes
UNB	=	Generate UNB, UNZ control envelopes

## IIST-PARTNER-INT-VERSION

Description: *Version*. Use the partner interchange version when more than one control record with the same interchange header option is needed. Setting up partners with the partner interchange version should only be used when multiple like interchange envelopes need to be generated with different information.



**GENBG Record***Description*

The GENBG record controls outbound envelope generation.

*Record Layout*

<b>Name</b>	<b>Data Type</b>	<b>Position</b>
01 IBGT-INPUT-PART-GEN-BG-TRAN.		
05 IBGT-ACTION-CODE	X(01)	001 - 001
05 IBGT-PARTNER-KEY.		
10 IBGT-PARTNER-ID	X(35)	002 - 036
10 IBGT-PARTNER-ID-QUAL	X(04)	037 - 040
10 IBGT-RECORD-TYPE	X(16)	041 - 056
05 IBGT-USRPAR-KEY		
REDEFINES IBGT-PARTNER-KEY.		
10 IBGT-USRPAR-USER-ID	X(15)	002 - 016
10 IBGT-USRPAR-PARTNER-ID	X(15)	017 - 031
10 FILLER	X(09)	032 - 040
10 IBGT-USRPAR-RECORD-TYPE	X(16)	041 - 056
05 IBGT-ELEMENT-SEPARATOR.		
10 IBGT-ELEMENT-SEPARATOR-CHAR	X(01)	057 - 057
10 IBGT-ELEMENT-SEPARATOR-PART2	X(02)	058 - 059
05 IBGT-INTRCHG-SENDER-ID	X(15)	060 - 074
05 IBGT-INTRCHG-RECEIVER-ID	X(15)	075 - 089
05 IBGT-SECURITY-CODE	X(10)	090 - 099
05 IBGT-AUTHORIZATION-CODE	X(10)	100 - 109
05 IBGT-TRANSMISSION-CONTROL-NUM	X(09)	110 - 118
05 IBGT-SEGMENT-TERMINATOR.		
10 IBGT-SEGMENT-TERMINATOR-CHAR	X(01)	119 - 119
10 IBGT-SEGMENT-TERMINATOR-PART2	X(02)	120 - 121
05 IBGT-MULTIPLE-ENVELOPE-ID	X(03)	122 - 124
05 IBGT-PARTNER-INT-VERSION	X(12)	125 - 136

**GENBG FMS Transaction Definitions****IBGT-ACTION-CODE (Required)**

Description: *Transaction Action Code*

Valid Value(s):

A	=	Add
C	=	Change
D	=	Delete

**Partner/Qualifier Mode****IBGT-PARTNER-ID (Required)**

Description: *Partner ID*

**IBGT-PARTNER-ID-QUAL (Required)**

Description: *Partner ID Qualifier*

## IBGT-RECORD-TYPE (Required)

Description: *Transaction Type*

Valid Value(s):

Constant GENBG (left-justified and space-filled).

**Relationship Mode**

## IBGT-USRPAR-USER-ID (Required)

Description: *User ID*

## IBGT-USRPAR-PARTNER-ID (Required)

Description: *Partner ID*

## IBGT-USRPAR-RECORD-TYPE (Required)

Description: *Transaction Type*

Valid Value(s):

Constant GENBG (left-justified and space-filled).

## IBGT-ELEMENT-SEPARATOR-CHAR

Description: *Element Separator Character*. Enter the single character for the element separator, unless the next field is non-blank.

Valid Value(s) (If the next field is non-blank):

H = The hexadecimal value for the element separator

O = The octal value for the element separator

## IBGT-ELEMENT-SEPARATOR-PART2

Description: If this field is not blank, enter a valid 2-digit hexadecimal or octal number for the element separator.

Valid Value(s) (If position 57 specify H):

Blank

Valid hexadecimal number

Valid Value(s) (If position 57 specify O):

Blank

Valid octal number

## IBGT-INTRCHG-SENDER-ID

Description: *Interchange Sender ID*. Identify the sender to use when generating the BG segment (BG03).

## IBGT-INTRCHG-RECEIVER-ID

Description: *Interchange Receiver ID*. Identify the receiver to use when generating the BG segment (BG04).

## IBGT-SECURITY-CODE

Description: *Security Code*. Enter the security code for BG (BG01).

## IBGT-AUTHORIZATION-CODE

Description: *Authorization Code*. Enter the authorization code for BG(BG02).

## IBGT-TRANSMISSION-CONTROL-NUM

Description: *Transmission Control Number*. Enter the transmission control number used to generate the BG segment (BG07).

## IBGT-SEGMENT-TERMINATOR-CHAR

Description: *Segment Terminator Character*. Enter the single character for the segment terminator, unless the next field is non-blank.

Valid Value(s) (If the next field is non-blank):

H	=	The hexadecimal value for the segment terminator
O	=	The octal value for the segment terminator

## IBGT-SEGMENT-TERMINATOR-PART2

Description: If this field is not blank, enter a valid 2-digit hexadecimal or octal number for the element separator.

Valid Value(s) (If position 123 specify H):

Blank  
Valid hexadecimal number

Valid Value(s) (If position 123 specify O):

Blank  
Valid octal number

## IBGT-MULTIPLE-ENVELOPE-ID

Description: *Multiple Envelope ID*. Determine which interchange envelope to generate for outbound transmissions.

Valid Value(s):

ISA	=	Generate ISA, IEA control envelopes
ICS	=	Generate ICS, ICE control envelopes
BG	=	Generate BG-EG control envelopes
GS	=	Do not generate transmission control envelopes; use GS, GE instead
SCH	=	Generate SCH, END
STX	=	Generate STX, END
UNA	=	Generate UNA, UNB, UNZ control envelopes
UNB	=	Generate UNB, UNZ control envelopes

## IBGT-PARTNER-INT-VERSION

Description: *Version*. Use the partner interchange version when more than one control record with the same interchange header option is selected. Setting up partners with the partner interchange version should only be used when multiple like interchange envelopes need to be generated with different information.

**GENUNA Record***Description*

The GENUNA record controls outbound envelope generation.

*Record Layout*

Name	Data Type	Position
01 IGAT-INPUT-PART-GEN-UNA-TRAN.		
05 IGAT-ACTION-CODE	X(01)	001 - 001
05 IGAT-PARTNER-KEY.		
10 IGAT-PARTNER-ID	X(35)	002 - 036
10 IGAT-PARTNER-ID-QUAL	X(04)	037 - 040
10 IGAT-RECORD-TYPE	X(16)	041 - 056
05 IGAT-USRPAR-KEY		
REDEFINES IGAT-PARTNER-KEY.		
10 IGAT-USRPAR-USER-ID	X(15)	002 - 016
10 IGAT-USRPAR-PARTNER-ID	X(15)	017 - 031
10 FILLER	X(09)	032 - 040
10 IGAT-USRPAR-RECORD-TYPE	X(16)	041 - 056
05 IGAT-UNA-COMPONENT-ELT-SEP.		
10 IGAT-UNA-COMPONENT-ELT-SEP-CHAR	X(01)	057 - 057
10 IGAT-UNA-COMPONENT-ELT-SEP-PART2	X(02)	058 - 059
05 IGAT-UNA-DATA-ELEMENT-SEP.		
10 IGAT-UNA-DATA-ELT-SEP-CHAR	X(01)	060 - 060
10 IGAT-UNA-DATA-ELT-SEP-PART2	X(02)	061 - 062
05 IGAT-UNA-DECIMAL-NOTATION	X(01)	063 - 063
05 IGAT-UNA-RELEASE-INDICATOR	X(01)	064 - 064
05 FILLER	X(01)	065 - 065
05 IGAT-UNA-SEGMENT-TERMINATOR.		
10 IGAT-UNA-SEGMENT-TERM-CHAR	X(01)	066 - 066
10 IGAT-UNA-SEGMENT-TERM-PART2	X(02)	067 - 068
05 IGAT-MULTIPLE-ENVELOPE-ID	X(03)	069 - 071
05 IGAT-PARTNER-INT-VERSION	X(12)	072 - 183
05 IGAT-UNA-REPETITION-SEP.		
10 IGAT-UNA-REPETITION-SEP-CHAR	X(01)	084 - 084
10 IGAT-UNA-REPETITION-SEP-PART2	X(02)	085 - 086

*GENUNA FMS Transaction Definitions***IGAT-ACTION-CODE (Required)**

Description: *Transaction Action Code*

Valid Value(s):

A	=	Add
C	=	Change
D	=	Delete

**Partner/Qualifier Mode****IGAT-PARTNER-ID (Required)**

Description: *Partner ID*

IGAT-PARTNER-ID-QUAL (Required)

Description: *Partner ID Qualifier*

IGAT-RECORD-TYPE (Required)

Description: *Transaction Type*

Valid Value(s):

Constant GENUA (left-justified and space-filled).

### Relationship Mode

IGAT-USRPAR-USER-ID (Required)

Description: *User ID*

IGAT-USRPAR-PARTNER-ID (Required)

Description: *Partner ID*

IGAT-USRPAR-RECORD-TYPE (Required)

Description: *Transaction Type*

Valid Value(s):

Constant GENUNA (left-justified and space-filled).

IGAT-UNA-COMPONT-ELT-SEP-CHAR

Description: *Component Element Separator Character.* Enter the single character for the subelement separator, unless the next field is non-blank.

Valid Value(s) (If the next field is non-blank):

H	=	The hexadecimal value for the subelement separator
O	=	The octal value for the subelement separator

IGAT-UNA-COMPONT-ELT-SEP-PART2

Description: If this field is not blank, enter a valid 2-digit hexadecimal or octal number for the element separator.

Valid Value(s) (If position 57 specify H):

Blank  
Valid hexadecimal number

Valid Value(s) (If position 57 specify O):

Blank  
Valid octal number

IGAT-UNA-DATA-ELT-SEP-CHAR

Description: *Data Element Separator Character.* Enter the single character for the element separator, unless the next field is non-blank.

Valid Value(s) (If the next field is non-blank):

H	=	The hexadecimal value for the element separator
O	=	The octal value for the element separator

## IGAT-UNA-DATA-ELT-SEP-PART2

Description: If this field is not blank, enter a valid 2-digit hexadecimal or octal number for the element separator.

Valid Value(s) (If position 60 specify H):

Blank  
Valid hexadecimal number

Valid Value(s) (If position 60 specify O):

Blank  
Valid octal number

## IGAT-UNA-DECIMAL-NOTATION

Description: *Decimal Notation.* Leave this field blank, or enter a comma, period, or exclamation point, to generate the decimal notation in UNA (UNA03).

Valid Value(s):

. = Period  
, = Comma  
Space = Period

## IGAT-UNA-RELEASE-INDICATOR

Description: *Release Indicator.* Specify whether to generate the release indicator in the UNA segment (UNA04).

## IGAT-UNA-SEGMENT-TERM-CHAR

Description: *Segment Terminator Character.* Enter the single character for the repetition separator, unless the next field is non-blank.

Valid Value(s) (If the next field is non-blank):

H = The hexadecimal value for the repetition separator  
O = The octal value for the repetition separator

## IGAT-UNA-SEGMENT-TERM-PART2

Description: If this field is not blank, Enter a valid 2-digit hexadecimal or octal number for the element separator.

Valid Value(s) (If position 66 specify H):

Blank  
Valid hexadecimal number

Valid Value(s) (If position 66 specify O)

Blank  
Valid octal number

## IGAT-MULTIPLE-ENVELOPE-ID

Description: *Multiple Envelope ID.* Determine which interchange envelope to generate for outbound transmissions.

Valid Value(s):

ISA = Generate ISA, IEA control envelopes  
ICS = Generate ICS, ICE control envelopes  
BG = Generate BG-EG control envelopes  
GS = Do not generate transmission control envelopes; use GS, GE instead  
SCH = Generate SCH, END  
STX = Generate STX, END

UNA = Generate UNA, UNB, UNZ control envelopes  
UNB = Generate UNB, UNZ control envelopes

**IGAT-PARTNER-INT-VERSION**

Description: *Version.* Use the partner interchange version when more than one control record with the same interchange header option is needed. Setting up partners with the partner interchange version should only be used when multiple like interchange envelopes need to be generated with different information.

**IGAT-UNA-REPETITION-SEP-CHAR**

Description: *Repetition Separator Character.* Specify the single character for this repetition separator, unless the next field is non-blank.

Valid Value (if the next field is non-blank):

H = The hexadecimal value for this repetition separator

**IAGT-UNA-REPETITION-SEP-PART2**

Description: If this field is not blank, enter a valid 2-digit hexadecimal number for the repetition separator.

Valid Value (if position 84 specifies H):

Valid hexadecimal number

**GENUNB1 Record***Description*

The GENUNB1, GENUNB2, GENUNB3, and GENUNB4 records control outbound envelope generation.

*Record Layout*

Name	Data Type	Position
01 IB1T-INPUT-PART-GEN-UNB1-TRAN.		
05 IB1T-ACTION-CODE	X(01)	001 - 001
05 IB1T-PARTNER-KEY.		
10 IB1T-PARTNER-ID	X(35)	002 - 036
10 IB1T-PARTNER-ID-QUAL	X(04)	037 - 040
10 IB1T-RECORD-TYPE	X(16)	041 - 056
05 IB1T-USRPAR-KEY		
REDEFINES IB1T-PARTNER-KEY.		
10 IB1T-USRPAR-USER-ID	X(15)	002 - 016
10 IB1T-USRPAR-PARTNER-ID	X(15)	017 - 031
10 FILLER	X(09)	032 - 040
10 IB1T-USRPAR-RECORD-TYPE	X(16)	041 - 056
05 IB1T-UNB-SYNTAX-IDENTIFIER	X(04)	057 - 060
05 IB1T-UNB-SYNTAX-VERSION	X(01)	061 - 061
05 IB1T-UNB-CONTROL-REFERENCE	X(14)	062 - 075
05 IB1T-UNB-PASSWORD	X(14)	076 - 089
05 IB1T-UNB-PASSWORD-QUAL	X(02)	090 - 091
05 IB1T-UNB-APPLICATION-REFERENCE	X(14)	092 - 105
05 IB1T-UNB-PROCESSING-PRTY-CODE	X(01)	106 - 106
05 IB1T-UNB-ACKNOWLEDGEMENT-REQ	X(01)	107 - 107
05 IB1T-UNB-COMM-AGREEMENT-ID	X(35)	108 - 142
05 IB1T-UNB-TEST-INDICATOR	X(01)	143 - 143
05 IB1T-UNG-MULTIPLE-ENVELOPE-ID	X(03)	144 - 146
05 IB1T-PARTNER-INT-VERSION	X(12)	147 - 158

*GENUNB1 FMS Transaction Definitions***IB1T-ACTION-CODE (Required)**

Description: *Transaction Action Code*

Valid Value(s):

A	=	Add
C	=	Change
D	=	Delete

**Partner/Qualifier Mode****IB1T-PARTNER-ID (Required)**

Description: *Partner ID*

**IB1T-PARTNER-ID-QUAL (Required)**

Description: *Partner ID Qualifier*



IB1T-RECORD-TYPE (Required)

Description: *Transaction Type*

Valid Value(s):

Constant GENUNB1 (left-justified and space-filled).

**Relationship Mode**

IB1T-USRPAR-USER-ID (Required)

Description: *User ID*

IB1T-USRPAR-PARTNER-ID (Required)

Description: *Partner ID*

IB1T-USRPAR-RECORD-TYPE (Required)

Description: *Transaction Type*

Valid Value(s):

Constant GENUNB1 (left-justified and space-filled).

IB1T-UNB-SYNTAX-IDENTIFIER

Description: *Syntax Identifier*. Enter the syntax identifier to use when generating the UNB (UNB01:01).

IB1T-UNB-SYNTAX-VERSION

Description: *Syntax Version*. Enter the syntax identifier to use when generating the UNB (UNB01:02).

IB1T-UNB-CONTROL-REFERENCE

Description: *Control Reference*. Enter the control reference to use when generating the UNB (UNB05).

IB1T-UNB-PASSWORD

Description: *Password*. Enter the password to use when generating the UNB (UNB06:01).

IB1T-UNB-PASSWORD-QUAL

Description: *Password Qualifier*. Enter the password Qualifier to use when generating the UNB (UNB06:02).

IB1T-UNB-APPLICATION-REFERENCE

Description: *Application Reference*. Enter the application reference to use when generating the UNB (UNB07).

IB1T-UNB-PROCESSING-PRTY-CODE

Description: *Processing Priority Code*. Enter the processing priority code to use to generate the UNB (UNB08).

## IB1T-UNB-ACKNOWLEDGEMENT-REQ

Description: *Acknowledgement Request*. Specify the acknowledgment request to use to generate the UNB (UNB09).

Valid Value(s):

1	=	Yes
Y	=	Yes
Space	=	No

## IB1T-UNB-COMM-AGREEMENT-ID

Description: *Communications Agreement ID*. Enter the Communications Agreement ID to use when generating the UNB.

## IB1T-UNB-TEST-INDICATOR

Description: *Test Indicator*. Specify whether to test processing after generating the UNB (UNB11).

Valid Value(s):

1	=	Yes
Y	=	Yes
Space	=	No

## IB1T-MULTIPLE-ENVELOPE-ID

Description: *Multiple Envelope ID*. Determine which interchange envelope to generate for outbound transmissions.

Valid Value(s):

ISA	=	Generate ISA, IEA control envelopes
ICS	=	Generate ICS, ICE control envelopes
BG	=	Generate BG-EG control envelopes
GS	=	Do not generate transmission control envelopes; use GS, GE instead
SCH	=	Generate SCH, END
STX	=	Generate STX, END
UNA	=	Generate UNA, UNB, UNZ control envelopes
UNB	=	Generate UNB, UNZ control envelopes

## IB1T-PARTNER-INT-VERSION

Description: *Version*. Use the partner interchange version when more than one control record with the same interchange header option is needed. Setting up partners with the partner interchange version should only be used when multiple like interchange envelopes need to be generated with different information.

**GENUNB2 Record***Description*

The GENUNB1, GENUNB2, GENUNB3, and GENUNB4 records control outbound envelope generation.

*Record Layout*

Name	Data Type	Position
01 IB2T-INPUT-PART-GEN-UNB2-TRAN.		
05 IB2T-ACTION-CODE	X(01)	001 - 001
05 IB2T-PARTNER-KEY.		
10 IB2T-PARTNER-ID	X(35)	002 - 036
10 IB2T-PARTNER-ID-QUAL	X(04)	037 - 040
10 IB2T-RECORD-TYPE	X(16)	041 - 056
05 IB2T-USRPAR-PARTNER-KEY		
REDEFINES IB2T-PARTNER-KEY.		
10 IB2T-USRPAR-USER-ID	X(15)	002 - 016
10 IB2T-USRPAR-PARTNER-ID	X(15)	017 - 031
10 FILLER	X(09)	032 - 040
10 IB2T-USRPAR-RECORD-TYPE	X(16)	041 - 056
05 IB2T-UNB-SENDER-ID	X(35)	057 - 091
05 IB2T-UNB-SENDER-ID-QUAL	X(04)	092 - 095
05 IB2T-UNB-REVERSE-ROUTE-ADDR	X(14)	096 - 109
05 IB2T-MULTIPLE-ENVELOPE-ID	X(03)	110 - 112
05 IB2T-PARTNER-INT-VERSION	X(12)	113 - 124

**GENUNB2 FMS Transaction Definitions**

IB2T-ACTION-CODE (Required)

Description: *Transaction Action Code*

Valid Value(s):

A	=	Add
C	=	Change
D	=	Delete

**Partner/Qualifier Mode**

IB2T-PARTNER-ID (Required)

Description: *Partner ID*

IB2T-PARTNER-ID-QUAL (Required)

Description: *Partner ID Qualifier*

**Relationship Mode**

IB2T-USRPAR-USER-ID (Required)

Description: *User ID*

IB2T-USRPAR-PARTNER-ID (Required)

Description: *Partner ID*

## IB2T-USRPAR-RECORD-TYPE (Required)

Description: *Transaction Type*

Valid Value(s):

Constant GENUNB2 (left-justified and space-filled).

## IB2T-UNB-SENDER-ID

Description: *Sender ID*. Enter the Sender ID to use when generating the UNB segment (UNB02:01).

## IB2T-UNB-SENDER-ID-QUAL

Description: *Sender ID Qualifier*. Enter the Sender ID and Qualifier to use to generate the UNB segment (UNB02:02).

**Note:** This field supports the use of underscore substitution characters (USCs). To use a USC, the substitution character must exist on the Partner Header record.

See the *Gentran:Basic for zSeries Release 6.4 User's Guide* for more information about this feature.

## IB2T-UNB-REVERSE-ROUTE-ADDR

Description: *Reverse Route Address*. Enter the reverse routing address to use when generating the UNB segment (UNB02:03).

**Note:** This field supports the use of underscore substitution characters (USCs). To use a USC, the substitution character must exist on the Partner Header record.

See the *Gentran:Basic for zSeries Release 6.4 User's Guide* for more information about this feature.

## IB2T-MULTIPLE-ENVELOPE-ID

Description: *Multiple Envelope ID*. Determine which interchange envelope to generate for outbound transmissions.

Valid Value(s):

ISA	=	Generate ISA, IEA control envelopes
ICS	=	Generate ICS, ICE control envelopes
BG	=	Generate BG-EG control envelopes
GS	=	Do not generate transmission control envelopes; use GS, GE instead
SCH	=	Generate SCH, END
STX	=	Generate STX, END
UNA	=	Generate UNA, UNB, UNZ control envelopes
UNB	=	Generate UNB, UNZ control envelopes

## IB2T-PARTNER-INT-VERSION

Description: *Version*. Use the partner interchange version when more than one control record with the same interchange header option is needed. Setting up partners with the partner interchange version should only be used when multiple like interchange envelopes need to be generated with different information.

**GENUNB3 Record***Description*

The GENUNB1, GENUNB2, GENUNB3, and GENUNB4 records control outbound envelope generation.

*Record Layout*

Name	Data Type	Position
01 IB3T-INPUT-PART-GEN-UNB3-TRAN.		
05 IB3T-ACTION-CODE	X(01)	001 - 001
05 IB3T-PARTNER-KEY.		
10 IB3T-PARTNER-ID	X(35)	002 - 036
10 IB3T-PARTNER-ID-QUAL	X(04)	037 - 040
10 IB3T-RECORD-TYPE	X(16)	041 - 056
05 IB3T-USRPAR-KEY		
REDEFINES IB3T-PARTNER-KEY.		
10 IB3T-USRPAR-USER-ID	X(15)	002 - 016
10 IB3T-USRPAR-PARTNER-ID	X(15)	017 - 031
10 FILLER	X(09)	032 - 040
10 IB3T-USRPAR-RECORD-TYPE	X(16)	041 - 056
05 IB3T-UNB-RECEIVER-ID	X(35)	057 - 091
05 IB3T-UNB-RECEIVER-ID-QUAL	X(04)	092 - 095
05 IB3T-UNB-RECEIVER-ROUTE-ADDR	X(14)	096 - 109
05 IB3T-MULTIPLE-ENVELOPE-ID	X(03)	110 - 112
05 IB3T-PARTNER-INT-VERSION	X(12)	113 - 124

*GENUNB3 FMS Transaction Definitions***IB3T-ACTION-CODE (Required)**

Description: *Transaction Action Code*

Valid Value(s)

A	=	Add
C	=	Change
D	=	Delete

**Partner/Qualifier Mode****IB3T-PARTNER-ID (Required)**

Description: *Partner ID*

**IB3T-PARTNER-ID-QUAL (Required)**

Description: *Partner ID Qualifier*

**IB3T-RECORD-TYPE (Required)**

Description: *Transaction Type*

Valid Value(s):

Constant GENUNB3 (left-justified and space-filled).

**Relationship Mode**

IB3T-USRPAR-USER-ID (Required)

Description: *User ID*

IB3T-USRPAR-PARTNER-ID (Required)

Description: *Partner ID*

IB3T-USRPAR-RECORD-TYPE (Required)

Description: *Transaction Type*

Valid Value(s):

Constant GENUNB3 (left-justified and space-filled)

IB3T-UNB-RECEIVER-ID

Description: *Receiver ID*. Enter the ID of the receiver to use when generating the UNB segment (UNB03:01).

**Note:** This field supports the use of underscore substitution characters (USCs). To use a USC, the substitution character must exist on the Partner Header record.

See the *Gentran:Basic for zSeries Release 6.4 User's Guide* for more information about this feature.

IB3T-UNB-RECEIVER-ID-QUAL

Description: *Receiver ID Qualifier*. Enter the ID of the receiver and the Qualifier to use when generating the UNB segment (UNB03:02).

IB3T-UNB-RECEIVER-ROUTE-ADDR

Description: *Receiver Route Address*. Enter the receiver routing address to use when generating the UNB segment (UNB03:03).

**Note:** This field supports the use of underscore substitution characters (USCs). To use a USC, the substitution character must exist on the Partner Header record.

See the *Gentran:Basic for zSeries Release 6.4 User's Guide* for more information about this feature.

IB3T-MULTIPLE-ENVELOPE-ID

Description: *Multiple Envelope ID*. Determine which interchange envelope to use when generating outbound transmissions.

Valid Value(s):

ISA	=	Generate ISA, IEA control envelopes
ICS	=	Generate ICS, ICE control envelopes
BG	=	Generate BG-EG control envelopes
GS	=	Do not generate transmission control envelopes; use GS, GE instead
SCH	=	Generate SCH, END
STX	=	Generate STX, END
UNA	=	Generate UNA, UNB, UNZ control envelopes
UNB	=	Generate UNB, UNZ control envelopes

## IB3T-PARTNER-INT-VERSION

Description: *Version.* Use the partner interchange version when more than one control record with the same interchange header option is needed. Setting up partners with the partner interchange version should only be used when multiple like interchange envelopes need to be generated with different information.

**GENUNB4 Record***Description*

The GENUNB1, GENUNB2, GENUNB3, and GENUNB4 records control outbound envelope generation.

*Record Layout*

Name	Data Type	Position
01 IB4T-INPUT-PART-GEN-UNB4-TRAN.		
05 IB4T-ACTION-CODE	X (01)	001 - 001
05 IB4T-PARTNER-KEY.		
10 IB4T-PARTNER-ID	X (35)	002 - 036
10 IB4T-PARTNER-ID-QUAL	X (04)	037 - 040
10 IB4T-RECORD-TYPE	X (16)	041 - 056
05 IB4T-USRPAR-KEY		
REDEFINES IB4T-PARTNER-KEY.		
10 IB4T-USRPAR-USER-ID	X (15)	002 - 016
10 IB4T-USRPAR-PARTNER-ID	X (15)	017 - 031
10 FILLER	X (09)	032 - 040
10 IB4T-USRPAR-RECORD-TYPE	X (16)	041 - 056
05 IB4T-UNB-CODE-LIST-VERSION	X (06)	057 - 062
05 IB4T-UNB-CHARACTER-ENCODING	X (03)	063 - 065
05 IB4T-UNB-SENDER-INTERNAL-ID	X (35)	066 - 100
05 IB4T-UNB-SENDER-INT-SUB-ID	X (35)	101 - 135
05 IB4T-UNB-RECEIVER-INTERNAL-ID	X (35)	136 - 170
05 IB4T-UNB-RECEIVER-INT-SUB-ID	X (35)	171 - 205
05 IB4T-MULTIPLE-ENVELOPE-ID	X (03)	206 - 208
05 IB4T-PARTNER-INT-VERSION	X (12)	209 - 220

**GENUNB4 FMS Transaction Definitions****IB4T-ACTION-CODE (Required)**

Description: *Transaction Action Code*

Valid Value(s):

A	=	Add
C	=	Change
D	=	Delete

**Partner/Qualifier Mode****IB4T-PARTNER-ID (Required)**

Description: *Partner ID*

**IB4T-PARTNER-ID-QUAL (Required)**

Description: *Partner ID Qualifier*

**IB4T-RECORD-TYPE (Required)**

Description: *Transaction Type*

Valid Value(s):

Constant GENUNB4 (left-justified and space-filled).



**Relationship Mode**

IB4T-USRPAR-USER-ID (Required)

Description: *User ID*

IB4T-USRPAR-PARTNER-ID (Required)

Description: *Partner ID*

IB4T-USRPAR-RECORD-TYPE (Required)

Description: *Transaction Type*

Valid Value(s):

Constant GENUNB4 (left-justified and space-filled).

IB4T-UNB-CODE-LIST-VERSION

Description: *Code List Version*. Define the service code list directory version number.

IB4T-UNB-CHARACTER-ENCODING

Description: *Character Encoding*. Define the character encoding value.

IB4T-UNB-SENDER-INTERNAL-ID

Description: *Sender Internal ID*. Define the interchange Sender Internal ID.

**Note:** This field supports the use of underscore substitution characters (USCs). To use a USC, the substitution character must exist on the Partner Header record.

See the *Gentran:Basic for zSeries Release 6.4 User's Guide* for more information about this feature.

IB4T-UNB-SENDER-INT-SUB-ID

Description: *Sender Internal SubID*. Define the interchange Sender Internal Sub-ID.

**Note:** This field supports the use of underscore substitution characters (USCs). To use a USC, the substitution character must exist on the Partner Header record.

See the *Gentran:Basic for zSeries Release 6.4 User's Guide* for more information about this feature.

IB4T-UNB-RECEIVER-INTERNAL-ID

Description: *Receiver Internal ID*. Define the interchange Recipient Internal ID.

**Note:** This field supports the use of underscore substitution characters (USCs). To use a USC, the substitution character must exist on the Partner Header record.

See the *Gentran:Basic for zSeries Release 6.4 User's Guide* for more information about this feature.

IB4T-UNB-RECEIVER-INT-SUB-ID

Description: *Receiver Internal SubID*. Define the interchange Receiver Internal Sub-ID.

**Note:** This field supports the use of underscore substitution characters (USCs). To use a USC, the substitution character must exist on the Partner Header record.

See the *Gentran:Basic for zSeries Release 6.4 User's Guide* for more information about this feature.

**IB4T-MULTIPLE-ENVELOPE-ID**

Description: *Multiple Envelope ID*. Determine which interchange envelope to generate for outbound transmissions.

Valid Value(s):

ISA	=	Generate ISA, IEA control envelopes
ICS	=	Generate ICS, ICE control envelopes
BG	=	Generate BG-EG control envelopes
GS	=	Do not generate transmission control envelopes; use GS, GE instead
SCH	=	Generate SCH, END
STX	=	Generate STX, END
UNA	=	Generate UNA, UNB, UNZ control envelopes
UNB	=	Generate UNB, UNZ control envelopes

**IB4T-PARTNER-INT-VERSION**

Description: *Version*. Use the partner interchange version when more than one control record with the same interchange header option is needed. Setting up partners with the partner interchange version should only be used when multiple like interchange envelopes need to be generated with different information.

**GENSCH Record***Description*

The GENSCH record controls outbound envelope generation.

*Record Layout*

<b>Name</b>	<b>Data Type</b>	<b>Position</b>
01 IGBT-INPUT-PART-GEN-SCH-TRAN.		
05 IGBT-ACTION-CODE	X(01)	001 - 001
05 IGBT-PARTNER-KEY.		
10 IGBT-PARTNER-ID	X(35)	002 - 036
10 IGBT-PARTNER-ID-QUAL	X(04)	037 - 040
10 IGBT-RECORD-TYPE	X(16)	041 - 056
05 IGBT-USRPAR-KEY		
REDEFINES IGBT-PARTNER-KEY.		
10 IGBT-USRPAR-USER-ID	X(15)	002 - 016
10 IGBT-USRPAR-PARTNER-ID	X(15)	017 - 031
10 FILLER	X(09)	032 - 040
10 IGBT-USRPAR-RECORD-TYPE	X(16)	041 - 056
05 IGBT-SCH-SEGMENT-SEPARATOR.		
10 IGBT-SCH-SEGMENT-SEP-CHAR	X(01)	057 - 057
10 IGBT-SCH-SEGMENT-SEP-PART2	X(02)	058 - 059
05 IGBT-SCH-COMPONENT-ELT-SEP.		
10 IGBT-SCH-COMPONT-ELT-SEP-CHAR	X(01)	060 - 060
10 IGBT-SCH-COMPONT-ELT-SEP-PART2	X(02)	061 - 062
05 IGBT-SCH-DATA-ELEMENT-SEP.		
10 IGBT-SCH-DATA-ELT-SEP-CHAR	X(01)	063 - 063
10 IGBT-SCH-DATA-ELT-SEP-PART2	X(02)	064 - 065
05 IGBT-SCH-SEGMENT-TERMINATOR.		
10 IGBT-SCH-SEGMENT-TERM-CHAR	X(01)	066 - 066
10 IGBT-SCH-SEGMENT-TERM-PART2	X(02)	067 - 068
05 IGBT-SCH-RELEASE-INDICATOR	X(01)	069 - 069
05 IGBT-MULTIPLE-ENVELOPE-ID	X(03)	070 - 072
05 IGBT-PARTNER-INT-VERSION	X(12)	073 - 084

*GENSCH FMS Transaction Definitions***IGBT-ACTION-CODE (Required)**

Description: *Transaction Action Code*

Valid Value(s):

A	=	Add
C	=	Change
D	=	Delete

**Partner/Qualifier Mode**

**IGBT-PARTNER-ID (Required)**

Description: *Partner ID*

IGBT-PARTNER-ID-QUAL (Required)

Description: *Partner ID Qualifier*

IGBT-RECORD-TYPE (Required)

Description: *Transaction Type*

Valid Value(s):

Constant GENSCH (left-justified and space-filled).

### Relationship Mode

IGBT-USRPAR-USER-ID (Required)

Description: *User ID*

IGBT-USRPAR-PARTNER-ID (Required)

Description: *Partner ID*

IGBT-USRPAR-RECORD-TYPE (Required)

Description: *Transaction Type*

Valid Value(s):

Constant GENSCH (left-justified and space-filled).

IGBT-SCH-SEGMENT-SEPARATOR

Description: *Segment Separator*. Define the value that is used to separate the segments in a composite data element.

Valid Value(s):

In the first field, a special character.

In the second field, the hexadecimal value.

**Note:** The default value is “=” for ANA TRADACOMS.

IGBT-SCH-COMPONENT-ELT-SEP

Description: *Component Element Separator*. Define the value that is used to separate the component data elements in a composite data element.

Valid Value(s):

In the first field, a special character.

In the second field, the hexadecimal value.

**Note:** The default value is “:” for ANA TRADACOMS.

IGBT-SCH-DATA-ELEMENT-SEP

Description: *Data Element Separator*. Define the value that is used to separate elements in an outbound data segment.

Valid Value(s):

In the first field, a special character.

In the second field, the hexadecimal value.

**Note:** The default value is “+” for ANA TRADACOMS.

## IGBT-SCH-SEGMENT-TERMINATOR

Description: *Segment Terminator*. Define the value used to identify the end of an outbound data segment.

Valid Value(s):

In the first field, a special character.

**Note:** In the second field, the hexadecimal value. The default value is “” for ANA TRADACOMS.

## IGBT-SCH-RELEASE-INDICATOR

Description: *Release Indicator*. Define a unique character that restores a syntactical character back to its original meaning when used in conjunction with that syntactical separator or terminator character in a document.

**Note:** The default value is “?” for ANA TRADACOMS.

## IGBT-MULTIPLE-ENVELOPE-ID

Description: *Multiple Envelope ID*. Determine which interchange envelope to generate for outbound transmissions.

Valid Value(s):

ISA	=	Generate ISA, IEA control envelopes
ICS	=	Generate ICS, ICE control envelopes
BG	=	Generate BG-EG control envelopes
GS	=	Do not generate transmission control envelopes; use GS, GE instead
SCH	=	Generate SCH, END
STX	=	Generate STX, END
UNA	=	Generate UNA, UNB, UNZ control envelopes
UNB	=	Generate UNB, UNZ control envelopes

## IGBT-PARTNER-INT-VERSION

Description: *Version*. Use the partner interchange version when more than one control record with the same interchange header option is needed. Setting up partners with the partner interchange version should only be used when multiple like interchange envelopes need to be generated with different information.

**GENSTX1 Record***Description*

The GENSTX1, GENSTX2, and GENSTX3 records control outbound envelope generation.

*Record Layout*

<b>Name</b>	<b>Data Type</b>	<b>Position</b>
01 IS1T-INPUT-PART-GEN-STX1-TRAN.		
05 IS1T-ACTION-CODE	X(01)	001 - 001
05 IS1T-PARTNER-KEY.		
10 IS1T-PARTNER-ID	X(35)	002 - 036
10 IS1T-PARTNER-ID-QUAL	X(04)	037 - 040
10 IS1T-RECORD-TYPE	X(16)	041 - 056
05 IS1T-USRPAR-KEY		
REDEFINES IS1T-PARTNER-KEY.		
10 IS1T-USRPAR-USER-ID	X(15)	002 - 016
10 IS1T-USRPAR-PARTNER-ID	X(15)	017 - 031
10 FILLER	X(09)	032 - 040
10 IS1T-USRPAR-RECORD-TYPE	X(16)	041 - 056
05 IS1T-STX-SYNTAX-IDENTIFIER	X(04)	057 - 060
05 IS1T-STX-SYNTAX-VERSION	X(01)	061 - 061
05 IS1T-STX-CONTROL-REFERENCE	X(14)	062 - 075
05 IS1T-STX-RECIPIENT-REFERENCE	X(14)	076 - 089
05 IS1T-STX-APPLICATION-REFERENCE	X(14)	090 - 103
05 IS1T-STX-PROCESSING-PRTY-CODE	X(01)	104 - 104
05 IS1T-STX-NETWORK-ID	X(10)	105 - 114
05 IS1T-MULTIPLE-ENVELOPE-ID	X(03)	115 - 117
05 IS1T-PARTNER-INT-VERSION	X(12)	118 - 129

**GENSTX1 FMS Transaction Definitions**

IS1T-ACTION-CODE (Required)

Description: *Transaction Action Code*

Valid Value(s):

A	=	Add
C	=	Change
D	=	Delete

**Partner/Qualifier Mode**

IS1T-PARTNER-ID (Required)

Description: *Partner ID*

IS1T-PARTNER-ID-QUAL (Required)

Description: *Partner ID Qualifier*

IS1T-RECORD-TYPE (Required)

Description: *Transaction Type*

Valid Value(s):

Constant GENSTX1 (left-justified and space-filled).

**Relationship Mode**

IS1T-USRPAR-USER-ID (Required)

Description: *User ID*

IS1T-USRPAR-PARTNER-ID (Required)

Description: *Partner ID*

IS1T-USRPAR-RECORD-TYPE (Required)

Description: *Transaction Type*

Valid Value(s):

Constant GENSTX1 (left-justified and space-filled).

IS1T-STX-SYNTAX-IDENTIFIER

Description: *Syntax Identifier*. Enter the syntax identifier to use when generating the STX (STX01:01).

IS1T-STX-SYNTAX-VERSION

Description: *Syntax Version*. Enter the syntax identifier to use when generating the STX (STX01:02).

IS1T-STX-CONTROL-REFERENCE

Description: *Control Reference*. Enter the control reference to use when generating the STX (STX05).

IS1T-STX-RECIPIENT-REFERENCE

Description: *Recipient Reference*. Enter the recipient reference information to use when generating the STX (STX06:01).

IS1T-STX-APPLICATION-REFERENCE

Description: *Application Reference*. Enter the application reference to use when generating the STX (STX07).

IS1T-STX-PROCESSING-PRTY-CODE

Description: *Processing Priority Code*. Enter the processing priority code for generating the STX (STX08).

IS1T-STX-NETWORK-ID

Description: *Network ID*. Enter the TUN number for TRADANET.

## IS1T-MULTIPLE-ENVELOPE-ID

Description: *Multiple Envelope ID*. Determine which interchange envelope to generate for outbound transmissions.

Valid Value(s):

ISA	=	Generate ISA, IEA control envelopes
ICS	=	Generate ICS, ICE control envelopes
BG	=	Generate BG-EG control envelopes
GS	=	Do not generate transmission control envelopes; use GS, GE instead
SCH	=	Generate SCH, END
STX	=	Generate STX, END
UNA	=	Generate UNA, UNB, UNZ control envelopes
UNB	=	Generate UNB, UNZ control envelopes

## IS1T-PARTNER-INT-VERSION

Description: *Version*. Use the partner interchange version when more than one control record with the same interchange header option is needed. Setting up partners with the partner interchange version should only be used when multiple like interchange envelopes need to be generated with different information.



**GENSTX2 Record***Description*

The GENSTX1, GENSTX2, and GENSTX3 records control outbound envelope generation.

*Record Layout*

Name	Data Type	Position
01 IS2T-INPUT-PART-GEN-STX2-TRAN.		
05 IS2T-ACTION-CODE	X(01)	001 - 001
05 IS2T-PARTNER-KEY.		
10 IS2T-PARTNER-ID	X(35)	002 - 036
10 IS2T-PARTNER-ID-QUAL	X(04)	037 - 040
10 IS2T-RECORD-TYPE	X(16)	041 - 056
05 IS2T-USRPAR-KEY		
REDEFINES IS2T-PARTNER-KEY.		
10 IS2T-USRPAR-USER-ID	X(15)	002 - 016
10 IS2T-USRPAR-PARTNER-ID	X(15)	017 - 031
10 FILLER	X(09)	032 - 040
10 IS2T-USRPAR-RECORD-TYPE	X(16)	041 - 056
05 IS2T-STX-SENDER-NAME	X(35)	057 - 091
05 IS2T-STX-SENDER-ANA-CODE	X(14)	092 - 105
05 IS2T-MULTIPLE-ENVELOPE-ID	X(03)	106 - 108
05 IS2T-PARTNER-INT-VERSION	X(12)	109 - 120

**GENSTX2 FMS Transaction Definitions****IS2T-ACTION-CODE (Required)**

Description: *Transaction Action Code*

Valid Value(s):

A	=	Add
C	=	Change
D	=	Delete

**Partner/Qualifier Mode****IS2T-PARTNER-ID (Required)**

Description: *Partner ID*

**IS2T-PARTNER-ID-QUAL (Required)**

Description: *Partner ID Qualifier*

**IS2T-RECORD-TYPE (Required)**

Description: *Transaction Type*

Valid Value(s):

Constant GENSTX2 (left-justified and space-filled).

**Relationship Mode****IS2T-USRPAR-USER-ID (Required)**

Description: *User ID*

## IS2T-USRPAR-PARTNER-ID (Required)

Description: *Partner ID*

## IS2T-USRPAR-RECORD-TYPE (Required)

Description: *Transaction Type*

Valid Value(s):

Constant GENSTX2 (left-justified and space-filled).

## IS2T-STX-SENDER-NAME

Description: *Sender Name*. Enter the name of the sender to use when generating the STX segment (STX02:01).

**Note:** This field supports the use of underscore substitution characters (USCs). To use a USC, the substitution character must exist on the Partner Header record.

See the *Gentran:Basic for zSeries Release 6.4 User's Guide* for more information about this feature.

## IS2T-STX-SENDER-ANA-CODE

Description: *Sender ID Code*. Enter the Sender ID interchange identifier code used to generate the STX segment (STX02:02).

**Note:** This field supports the use of underscore substitution characters (USCs). To use a USC, the substitution character must exist on the Partner Header record.

See the *Gentran:Basic for zSeries Release 6.4 User's Guide* for more information about this feature.

## IS2T-MULTIPLE-ENVELOPE-ID

Description: *Multiple Envelope ID*. Determine which interchange envelope to generate for outbound transmissions.

Valid Value(s):

ISA	=	Generate ISA, IEA control envelopes
ICS	=	Generate ICS, ICE control envelopes
BG	=	Generate BG-EG control envelopes
GS	=	Do not generate transmission control envelopes; use GS, GE instead
SCH	=	Generate SCH, END
STX	=	Generate STX, END
UNA	=	Generate UNA, UNB, UNZ control envelopes
UNB	=	Generate UNB, UNZ control envelopes

## IS2T-PARTNER-INT-VERSION

Description: *Version*. Use the partner interchange version when more than one control record with the same interchange header option is needed. Setting up partners with the partner interchange version should only be used when multiple like interchange envelopes need to be generated with different information.

**GENSTX3 Record***Description*

The GENSTX1, GENSTX2, and GENSTX3 records control outbound envelope generation.

*Record Layout*

Name	Data Type	Position
01 IS3T-INPUT-PART-GEN-STX3-TRAN.		
05 IS3T-ACTION-CODE	X(01)	001 - 001
05 IS3T-PARTNER-KEY.		
10 IS3T-PARTNER-ID	X(35)	002 - 036
10 IS3T-PARTNER-ID-QUAL	X(04)	037 - 040
10 IS3T-RECORD-TYPE	X(16)	041 - 056
05 IS3T-USRPAR-KEY		
REDEFINES IS3T-PARTNER-KEY.		
10 IS3T-USRPAR-USER-ID	X(15)	002 - 016
10 IS3T-USRPAR-PARTNER-ID	X(15)	017 - 031
10 FILLER	X(09)	032 - 040
10 IS3T-USRPAR-RECORD-TYPE	X(16)	041 - 056
05 IS3T-STX-RECEIVER-NAME	X(35)	057 - 091
05 IS3T-STX-RECEIVER-ANA-CODE	X(14)	092 - 105
05 IS3T-MULTIPLE-ENVELOPE-ID	X(03)	106 - 108
05 IS3T-PARTNER-INT-VERSION	X(12)	109 - 120

**GENSTX3 FMS Transaction Definitions****IS3T-ACTION-CODE (Required)**

Description: *Transaction Action Code*

Valid Value(s):

A	=	Add
C	=	Change
D	=	Delete

**Partner/Qualifier Mode****IS3T-PARTNER-ID (Required)**

Description: *Partner ID*

**IS3T-PARTNER-ID-QUAL (Required)**

Description: *Partner ID Qualifier*

**IS3T-RECORD-TYPE (Required)**

Description: *Transaction Type*

Valid Value(s):

Constant GENSTX3 (left-justified and space-filled).

**Relationship Mode****IS3T-USRPAR-USER-ID (Required)**

Description: *User ID*

## IS3T-USRPAR-PARTNER-ID (Required)

Description: *Partner ID*

## IS3T-USRPAR-RECORD-TYPE (Required)

Description: *Transaction Type*

Valid Value(s):

Constant GENSTX1 (left-justified and space-filled).

## IS3T-STX-RECEIVER-NAME

Description: *Receiver Name*. Enter the name of the receiver to use when generating the STX segment (STX03:01).

**Note:** This field supports the use of underscore substitution characters (USCs). To use a USC, the substitution character must exist on the Partner Header record.

See the *Gentran:Basic for zSeries Release 6.4 User's Guide* for more information about this feature.

## IS3T-STX-RECEIVER-ANA-CODE

Description: *Receiver ID Code*. Enter the Receiver ID interchange identifier code used to generate the STX segment (STX03:02).

**Note:** This field supports the use of underscore substitution characters (USCs). To use a USC, the substitution character must exist on the Partner Header record.

See the *Gentran:Basic for zSeries Release 6.4 User's Guide* for more information about this feature.

## IS3T-MULTIPLE-ENVELOPE-ID

Description: *Multiple Envelope ID*. Determine which interchange envelope to generate for outbound transmissions.

Valid Value(s):

ISA	=	Generate ISA, IEA control envelopes
ICS	=	Generate ICS, ICE control envelopes
BG	=	Generate BG-EG control envelopes
GS	=	Do not generate transmission control envelopes; use GS, GE instead
SCH	=	Generate SCH, END
STX	=	Generate STX, END
UNA	=	Generate UNA, UNB, UNZ control envelopes
UNB	=	Generate UNB, UNZ control envelopes

## IS3T-PARTNER-INT-VERSION

Description: *Version*. Use the partner interchange version when more than one control record with the same interchange header option is needed. Setting up partners with the partner interchange version should only be used when multiple like interchange envelopes need to be generated with different information.

## Group Records

### GRPMISC Record

#### Description

The GRPMISC transaction record updates some of the fields on the partner profile Group record, including:

- Which group and transaction types to generate for enveloping
- Whether to expect the partner to acknowledge this group
- Which kind of acknowledgment to send when this group type is received

#### Record Layout

Name	Data Type	Position
01 IGMT-INPUT-GROUP-MISC-TRAN.		
05 IGMT-ACTION-CODE	X (01)	001 - 001
05 IGMT-PARTNER-KEY.		
10 IGMT-PARTNER-ID	X (35)	002 - 036
10 IGMT-PARTNER-ID-QUAL	X (04)	037 - 040
10 IGMT-RECORD-TYPE.		
15 IGMT-RECORD-TYPE-VAL-GR	X (07)	041 - 047
15 IGMT-RECORD-TYPE-GROUP-ID.		
20 IGMT-GROUP-ID-CHAR-1	X (01)	048 - 048
20 IGMT-REST-OF-ID	X (05)	049 - 053
10 IGMT-TRANSACTION-TYPE-REST	X (03)	054 - 056
05 IGMT-USRPAR-KEY		
REDEFINES IGMT-PARTNER-KEY.		
10 IGMT-USRPAR-USER-ID	X (15)	002 - 016
10 IGMT-USRPAR-PARTNER-ID	X (15)	017 - 031
10 FILLER	X (09)	032 - 040
10 IGMT-USRPAR-RECORD-TYPE	X (16)	041 - 056
05 IGMT-GROUP-CNTL-SEGMENT-SELECT	X (03)	057 - 059
05 IGMT-TRANS-CNTL-SEGMENT-SELECT	X (03)	060 - 062
05 IGMT-EXPECT-GROUP-ACK	X (01)	063 - 063
05 IGMT-GROUP-ACK-TRANS	X (06)	064 - 069
05 IGMT-UNG-ACK-TRANS	X (06)	070 - 075
05 IGMT-UNH-CONTROL-NUMBER	X (14)	076 - 089
05 IGMT-BAT-CONTROL-REFERENCE	X (12)	090 - 101
05 IGMT-MHD-CONTROL-NUMBER	X (12)	102 - 113
05 IGMT-ACKNOWLEDGE-GRP-OR-TRNS	X (01)	114 - 114
05 IGMT-ACKNOWLEDGE-OVERDUE.		
10 IGMT-ACKNOWLEDGE-OVERDUE-MM	X (02)	115 - 116
10 IGMT-ACKNOWLEDGE-OVERDUE-HHH	X (05)	117 - 119
05 IGMT-ALT-ACK-PARTNER-ID	X (35)	120 - 154
05 IGMT-ALT-ACK-PARTNER-ID-QUAL	X (04)	155 - 158
05 IGMT-VERSION	X (12)	159 - 170
05 IGMT-VIEWPOINT	X (01)	171 - 171
05 IGMT-VIEWPOINT-TM	X (01)	172 - 172
05 IGMT-MULTIPLE-ENVELOPE-ID	X (03)	173 - 175

**GRPMISC FMS Transaction Definitions**

IGMT-ACTION-CODE (Required)

Description: *Transaction Action Code*

Valid Value(s):

A	=	Add
C	=	Change
D	=	Delete

**Partner/Qualifier Mode**

IGMT-PARTNER-ID (Required)

Description: *Partner ID*

IGMT-PARTNER-ID-QUAL (Required)

Description: *Partner ID Qualifier*

IGMT-RECORD-TYPE-VAL-GR (Required)

Description: *Transaction Type*

Valid Value(s):

Constant GRPMISC (left-justified and space-filled).

IGMT-RECORD-TYPE-GROUP-ID (Required)

Description: *Functional Group Code* (left-justified and space-filled).

IGMT-TRANSACTION-TYPE-REST

Description: *Transaction Type Rest*. This field is blank.

Valid Value(s):

Spaces

**Relationship Mode**

IGMT-USRPAR-USER-ID (Required)

Description: *User ID*

IGMT-USRPAR-PARTNER-ID (Required)

Description: *Partner ID*

IGMT-USRPAR-RECORD-TYPE (Required)

Description: The following fields are used to build this field:

- IGMT-RECORD-TYPE-VAL-GR
- IGMT-RECORD-TYPE-GROUP-ID
- IGMT-TRANSACTION-TYPE-REST

See the definitions for the previous Partner/Qualifier mode section for more information about these fields.

IGMT-GROUP-CNTL-SEGMENT-SELECT

Description: *Group Control Segment Select*. Select which functional group header to generate for this group.

Valid Value(s):

GS, UNG, and BAT

## IGMT-TRANS-CNTL-SEGMENT-SELECT

Description: *Transaction Control Segment Select*. Select which transaction control header to generate for transactions in this group that are not specifically designated elsewhere.

Valid Value(s):

ST, UNH, and MHD

## IGMT-EXPECT-GROUP-ACK

Description: *Expect Group Acknowledgment*. Indicate whether the partner returns acknowledgments for this group.

Valid Value(s):

Y = Expect acknowledgments from the partner  
N = Do not expect acknowledgments from the partner  
Blank = Do not expect acknowledgments from the partner

## IGMT-GROUP-ACK-TRANS

Description: *Group Acknowledgment Transaction*. Enter the code (left-justified) of transaction to use to acknowledge this functional group.

Valid Value(s):

997  
999  
Blank

Default Value(s):

999  
CONTRL

## IGMT-UNG-ACK-TRANS

Description: *Acknowledge Transaction* (left-justified). Enter the code of the transaction used to acknowledge this functional group.

Valid Value(s):

997  
999  
Blank

Default Value(s):

997  
CONTRL

## IGMT-UNH-CONTROL-NUMBER

Description: *Control Number*. Enter the control number to use when generating UNH segments for transactions in this functional group that are not specified elsewhere (UNH01).

## IGMT-BAT-CONTROL-REFERENCE

Description: *Control Reference*. Control number for the BAT sequence within the transaction or interchange.

## IGMT-MHD-CONTROL-NUMBER

Description: *Control Number*. The default value if the transaction does not exist.

## IGMT-ACKNOWLEDGE-GRP-OR-TRNS

Description: *Acknowledge Group or Transaction*. Identify the group or transaction acknowledgments should be generated.

Valid Value(s):

G	=	Generate group acknowledgments
T	=	Generate transaction acknowledgments

## IGMT-ACKNOWLEDGE-OVERDUE-MM

Description: *Acknowledge Overdue Minutes*. This field is used in reconciliation with received acknowledgments and outbound groups. Enter a value greater than zero (0) in this field if you want to monitor overdue acknowledgment in minutes.

Valid Value(s):

Numeric only (00 through 59)

## IGMT-ACKNOWLEDGE-OVERDUE-HHH

Description: *Acknowledge Overdue Hours*. This field is used in reconciliation with received acknowledgments and outbound groups. Enter a value greater than zero (0) in this field if you want to monitor overdue acknowledgment in hours.

Valid Value(s):

Numeric only (000 through 999)

## IGMT-ALT-ACK-PARTNER-ID

Description: *Alternate Acknowledge Partner ID*. This field is used in reconciliation of group directories with databanking. Add an alternate partner for multi-division reconciliations against outbound EDI databank.

## IGMT-ALT-ACK PARTNER-ID-QUAL

Description: *Alternate Acknowledge ID Partner Qualifier*. Add an alternate Partner ID and Qualifier for IGMT-ALT-ACK-PARTNER-ID.

## IGMT-VERSION

Description: *Version*. Associate a version/industry/association ID with this group or group by version (left-justified and space-filled).

## IGMT-VIEWPOINT

Description: *Viewpoint*. Indicate whether Gentran:Viewpoint Exception Management activity should be performed at the group level.

Valid Value(s):

Y	=	Gentran:Viewpoint Exception Management is performed
N	=	Gentran:Viewpoint Exception Management is not performed.
Blank	=	This field must be blank if Viewpoint is not installed.

## IGMT-VIEWPOINT-TM

Description: *Viewpoint Tracking Management*. Indicate whether Gentran:Viewpoint tracking management activity should be performed at the group level.

Valid Value(s):

Y	=	Gentran:Viewpoint Tracking Management is performed.
N	=	Gentran:Viewpoint Tracking Management is not performed.
Blank	=	This field must be blank if Viewpoint is not installed.



## IMGT-MULTIPLE-ENVELOPE-ID

Description: *Multiple Envelope ID*. Determine which interchange envelope to generate for outbound transmissions.

## Valid Value(s):

ISA	=	Generate ISA, IEA control envelopes
ICS	=	Generate ICS, ICE control envelopes
BG	=	Generate BG-EG control envelopes
GS	=	Do not generate transmission control envelopes; use GS, GE instead
SCH	=	Generate SCH, END
STX	=	Generate STX, END
UNA	=	Generate UNA, UNB, UNZ control envelopes
UNB	=	Generate UNB, UNZ control envelopes

**GRPIN Record***Description*

The GRPIN transaction continues the same function as GRPMISC loading fields in the partner profile Group record.

*Record Layout*

Name	Data Type	Position
01	IGIT-INPUT-GROUP-INCOMING-TRAN.	
05	IGIT-ACTION-CODE	X(01) 001 - 001
05	IGIT-PARTNER-KEY.	
10	IGIT-PARTNER-ID	X(35) 002 - 036
10	IGIT-PARTNER-ID-QUAL	X(04) 037 - 040
10	IGIT-RECORD-TYPE.	
15	IGIT-RECORD-TYPE-VAL-GR	X(05) 041 - 045
15	IGIT-RECORD-TYPE-GROUP-ID.	
20	IGIT-GROUP-ID-CHAR-1	X(01) 046 - 046
20	IGIT-REST-OF-ID	X(05) 047 - 051
10	IGIT-TRANSACTION-TYPE-REST	X(05) 052 - 056
05	IGIT-USRPAR-KEY	
	REDEFINES IGIT-PARTNER-KEY.	
10	IGIT-USRPAR-USER-ID	X(15) 002 - 016
10	IGIT-USRPAR-PARTNER-ID	X(15) 017 - 031
10	FILLER	X(09) 032 - 040
10	IGIT-USRPAR-RECORD-TYPE	X(16) 041 - 056
05	IGIT-US-PARAMETERS.	
10	IGIT-GROUP-ACCEPT-FLAG	X(01) 057 - 057
10	IGIT-GROUP-VERSION-USE	X(01) 058 - 058
10	IGIT-INBD-GROUP-VERSION	X(12) 059 - 070
10	IGIT-LAST-INBD-GS-CNTL-NO	X(09) 071 - 079
05	IGIT-INTERNATIONAL-PARAMETERS.	
10	IGIT-LAST-INBD-INTL-CNTL-NO	X(14) 080 - 093
10	IGIT-INBD-INTL-PASSWORD	X(14) 094 - 107
10	IGIT-INBD-INTL-VERSION-USE	X(01) 108 - 108
10	IGIT-INBD-INTL-VERSION	X(12) 109 - 120
05	IGIT-GROUP-INBD-DB-LEVEL	X(01) 121 - 121
05	IGIT-APPL-INBD-DB-LEVEL	X(01) 122 - 122
05	IGIT-VERSION	X(12) 123 - 134
05	IGIT-MULTIPLE-ENVELOPE-ID	X(03) 135 - 137

*GRPIN FMS Transaction Definitions*

IGIT-ACTION-CODE (Required)

Description: *Transaction Action Code*

Valid Value(s):

A	=	Add
C	=	Change
D	=	Delete

**Partner/Qualifier Mode**

IGIT-PARTNER-ID (Required)

Description: *Partner ID*

IGIT-PARTNER-ID-QUAL (Required)

Description: *Partner ID Qualifier*

IGIT-RECORD-TYPE-VAL-GR (Required)

Description: *Transaction Type*

Valid Value(s):

Constant GRPIN (left-justified and space-filled).

IGIT-RECORD-TYPE-GROUP-ID

Description: *Functional Group Code* (left-justified and space-filled)

IGIT-TRANSACTION-TYPE-REST

Description: *Transaction Type Rest*. This field is blank.

Valid Value(s):

Spaces

**Relationship Mode**

IGIT-USRPAR-USER-ID (Required)

Description: *User ID*

IGIT-USRPAR-PARTNER-ID (Required)

Description: *Partner ID*

IGIT-USRPAR-RECORD-TYPE (Required)

Description: The following fields are used to build this field:

- IGIT-RECORD-TYPE-VAL-GR
- IGIT-RECORD-TYPE-GROUP-ID
- IGIT-TRANSACTION-TYPE-REST

See the definitions for the previous Partner/Qualifier mode section for more information about these fields.

IGIT-GROUP-ACCEPT-FLAG

Description: *Group Accept Flag*. Indicate whether to accept this functional group from this partner.

Valid Value(s):

Y	=	Accept this group from this partner
Space	=	Accept this group from this partner
N	=	Do not accept this group from this partner

IGIT-GROUP-VERSION-USE

Description: *Group Version Use*. Define the version use for this group.

Valid Value(s):

A	=	Always use this version from this partner
I	=	Use the version from the data
D	=	Use this version if cannot find version on input data

## IGIT-INBD-GROUP-VERSION

Description: *Inbound Group Version*. Indicate the Version ID of the standard to use in compliance checking for the group record.

## IGIT-LAST-INBD-GS-CNTL-NO

Description: *Last Inbound GS Control Number*. Set up how Gentran checks the last control number received on GS interchange.

Valid Value(s): Zeros = Interchange control number checking starts at 1  
 Non-zero numeral = Interchange control number checking begins at n+1  
 Blank = Checking function is not used.

## IGIT-LAST-INBD-INTL-CNTL-NO

Reserved for future use.

## IGIT-INBD-INTL-PASSWORD

Reserved for future use.

## IGIT-INBD-INTL-VERSION-USE

Reserved for future use.

## IGIT-INBD-INTL-VERSION

Reserved for future use.

## IGIT-GROUP-INBD-DB-LEVEL

Description: *Group Inbound Databank Level*. Set the inbound EDI databanking level for partner.

Valid Value(s): D = Directory level  
 N = None

## IGIT-APPL-INBD-DB-LEVEL

Description: *Application Inbound Databank Level*. Not currently used.

## IGIT-VERSION

Description: *Version*. Associate the version/industry/association ID with this group or group by version (left-justified and space-filled).

## IGIT-MULTIPLE-ENVELOPE-ID

Description: *Multiple Envelope ID*. Determine which interchange envelope to generate for outbound transmissions.

Valid Value(s): ISA = Generate ISA, IEA control envelopes  
 ICS = Generate ICS, ICE control envelopes  
 BG = Generate BG-EG control envelopes  
 GS = Do not generate transmission control envelopes; use GS, GE  
 SCH = Generate SCH, END  
 STX = Generate STX, END  
 UNA = Generate UNA, UNB, UNZ control envelopes  
 UNB = Generate UNB, UNZ control envelopes

**GRPOUT Record***Description*

The GRPOUT transaction determines:

- Whether to send this group to this partner
- Which version to use for this partner and group when generating envelopes

*Record Layout*

Name	Data Type	Position
01 IGOT-INPUT-GROUP-OUTGOING-TRAN.		
05 IGOT-ACTION-CODE	X(01)	001 - 001
05 IGOT-PARTNER-KEY.		
10 IGOT-PARTNER-ID	X(35)	002 - 036
10 IGOT-PARTNER-ID-QUAL	X(04)	037 - 040
10 IGOT-RECORD-TYPE.		
15 IGOT-RECORD-TYPE-VAL-GR	X(06)	041 - 046
15 IGOT-RECORD-TYPE-GROUP-ID.		
20 IGOT-GROUP-ID-CHAR-1	X(01)	047 - 047
20 IGOT-REST-OF-ID	X(05)	048 - 052
10 IGOT-TRANSACTION-TYPE-REST	X(04)	053 - 056
05 IGOT-USRPAR-KEY		
REDEFINES IGOT-PARTNER-KEY.		
10 IGOT-USRPAR-USER-ID	X(15)	002 - 016
10 IGOT-USRPAR-PARTNER-ID	X(15)	017 - 031
10 FILLER	X(09)	032 - 040
10 IGOT-USRPAR-RECORD-TYPE	X(16)	041 - 056
05 IGOT-US-PARAMETERS.		
10 IGOT-GROUP-SEND-FLAG	X(01)	057 - 057
10 IGOT-OUTBD-GROUP-VERSN-USE	X(01)	058 - 058
10 IGOT-OUTBD-GROUP-VERSION	X(12)	059 - 070
10 IGOT-LAST-OUTBD-GS-CNTL-NO	X(09)	071 - 079
05 IGOT-INTERNATIONAL-PARAMETERS.		
10 FILLER	X(14)	080 - 093
10 IGOT-LAST-OUTBD-INTL-CNTL-NO	X(14)	094 - 107
10 IGOT-OUTBD-INTL-VERSION-USE	X(01)	108 - 108
10 IGOT-OUTBD-INTL-VERSION	X(12)	109 - 120
05 IGOT-GROUP-OUTBD-DB-LEVEL	X(01)	121 - 121
05 IGOT-APPL-OUTBD-DB-LEVEL	X(01)	122 - 122
05 IGOT-VERSION	X(12)	123 - 134
05 IGOT-MULTIPLE-ENVELOPE-ID	X(03)	135 - 137

---

**GRPOUT FMS Transaction Definitions**

IGOT-ACTION-CODE (Required)

Description: *Transaction Action Code*

Valid Value(s):

A	=	Add
C	=	Change
D	=	Delete

**Partner/Qualifier Mode**

IGOT-PARTNER-ID (Required)

Description: *Partner ID*

IGOT-PARTNER-ID-QUAL (Required)

Description: *Partner ID Qualifier*

IGOT-RECORD-TYPE-VAL-GR (Required)

Description: *Transaction Type*

Valid Value(s):

Constant GRPOUT (left-justified and space-filled).

IGOT-RECORD-TYPE-GROUP-ID

Description: *Functional Group Code* (left-justified and space-filled).

IGOT-TRANSACTION-TYPE-REST

Description: *Transaction Type Rest*. This field is blank.

Valid Value(s):

Spaces

**Relationship Mode**

IGOT-USRPAR-USER-ID (Required)

Description: *User ID*

IGOT-USRPAR-PARTNER-ID (Required)

Description: *Partner ID*

IGOT-USRPAR-RECORD-TYPE (Required)

Description: The following fields are used to build this field:

- IGOT-RECORD-TYPE-VAL-GR
- IGOT-RECORD-TYPE-GROUP-ID
- IGOT-TRANSACTION-TYPE-REST

See the definitions for the previous Partner/Qualifier mode section for more information about these fields.

## IGOT-GROUP-SEND-FLAG

Description: *Group Send Flag*. Indicate whether to send this functional group from this partner.

Valid Value(s):

Y	=	Send this functional group from this partner
Spaces	=	Send this functional group from this partner
N	=	Do <i>not</i> send this functional group from this partner

Default Value(s):

Spaces

## IGOT-OUTBD-GROUP-VERSN-USE

Description: *Outbound Group Version Use*. Define the version use for this group.

Valid Value(s):

A	=	Always use this version for this partner
I	=	Use the version from the data
Space	=	Use the version from the data
D	=	Use this version if cannot find version on input data

## IGOT-OUTBD-GROUP-VERSION

Reserved for future use.

## IGOT-LAST-OUTBD-GS-CNTL-NO

Description: *Last Outbound GS Control Number*. Define the last outbound GS control number used for this group.

Valid Value(s):

Zeros	=	Interchange control number checking starts at 1
Non-zero numeral	=	Interchange control number checking begins at n+1
Blank	=	Checking function is not used.

## IGOT-LAST-OUTBD-INTL-CNTL-NO

Reserved for future use.

## IGOT-OUTBD-INTL-VERSION-USE

Reserved for future use.

## IGOT-OUTBD-INTL-VERSION

Reserved for future use.

## IGOT-GROUP-OUTBD-DB-LEVEL

Description: *Group Outbound Databank Level*. Set the outbound EDI databanking level for the partner.

Valid Value(s):

D	=	Directory level
N	=	None

## IGOT-APPL-OUTBD-DB-LEVEL

Description: *Application Outbound Databank Level*. Not currently used.

## IGOT-VERSION

Description: *Version*. Associate the version/industry/association ID with this group or group by version (left-justified and space-filled).

## IGOT-MULTIPLE-ENVELOPE-ID

Description: *Multiple Envelope ID*. Determine which interchange envelope to generate for outbound transmissions.

## Valid Value(s):

ISA	=	Generate ISA, IEA control envelopes
ICS	=	Generate ICS, ICE control envelopes
BG	=	Generate BG-EG control envelopes
GS	=	Do not generate transmission control envelopes; use GS, GE instead
SCH	=	Generate SCH, END
STX	=	Generate STX, END
UNA	=	Generate UNA, UNB, UNZ control envelopes
UNB	=	Generate UNB, UNZ control envelopes



**GENGS Record***Definition*

The GENGs record controls outbound envelope generation.

The record is part of the Group record. If the Group Control Segment Select is equal to spaces, and this transaction is submitted, GS is forced into Group Control Segment Select. If the Group Control Segment Select is equal to UNG, error 541 occurs.

*Record Layout*

Name	Data Type	Position
01 IGST-INPUT-GROUP-GEN-GS-TRAN.		
05 IGST-ACTION-CODE	X(01)	001 - 001
05 IGST-PARTNER-KEY.		
10 IGST-PARTNER-ID	X(35)	002 - 036
10 IGST-PARTNER-ID-QUAL	X(04)	037 - 040
10 IGST-RECORD-TYPE.		
15 IGST-RECORD-TYPE-VAL-GR	X(05)	041 - 045
15 IGST-RECORD-TYPE-GROUP-ID.		
20 IGST-GROUP-ID-CHAR-1	X(01)	046 - 046
20 IGST-REST-OF-ID	X(05)	047 - 051
10 IGST-TRANSACTION-TYPE-REST	X(05)	052 - 056
05 IGST-USRPAR-KEY		
REDEFINES IGST-PARTNER-KEY.		
10 IGST-USRPAR-USER-ID	X(15)	002 - 016
10 IGST-USRPAR-PARTNER-ID	X(15)	017 - 031
10 FILLER	X(09)	032 - 040
10 IGST-USRPAR-RECORD-TYPE	X(16)	041 - 056
05 IGST-GROUP-SENDER-ID	X(15)	057 - 071
05 IGST-GROUP-RECEIVER-ID	X(15)	072 - 086
05 IGST-GROUP-CONTROL-NUMBER	X(09)	087 - 095
05 IGST-GS-RESPONSIBLE-AGENCY	X(03)	096 - 098
05 IGST-GROUP-VERSION-ID	X(12)	099 - 110
05 IGST-TRANSACTION-CNTL-NUMBER	X(09)	111 - 119
05 IGST-VERSION	X(12)	120 - 131
05 IGST-MULTIPLE-ENVELOPE-ID	X(03)	132 - 134

*GENGS FMS Transaction Definitions***IGST-ACTION-CODE (Required)**

Description: *Transaction Action Code*

Valid Value(s):

A	=	Add
C	=	Change
D	=	Delete

**Partner/Qualifier Mode****IGST-PARTNER-ID (Required)**

Description: *Partner ID*

IGST-PARTNER-ID-QUAL (Required)

Description: *Partner ID Qualifier*

IGST-RECORD-TYPE-VAL-GR (Required)

Description: *Transaction Type*

Valid Value(s):

Constant GENGS

IGST-RECORD-TYPE-GROUP-ID

Description: *Functional Group Code* (left-justified and space-filled).

IGST-TRANSACTION-TYPE-REST

Description: *Transaction Type Rest*. This field is blank.

Valid Value(s):

Spaces

### **Relationship Mode**

IGST-USRPAR-USER-ID (Required)

Description: *User ID*

IGST-USRPAR-PARTNER-ID (Required)

Description: *Partner ID*

IGST-USRPAR-RECORD-TYPE (Required)

Description: The following fields are used to build this field:

- IGST-RECORD-TYPE-VAL-GR
- IGST-RECORD-TYPE-GROUP-ID
- IGST-TRANSACTION-TYPE-REST

See the definitions for the previous Partner/Qualifier mode section for more information about these fields.

IGST-GROUP-SENDER-ID

Description: *Group Sender ID*. Enter the group Sender ID code for GS (GS02).

**Note:** This field supports the use of underscore substitution characters (USCs). To use a USC, the substitution character must exist on the Partner Header record.

See the *Gentran:Basic for zSeries Release 6.4 User's Guide* for more information about this feature.

IGST-GROUP-RECEIVER-ID

Description: *Group Receiver ID*. Enter the group Receiver ID code for GS (GS03).

**Note:** This field supports the use of underscore substitution characters (USCs). To use a USC, the substitution character must exist on the Partner Header record.

See the *Gentran:Basic for zSeries Release 6.4 User's Guide* for more information about this feature.

## IGST-GROUP-CONTROL-NUMBER

Description: *Group Control Number*. Enter the group control number for this functional group.

## IGST-GS-RESPONSIBLE-AGENCY

Description: *Responsible Agency*. Enter the responsible agency code for GS segment (GS07).

## IGST-GROUP-VERSION-ID

Description: *Group Version ID*. Enter the Version ID for GS segment (GS08).

## IGST-TRANSACTION-CNTL-NUMBER

Description: *Transaction Control Number*. Enter the transaction control number (ST02) for the transactions in this functional group that are not specified elsewhere.

## IGST-VERSION

Description: *Version*. Associate the version/industry/association ID with this group or group by version (left-justified and space-filled).

## IGST-MULTIPLE-ENVELOPE-ID

Description: *Multiple Envelope ID*. Determine which interchange envelope to generate for outbound transmissions.

## Valid Value(s):

ISA	=	Generate ISA, IEA control envelopes
ICS	=	Generate ICS, ICE control envelopes
BG	=	Generate BG-EG control envelopes
GS	=	Do not generate transmission control envelopes; use GS, GE instead
SCH	=	Generate SCH, END
STX	=	Generate STX, END
UNA	=	Generate UNA, UNB, UNZ control envelopes
UNB	=	Generate UNB, UNZ control envelopes

**GENUNG1 Record***Description*

The GENUNG1 and GENUNG2 records control how UNG segments are generated for this partner. These transactions update the group record of the partner.

If the Group Control Segment Select is equal to spaces, and the transaction is submitted, UNG is forced into Group Control Segment Select. If Group Control Segment Select is any other value, an error results.

*Record Layout*

Name	Data Type	Position
01 IG1T-INPUT-GROUP-GEN-UNG1-TRAN.		
05 IG1T-ACTION-CODE	X(01)	001 - 001
05 IG1T-PARTNER-KEY.		
10 IG1T-PARTNER-ID	X(35)	002 - 036
10 IG1T-PARTNER-ID-QUAL	X(04)	037 - 040
10 IG1T-RECORD-TYPE.		
15 IG1T-RECORD-TYPE-VAL-GR	X(07)	041 - 047
15 IG1T-RECORD-TYPE-GROUP-ID.		
20 IG1T-GROUP-ID-CHAR-1	X(01)	048 - 048
20 IG1T-REST-OF-ID	X(05)	049 - 053
10 IG1T-TRANSACTION-TYPE-REST	X(03)	054 - 056
05 IGIT-USRPAR-KEY		
REDEFINES IGIT-PARTNER-KEY.		
10 IG1T-USRPAR-USER-ID	X(15)	002 - 015
10 IG1T-USRPAR-PARTNER-ID	X(15)	017 - 015
10 FILLER	X(09)	032 - 041
10 IG1T-USRPAR-RECORD-TYPE	X(16)	041 - 056
05 IG1T-UNG-CONTROL-REFERENCE	X(14)	057 - 070
05 IG1T-UNG-CONTROLLING-AGENCY	X(03)	071 - 073
05 IG1T-UNG-MESSAGE-VERSION.		
10 IG1T-UNG-VERSION-NUMBER	X(03)	074 - 076
10 IG1T-UNG-VERSION-RELEASE	X(03)	077 - 079
10 IG1T-UNG-VERSION-ASSOCIATN	X(06)	080 - 085
05 IG1T-UNG-APPLICATION-PASSWORD	X(14)	086 - 099
05 IG1T-VERSION	X(12)	100 - 111
05 IG1T-MULIPLE-ENVELOPE-ID	X(03)	112 - 114

*GENUNG1 FMS Transaction Definitions***IG1T-ACTION-CODE (Required)**

Description: *Transaction Action Code*

Valid Value(s):

A	=	Add
C	=	Change
D	=	Delete

**Partner/Qualifier Mode**

IG1T-PARTNER-ID (Required)

Description: *Partner ID*

IG1T-PARTNER-ID-QUAL (Required)

Description: *Partner ID Qualifier*

IG1T-RECORD-TYPE-VAL-GR (Required)

Description: *Transaction Type*

Valid Value(s):

Constant GENUNG1

IG1T-RECORD-TYPE-GROUP-ID

Description: *Functional Group Code* (left-justified and space-filled).

IG1T-TRANSACTION-TYPE-REST

Description: *Transaction Type Rest*. This field is blank.

Valid Value(s):

Spaces

**Relationship Mode**

IG1T-USRPAR-USER-ID (Required)

Description: *User ID*

IG1T-USRPAR-PARTNER-ID (Required)

Description: *Partner ID*

IG1T-USRPAR-RECORD-TYPE (Required)

Description: The following fields are used to build this field:

- IG1T-RECORD-TYPE-VAL-GR
- IG1T-RECORD-TYPE-GROUP-ID
- IG1T-TRANSACTION-TYPE-REST

See the definitions for the previous Partner/Qualifier Mode section for more information about these fields.

IG1T-UNG-CONTROL-REFERENCE

Description: *Control Reference*. Enter the control reference to use when generating the UNG segment (UNG05).

IG1T-UNG-CONTROLLING-AGENCY

Description: *Controlling Agency*. Enter the controlling a agency to use when generating the UNG segment (UNG02:04).

IG1T-UNG-VERSION-NUMBER

Description: *Version Number*. Enter the version number to use when generating the UNG (UNG02:02).

## IG1T-UNG-VERSION-RELEASE

Description: *Version Release*. Enter the release of the version to use generate the UNG (UNG02:03).

## IG1T-UNG-VERSION-ASSOCIATN

Description: *Version Association*. Enter the version association assigned part of the version to use to generate the UNG (UNG02:05).

## IG1T-UNG-APPLICATION-PASSWORD

Description: *Application Password*. Enter the application password to use when generating the UNG (UNG08).

## IG1T-VERSION

Description: *Version*. Associate the version/industry/association ID with this group or group by version (left-justified and space-filled).

## IG1T-MULTIPLE-ENVELOPE-ID

Description: *Multiple Envelope ID*. Determine which interchange envelope to generate for outbound transmissions.

## Valid Value(s):

ISA	=	Generate ISA, IEA control envelopes
ICS	=	Generate ICS, ICE control envelopes
BG	=	Generate BG-EG control envelopes
GS	=	Do not generate transmission control envelopes; use GS, GE instead
SCH	=	Generate SCH, END
STX	=	Generate STX, END
UNA	=	Generate UNA, UNB, UNZ control envelopes
UNB	=	Generate UNB, UNZ control envelopes

**GENUNG2 Record***Description*

The GENUNG1 and GENUNG2 transactions update the group record of the partner profile. If the Group Control Segment Select is equal to spaces, and the transaction is submitted, UNG is forced into Group Control Segment Select. If Group Control Segment Select is any other value, an error results.

*Record Layout*

Name	Data Type	Position
01 IG2T-INPUT-GROUP-GEN-UNG2-TRAN.		
05 IG2T-ACTION-CODE	X(01)	001 - 001
05 IG2T-PARTNER-KEY.		
10 IG2T-PARTNER-ID	X(35)	002 - 036
10 IG2T-PARTNER-ID-QUAL	X(04)	037 - 040
10 IG2T-RECORD-TYPE.		
15 IG2T-RECORD-TYPE-VAL-GR	X(07)	041 - 047
15 IG2T-RECORD-TYPE-GROUP-ID.		
20 IG2T-GROUP-ID-CHAR-1	X(01)	048 - 048
20 IG2T-REST-OF-ID	X(05)	049 - 053
10 IG2T-TRANSACTION-TYPE-REST	X(03)	054 - 056
05 IG2T-USRPAR-KEY		
REDEFINES IG2T-PARTNER-KEY.		
10 IG2T-USRPAR-USER-ID	X(15)	002 - 016
10 IG2T-USRPAR-PARTNER-ID	X(15)	017 - 031
10 FILLER	X(09)	032 - 040
10 IG2T-USRPAR-RECORD-TYPE	X(16)	041 - 056
05 IG2T-UNG-SENDER-ID	X(35)	057 - 091
05 IG2T-UNG-SENDER-ID-QUAL	X(04)	092 - 095
05 IG2T-UNG-RECEIVER-ID	X(35)	096 - 130
05 IG2T-UNG-RECEIVER-ID-QUAL	X(04)	131 - 134
05 IG2T-VERSION	X(12)	135 - 146
05 IG2T-MULTIPLE-ENVELOPE-ID	X(03)	147 - 149

*GENUNG2 FMS Transaction Definitions***IG2T-ACTION-CODE (Required)**

Description: *Transaction Action Code*

Valid Value(s):

A	=	Add
C	=	Change
D	=	Delete

**Partner/Qualifier Mode****IG2T-PARTNER-ID (Required)**

Description: *Partner ID*

**IG2T-PARTNER-ID-QUAL (Required)**

Description: *Partner ID Qualifier*

IG2T-RECORD-TYPE-VAL-GR (Required)

Description: *Transaction Type*

Valid Value(s): Constant GENUNG2

IGST-RECORD-TYPE-GROUP-ID

Description: *Functional Group Code* (left-justified and space-filled).

IG2T-TRANSACTION-TYPE-REST

Description: *Transaction Type Rest*. This field is blank.

Valid Value(s):

Spaces

### **Relationship Mode**

IG2T-USRPAR-USER-ID (Required)

Description: *User ID*

IG2T-USRPAR-PARTNER-ID (Required)

Description: *Partner ID*

IG2T-USRPAR-RECORD-TYPE (Required)

Description: The following fields are used to build this field:

- IG2T-RECORD-TYPE-VAL-GR
- IG2T-RECORD-TYPE-GROUP-ID
- IG2T-TRANSACTION-TYPE-REST

See the definitions for the previous Partner/Qualifier mode section for more information about these fields.

IG2T-UNG-SENDER-ID

Description: *Sender ID*. Enter the Sender ID to use when generating the UNG segment (UNG02:01).

**Note:** This field supports the use of underscore substitution characters (USCs). To use a USC, the substitution character must exist on the Partner Header record.

See the *Gentran:Basic for zSeries Release 6.4 User's Guide* for more information about this feature.

IG2T-UNG-SENDER-ID-QUAL

Description: *Sender ID Qualifier*. Enter the Sender ID and Qualifier used to generate the UNG segment (UNG02:02).



## IG2T-UNG-RECEIVER-ID

Description: *Receiver ID*. Enter the Receiver ID to use when generating the UNG segment (UNG03:01).

**Note:** This field supports the use of underscore substitution characters (USCs). To use a USC, the substitution character must exist on the Partner Header record.

See the *Gentran:Basic for zSeries Release 6.4 User's Guide* for more information about this feature.

## IG2T-UNG-RECEIVER-ID-QUAL

Description: *Receiver ID Qualifier*. Enter the Receiver ID and Qualifier used to generate the UNG segment (UNG03:02).

## IG2T-VERSION

Description: *Version*. Associate the version/industry/association ID with this group or group by version (left-justified and space-filled).

## IG2T-MULTIPLE-ENVELOPE-ID

Description: *Multiple Envelope ID*. Determine which interchange envelope to generate for outbound transmissions.

## Valid Value(s):

ISA	=	Generate ISA, IEA control envelopes
ICS	=	Generate ICS, ICE control envelopes
BG	=	Generate BG-EG control envelopes
GS	=	Do not generate transmission control envelopes; use GS, GE instead
SCH	=	Generate SCH, END
STX	=	Generate STX, END
UNA	=	Generate UNA, UNB, UNZ control envelopes
UNB	=	Generate UNB, UNZ control envelopes

## Transaction Records

### TRANMISC Record

#### Description

The TRANMISC transaction updates fields on the partner profile Transaction records, including:

- Which control segment to generate for this partner and transaction
- Whether to acknowledge the transaction (and how)
- Whether this is a test or production transaction

#### Record Layout

Name	Data Type	Position
01 ITMT-INPUT-TRAN-MISC-TRAN.		
05 ITMT-ACTION-CODE	X(01)	001 - 001
05 ITMT-PARTNER-KEY.		
10 ITMT-PARTNER-ID	X(35)	002 - 036
10 ITMT-PARTNER-ID-QUAL	X(04)	037 - 040
10 ITMT-RECORD-TYPE.		
15 ITMT-RECORD-TYPE-VAL-TC	X(08)	041 - 048
15 ITMT-RECORD-TYPE-TRANS-ID.		
20 ITMT-TRANS-ID-CHAR-1	X(01)	049 - 049
20 ITMT-REST-OF-ID	X(05)	050 - 054
10 ITMT-RECORD-TYPE-REL-NUM	X(01)	055 - 055
10 ITMT-TRANSACTION-TYPE-REST	X(01)	056 - 056
05 ITMT-USRPAR-KEY		
REDEFINES ITMT-PARTNER-KEY.		
10 ITMT-USRPAR-USER-ID	X(15)	002 - 016
10 ITMT-USRPAR-PARTNER-ID	X(15)	017 - 031
10 FILLER	X(09)	032 - 040
10 ITMT-USRPAR-RECORD-TYPE	X(16)	041 - 056
05 ITMT-TRANS-CNTL-SEGMENT-SELECT	X(03)	057 - 059
05 ITMT-EXPECT-TRANS-ACK	X(01)	060 - 060
05 ITMT-TRANSACTION-ACK-TRANS	X(06)	061 - 066
05 ITMT-USE-ACK-TRANS	X(01)	067 - 067
05 ITMT-TEST-TRANSACTION	X(01)	068 - 068
05 ITMT-FUNCTIONAL-GROUP-ID	X(06)	069 - 074
05 ITMT-VERSION	X(12)	075 - 086
05 ITMT-VIEWPOINT	X(01)	087 - 087
05 ITMT-VIEWPOINT-TM	X(01)	088 - 088
05 ITMT-ST03-IMPLEMENT-CONVENTION	X(35)	089 - 123
05 ITMT-MULTIPLE-ENVELOPE-ID	X(03)	124 - 126

**TRANMISC FMS Transaction Definitions**

ITMT-ACTION-CODE (Required)

Description: *Transaction Action Code*

Valid Value(s):

A	=	Add
C	=	Change
D	=	Delete

**Partner/Qualifier Mode**

ITMT-PARTNER-ID (Required)

Description: *Partner ID*

ITMT-PARTNER-ID-QUAL (Required)

Description: *Partner ID Qualifier*

ITMT-RECORD-TYPE-VAL-TC (Required)

Description: *Transaction Type*

Valid Value(s):

Constant TRANMISC (left-justified and space-filled).

ITMT-RECORD-TYPE-TRANS-ID

Description: *Record Type Transaction ID*. Identify the transaction controlled by this record (left-justified and space-filled).

ITMT-RECORD-TYPE-REL-NUM

Description: *Record Type Release Number*. Used for TRADACOMS partners. Enter the release number for TRADACOMS Transactions ID.

ITMT-TRANSACTION-TYPE-REST

Description: *Transaction Type Rest*. This field is blank.

Valid Value(s):

Spaces

**Relationship Mode**

ITMT-USRPAR-USER-ID (Required)

Description: *User ID*

ITMT-USRPAR-PARTNER-ID (Required)

Description: *Partner ID*

ITMT-USRPAR-RECORD-TYPE (Required)

Description: The following fields are used to build this field:

- ITMT-RECORD-TYPE-VAL-TC
- ITMT-RECORD-TYPE-TRANS-ID
- ITMT-RECORD-TYPE-REL-NUM
- ITMT-TRANSACTION-TYPE-REST

See the definitions for the previous Partner/Qualifier Mode section for more information about these fields.

## ITMT-TRANS-CNTL-SEGMENT-SELECT

Description: *Transaction Control Segment Select.* Select which control segment to generate for this transaction.

## ITMT-EXPECT-TRANS-ACK

Description: *Expect Transaction Acknowledgement.* Specify whether the partner should return acknowledgments.

Valid Value(s):

Y	=	Expect acknowledgments from the partner
N	=	Do not expect acknowledgments from the partner
Blank	=	Do not expect acknowledgments from the partner

## ITMT-TRANSACTION-ACK-TRANS

Description: *Transaction Acknowledgement Transaction.* Specify the transaction to use to acknowledge this transaction after it is received.

Valid Value(s):

997  
999

Default Value(s):

997

## ITMT-USE-ACK-TRANS

Description: *Use Acknowledgment Transaction.* Indicate whether to use ITMT-TRANSACTION ACK-TRANS.

Valid Value(s):

Y	=	ITMT-TRANSACTION ACK-TRANS is used
N	=	ITMT-TRANSACTION ACK-TRANS is <i>not</i> used

## ITMT-TEST-TRANSACTION

Description: *Test Transaction.* Indicate to the application whether the transaction is production or test for this partner. Determine test and production splitting for this partner.

## ITMT-FUNCTIONAL-GROUP-ID

Description: *Functional Group ID.* Enter the Functional Group ID for the transaction assigned by the responsible agency or the primary detail message within EDIFACT or TRADACOMS.

## ITMT-VERSION

Description: *Version.* Associate the version/industry/association ID with this group or group by version (left-justified and space-filled).

## ITMT-VIEWPOINT

Description: *Viewpoint.* Indicate whether Gentran:Viewpoint Exception Management activity should be performed at the transaction level.

Valid Value(s):

Y	=	Gentran:Viewpoint Exception Management is performed.
N	=	Gentran:Viewpoint Exception Management is not performed.
Blank	=	Defaults to Y and the Exception Management processing is controlled by the Configuration File Exception Management Partner Default Indicator.

## ITMT-VIEWPOINT-TM

Description: *Viewpoint Tracking Management*. Indicate whether Gentran:Viewpoint tracking management activity should be performed at the group level.

Valid Value(s):

Y	=	Gentran:Viewpoint Tracking Management is performed.
N	=	Gentran:Viewpoint Tracking Management is not performed.
Blank	=	Defaults to Y and the Tracking Management processing is controlled by the Configuration File Tracking Management Partner Default Indicator.

## ITMT-ST03-IMPLEMENT-CONVENTION

Description: *ST03 Implement Convention*. The reference assigned to identify the implementation conversion.

Valid Value(s):

1-35 alphanumeric (left-justified and space-filled).

## ITMT-MULTIPLE-ENVELOPE-ID

Description: *Multiple Envelope ID*. Determine which interchange envelope to generate for outbound transmissions.

Valid Value(s):

ISA	=	Generate ISA, IEA control envelopes
ICS	=	Generate ICS, ICE control envelopes
BG	=	Generate BG-EG control envelopes
GS	=	Do not generate transmission control envelopes; use GS, GE instead
SCH	=	Generate SCH, END
STX	=	Generate STX, END
UNA	=	Generate UNA, UNB, UNZ control envelopes
UNB	=	Generate UNB, UNZ control envelopes

**TRANIN Record***Description*

The TRANIN record indicates:

- Whether to accept this transaction from this partner
- Whether to sequentially increment transaction control numbers

*Record Layout*

Name	Data Type	Position
01 ITIT-INPUT-TRAN-INCOMING-TRAN.		
05 ITIT-ACTION-CODE	X(01)	001 - 001
05 ITIT-PARTNER-KEY.		
10 ITIT-PARTNER-ID	X(35)	002 - 036
10 ITIT-PARTNER-ID-QUAL	X(04)	037 - 040
10 ITIT-RECORD-TYPE.		
15 ITIT-RECORD-TYPE-VAL-TC	X(06)	041 - 046
15 ITIT-RECORD-TYPE-TRANS-ID.		
20 ITIT-TRANS-ID-CHAR-1	X(01)	047 - 047
20 ITIT-REST-OF-ID	X(05)	048 - 052
10 ITIT-RECORD-TYPE-REL-NUM	X(01)	053 - 053
10 ITIT-TRANSACTION-TYPE-REST	X(03)	054 - 056
05 ITIT-USRPAR-KEY		
REDEFINES ITIT-PARTNER-KEY.		
10 ITIT-USRPAR-USER-ID	X(15)	002 - 016
10 ITIT-USRPAR-PARTNER-ID	X(15)	017 - 031
10 FILLER	X(09)	032 - 040
10 ITIT-USRPAR-RECORD-TYPE	X(16)	041 - 056
05 ITIT-US-PARAMETERS.		
10 ITIT-TRANSACTION-ACCEPT-FLAG	X(01)	057 - 057
10 ITIT-LAST-INBD-ST-CNTL-NO	X(09)	058 - 066
05 ITIT-INTERNATIONAL-PARAMETERS.		
10 ITIT-LAST-INBD-INTL-CNTL-NO	X(14)	067 - 080
10 ITIT-INBD-INTL-VERSION-USE	X(01)	081 - 081
10 ITIT-INBD-INTL-VERSION	X(12)	082 - 093
10 ITIT-TRANSLATION-ID-INBOUND	X(10)	094 - 103
05 ITIT-TRANS-INBD-DB-LEVEL	X(01)	104 - 104
05 ITIT-APPL-INBD-DB-LEVEL	X(01)	105 - 105
05 ITIT-VERSION	X(12)	106 - 117
05 ITIT-MULTIPLE-ENVELOPE-ID	X(03)	118 - 120

*TRANIN FMS Transaction Definitions*

ITIT-ACTION-CODE (Required)

Description: *Transaction Action Code*

Valid Value(s):

A	=	Add
C	=	Change
D	=	Delete

**Partner/Qualifier Mode**

ITIT-PARTNER-ID (Required)

Description: *Partner ID*

ITIT-PARTNER-ID-QUAL (Required)

Description: *Partner ID Qualifier*

ITIT-RECORD-TYPE-VAL-TC (Required)

Description: *Transaction Type*

Valid Value(s):

Constant TRANIN (left-justified and space-filled).

ITIT-RECORD-TYPE-TRANS-ID

Description: *Record Type Transaction ID*. Transaction ID for the transaction controlled by this record (left-justified and space-filled).

ITIT-RECORD-TYPE-REL-NUM

Description: *Record Type Release Number*. Used for TRADACOMS partners. Release number for TRADACOMS Transaction ID.

ITIT-TRANSACTION-TYPE-REST

Description: *Transaction Type Rest*. This field is blank.

Valid Value(s):

Spaces

**Relationship Mode**

ITIT-USRPAR-USER-ID (Required)

Description: *User ID*

ITIT-USRPAR-PARTNER-ID (Required)

Description: *Partner ID*

ITIT-USRPAR-RECORD-TYPE (Required)

Description: The following fields are used to build this field:

- ITIT-RECORD-TYPE-VAL-TC
- ITIT-RECORD-TYPE-TRANS-ID
- ITIT-RECORD-TYPE-REL-NUM
- ITIT-TRANSACTION-TYPE-REST

See the definitions for the previous Partner/Qualifier mode section for more information about these fields.

## ITIT-TRANSACTION-ACCEPT-FLAG

Description: *Transaction Accept Flag*. Indicate whether to accept this transaction from this partner.

Valid Value(s):

Y	=	Accept this transaction
N	=	Do not accept this transaction
Blank	=	Accept this transaction

Default Value(s):

Accept this transaction

## ITIT-LAST-INBD-ST-CNTL-NO

Reserved for future use.

## ITIT-LAST-INBD-INTL-CNTL-NO

Reserved for future use.

## ITIT-INBD-INTL-VERSION-USE

Reserved for future use.

## ITIT-INBD-INTL-VERSION

Reserved for future use.

## ITIT-TRANSLATION-ID-INBOUND

Description: *Transaction ID Inbound*. Apply a Transaction ID for the for partner-specific map.

## ITIT-TRANS-INBD-DB-LEVEL

Description: *Transaction Inbound Databank Level*. Indicate whether to set the inbound EDI transaction databanking level for directory maintenance.

Valid Value(s):

D	=	Set for directory maintenance
N	=	Not set for directory maintenance

## ITIT-APPL-INBD-DB-LEVEL

Description: *Application Inbound Databank Level*. Set the partner profile inbound application databanking level for the partner.

Valid Value(s):

F	=	Full
D	=	Directory level
N	=	None

## ITIT-VERSION

Description: *Version*. Associate the version/industry/association ID with this transaction or transaction by version (left-justified and space-filled).



## ITIT-MULTIPLE-ENVELOPE-ID

Description: *Multiple Envelope ID*. Determine which interchange envelope to generate for outbound transmissions.

## Valid Value(s):

ISA	=	Generate ISA, IEA control envelopes
ICS	=	Generate ICS, ICE control envelopes
BG	=	Generate BG-EG control envelopes
GS	=	Do not generate transmission control envelopes; use GS, GE instead
SCH	=	Generate SCH, END
STX	=	Generate STX, END
UNA	=	Generate UNA, UNB, UNZ control envelopes
UNB	=	Generate UNB, UNZ control envelopes

**TRANOUT Record***Description*

The TRANOUT transaction indicates whether to send this transaction to this partner.

*Record Layout*

Name	Data Type	Position
01 ITOT-INPUT-TRAN-OUTGOING-TRAN.		
05 ITOT-ACTION-CODE	X(01)	001 - 001
05 ITOT-PARTNER-KEY.		
10 ITOT-PARTNER-ID	X(35)	002 - 036
10 ITOT-PARTNER-ID-QUAL	X(04)	037 - 040
10 ITOT-RECORD-TYPE.		
15 ITOT-RECORD-TYPE-VAL-TC	X(07)	041 - 047
15 ITOT-RECORD-TYPE-TRANS-ID.		
20 ITOT-TRANS-ID-CHAR-1	X(01)	048 - 048
20 ITOT-REST-OF-ID	X(05)	049 - 053
10 ITOT-RECORD-TYPE-REL-NUM	X(01)	054 - 054
10 ITOT-TRANSACTION-TYPE-REST	X(02)	055 - 056
05 ITOT-USRPAR-KEY		
REDEFINES ITOT-PARTNER-KEY.		
10 ITOT-USRPAR-USER-ID	X(15)	002 - 016
10 ITOT-USRPAR-PARTNER-ID	X(15)	017 - 031
10 FILLER	X(09)	032 - 040
10 ITOT-USRPAR-RECORD-TYPE	X(16)	041 - 056
05 ITOT-US-PARAMETERS.		
10 ITOT-TRANSACTION-SEND-FLAG	X(01)	057 - 057
10 ITOT-LAST-OUTBD-ST-CNTL-NO	X(09)	058 - 066
05 ITOT-INTERNATIONAL-PARAMETERS.		
10 ITOT-LAST-OUTBD-INTL-CNTL-NO	X(14)	067 - 080
10 ITOT-OUTBD-INTL-VERSION-USE	X(01)	081 - 081
10 ITOT-OUTBD-INTL-VERSION	X(12)	082 - 093
10 ITOT-TRANSLATION-ID-OUTBOUND	X(10)	094 - 103
05 ITOT-TRANS-OUTBD-DB-LEVEL	X(01)	104 - 104
05 ITOT-APPL-OUTBD-DB-LEVEL	X(01)	105 - 105
05 ITOT-VERSION	X(12)	106 - 117
05 ITOT-MULTIPLE-EVENLOPE-ID	X(03)	118 - 120

*TRANOUT FMS Transaction Definitions*

ITOT-ACTION-CODE (Required)

Description: *Transaction Action Code*

Valid Value(s):

A	=	Add
C	=	Change
D	=	Delete

**Partner/Qualifier Mode**

ITOT-PARTNER-ID (Required)

Description: *Partner ID*

ITOT-PARTNER-ID-QUAL (Required)

Description: *Partner ID Qualifier*

ITOT-RECORD-TYPE-VAL-TC (Required)

Description: *Transaction Type*

Valid Value(s):

Constant TRANOUT (left-justified and space-filled).

ITOT-RECORD-TYPE-TRANS-ID

Description: *Record Type Transaction ID*. Enter the Transaction ID for the transaction controlled by this record (left-justified and space-filled).

ITOT-RECORD-TYPE-REL-NUM

Description: *Record Type Release Number*. Used for TRADACOMS partners. Enter the release number for TRADACOMS Transaction ID.

ITOT-TRANSACTION-TYPE-REST

Description: *Transaction Type Rest*. This field is blank.

Valid Value(s):

Spaces

**Relationship Mode**

ITOT-USRPAR-USER-ID (Required)

Description: *User ID*

ITOT-USRPAR-PARTNER-ID (Required)

Description: *Partner ID*

ITOT-USRPAR-RECORD-TYPE (Required)

Description: The following fields are used to build this field:

- ITOT-RECORD-TYPE-VAL-TC
- ITOT-RECORD-TYPE-TRANS-ID
- ITOT-RECORD-TYPE-REL-NUM
- ITOT-TRANSACTION-TYPE-REST

See the definitions for the previous Partner/Qualifier mode section for more information about these fields.

ITOT-TRANSACTION-SEND-FLAG

Description: *Transaction Send Flag*. Indicate whether to send this transaction to this partner.

Valid Value(s):

Y	=	Send this transaction
N	=	Do not send this transaction
Blank	=	Do not send this transaction

## ITOT-LAST-OUTBD-ST-CNTL-NO

Reserved for future use.

## ITOT-LAST-OUTBD-INTL-CNTL-NO

Reserved for future use.

## ITOT-OUTBD-INTL-VERSION-USE

Reserved for future use.

## ITOT-OUTBD-INTL-VERSION

Reserved for future use.

## ITOT-TRANSLATION-ID-OUTBOUND

Description: *Translation ID Outbound*. Enter the Translation ID for a partner-specific map.

## ITOT-TRANS-OUTBD-DB-LEVEL

Description: *Transaction Outbound Databank Level*. Indicate whether to set the inbound EDI transaction databanking level for directory maintenance.

Valid Value(s):

D	=	Set for directory level maintenance
N	=	Do not set for directory level maintenance

## ITOT-APPL-OUTBD-DB-LEVEL

Description: *Application Outbound Databank Level*. Set the partner profile outbound application databanking level for partner.

Valid Value(s):

F	=	Full
D	=	Directory level
N	=	None

## ITOT-VERSION

Description: *Version*. Associate the version/industry/association ID with this transaction or transaction by version (left-justified and space-filled).

## ITOT-MULTIPLE-ENVELOPE-ID

Description: *Multiple Envelope ID*. Determine which interchange envelope to generate for outbound transmissions.

Valid Value(s):

ISA	=	Generate ISA, IEA control envelopes
ICS	=	Generate ICS, ICE control envelopes
BG	=	Generate BG-EG control envelopes
GS	=	Do not generate transmission control envelopes; use GS, GE instead
SCH	=	Generate SCH, END
STX	=	Generate STX, END
UNA	=	Generate UNA, UNB, UNZ control envelopes
UNB	=	Generate UNB, UNZ control envelopes

**GENST Record***Description*

The GENST transaction controls the version and the transaction control number for outbound interchanges. If the Transaction Control Segment is equal to spaces, and this transaction is submitted, ST is forced into the Transaction Control Segment. If the Transaction Control Segment contains any other value, an error will result.

*Record Layout*

Name	Data Type	Position
01 ISTT-INPUT-TRAN-GEN-ST-TRAN.		
05 ISTT-ACTION-CODE	X(01)	001 - 001
05 ISTT-PARTNER-KEY.		
10 ISTT-PARTNER-ID	X(35)	002 - 036
10 ISTT-PARTNER-ID-QUAL	X(04)	037 - 040
10 ISTT-RECORD-TYPE.		
15 ISTT-RECORD-TYPE-VAL-TC	X(05)	041 - 045
15 ISTT-RECORD-TYPE-TRANS-ID.		
20 ISTT-TRANS-ID-CHAR-1	X(01)	046 - 046
20 ISTT-REST-OF-ID	X(05)	047 - 051
10 ISTT-TRANSACTION-TYPE-REST	X(05)	051 - 056
05 ISTT-USRPAR-KEY		
REDEFINES ISTT-PARTNER-KEY.		
10 ISTT-USRPAR-USER-ID	X(15)	002 - 016
10 ISTT-USRPAR-PARTNER-ID	X(15)	017 - 031
10 FILLER	X(09)	032 - 040
10 ISTT-USRPAR-RECORD-TYPE	X(16)	041 - 056
05 ISTT-TRANSACTION-CNTL-NUMBER	X(09)	057 - 065
05 ISTT-TRANSACTION-SEND-VERSION	X(12)	066 - 077
05 ISTT-VERSION	X(12)	078 - 089
05 ISTT-MULTIPLE-ENVELOPE-ID	X(03)	090 - 092

*GENST FMS Transaction Definitions*

## ISTT-ACTION-CODE (Required)

Description: *Transaction Action Code*

Valid Value(s):

A	=	Add
C	=	Change
D	=	Delete

**Partner/Qualifier Mode**

ISTT-PARTNER-ID (Required)

Description: *Partner ID*

ISTT-PARTNER-ID-QUAL (Required)

Description: *Partner ID Qualifier*

## ISTT-RECORD-TYPE-VAL-TC (Required)

Description: *Transaction Type*

Valid Value(s):

Constant GENST (left-justified and space-filled).

## ISTT-RECORD-TYPE-TRANS-ID

Description: *Record Type Transaction ID*. Enter the Transaction ID for the transaction controlled by this record (left-justified and space-filled).

## ISTT-TRANSACTION-TYPE-REST

Description: *Transaction Type Rest*. This field is blank.

Valid Value(s):

Spaces

**Relationship Mode**

## ISTT-USRPAR-USER-ID (Required)

Description: *User ID*

## ISTT-USRPAR-PARTNER-ID (Required)

Description: *Partner ID*

## ISTT-USRPAR-RECORD-TYPE (Required)

Description: The following fields are used to build this field:

- ISTT-RECORD-TYPE-VAL-TC
- ISTT-RECORD-TYPE-TRANS-ID
- ISTT-TRANSACTION-TYPE-REST

See the definitions for the previous Partner/Qualifier mode section for more information about these fields.

## ISTT-TRANSACTION-CNTL-NUMBER

Description: *Transaction Control Number*. Set up how Gentran checks the control number for this transaction (ST02).

Valid Value(s):

Zeros	=	Interchange control number checking starts at 1
Non-zero numerals	=	Interchange control number checking begins at n + 1
Blank	=	Checking function is not used

## ISTT-TRANSACTION-SEND-VERSION

Description: *Transaction Send Version*. Enter the version to use when sending this transaction.

## ISTT-VERSION

Description: *Version*. Associate the version/industry/association ID with this transaction for transaction by version (left-justified and space-filled).

## ISTT-MULTIPLE-ENVELOPE-ID

Description: *Multiple Envelope ID*. Determine which interchange envelope to generate for outbound transmissions.

## Valid Value(s):

ISA	=	Generate ISA, IEA control envelopes
ICS	=	Generate ICS, ICE control envelopes
BG	=	Generate BG-EG control envelopes
GS	=	Do not generate transmission control envelopes; use GS, GE instead
SCH	=	Generate SCH, END
STX	=	Generate STX, END
UNA	=	Generate UNA, UNB, UNZ control envelopes
UNB	=	Generate UNB, UNZ control envelopes

**GENUNH Record***Description*

The GENUNH transaction contains the information to use to generate UNH control segments.

If the Transaction Control Segment is equal to spaces, and this transaction is submitted, UNH is forced into the Transaction Control Segment. If the Transaction Control Segment is equal to ST, the error 542 will result.

*Record Layout*

<b>Name</b>	<b>Data Type</b>	<b>Position</b>
01 IGH T-INPUT-TRAN-GEN-UNH-TRAN .		
05 IGH T-ACTION-CODE	X(01)	001 - 001
05 IGH T-PARTNER-KEY .		
10 IGH T-PARTNER-ID	X(35)	002 - 036
10 IGH T-PARTNER-ID-QUAL	X(04)	037 - 040
10 IGH T-RECORD-TYPE .		
15 IGH T-RECORD-TYPE-VAL-TC	X(06)	041 - 046
15 IGH T-RECORD-TYPE-TRANS-ID .		
20 IGH T-TRANS-ID-CHAR-1	X(01)	047 - 047
20 IGH T-REST-OF-ID	X(05)	048 - 052
10 IGH T-TRANSACTION-TYPE-REST	X(04)	053 - 056
05 IGH T-USRPAR-KEY		
REDEFINES IGH T-PARTNER-KEY .		
10 IGH T-USRPAR-USER-ID	X(15)	002 - 016
10 IGH T-USRPAR-PARTNER-ID	X(15)	017 - 031
10 FILLER	X(09)	032 - 040
10 IGH T-USRPAR-RECORD-TYPE	X(16)	041 - 056
05 IGH T-UNH-MESSAGE-REFERENCE-NBR	X(14)	057 - 070
05 IGH T-UNH-VERSION .		
10 IGH T-UNH-VERSION-NUMBER	X(03)	071 - 073
10 IGH T-UNH-VERSION-RELEASE	X(03)	074 - 076
10 IGH T-UNH-VERSION-ASSOCIATN	X(06)	077 - 082
05 IGH T-UNH-CONTROLLING-AGENCY	X(03)	083 - 085
05 IGH T-UNH-COMMON-ACCESS-REF	X(35)	086 - 120
05 IGH T-UNH-SEQUENCE-OF-TRANSFER	X(02)	121 - 122
05 IGH T-UNH-FIRST-N-LAST-TRANSFER	X(01)	123 - 123
05 IGH T-VERSION	X(12)	124 - 135
05 IGH T-UNH-CODE-LIST-VERSION	X(06)	136 - 141
05 IGH T-UNH-MSG-SUB-FUNCTION	X(06)	142 - 147
05 IGH T-UNH-MSG-SUBSET-ID	X(14)	148 - 161
05 IGH T-UNH-MSG-SUBSET-VERSION	X(03)	162 - 164
05 IGH T-UNH-MSG-SUBSET-RELEASE	X(03)	165 - 167
05 IGH T-UNH-MSG-SUBSET-AGENCY	X(03)	168 - 170
05 IGH T-UNH-IMP-GUIDE-ID	X(14)	171 - 184
05 IGH T-UNH-IMP-GUIDE-VERSION	X(03)	185 - 187
05 IGH T-UNH-IMP-GUIDE-RELEASE	X(03)	188 - 190
05 IGH T-UNH-IMP-GUIDE-AGENCY	X(03)	191 - 193
05 IGH T-UNH-SCENARIO-ID	X(14)	194 - 207
05 IGH T-UNH-SCENARIO-VERSION	X(03)	208 - 210
05 IGH T-UNH-SCENARIO-RELEASE	X(03)	211 - 213
05 IGH T-UNH-SCENARIO-AGENCY	X(03)	214 - 216
05 IGH T-MULTIPLE-ENVELOPE-ID	X(03)	217 - 219



**GENUNH FMS Transaction Definitions**

IGHT-ACTION-CODE (Required)

Description: *Transaction Action Code*

Valid Value(s):

A	=	Add
C	=	Change
D	=	Delete

**Partner/Qualifier Mode**

IGHT-PARTNER-ID (Required)

Description: *Partner ID*

IGHT-PARTNER-ID-QUAL (Required)

Description: *Partner ID Qualifier*

IGHT-RECORD-TYPE-VAL-TC (Required)

Description: *Transaction Type*

Valid Value(s):

Constant GENUNH (left-justified and space-filled).

IGHT-RECORD-TYPE-TRANS-ID

Description: *Record Type Transaction ID*. Enter the Transaction ID for the transaction controlled by this record (left-justified and space-filled).

IGHT-TRANSACTION-TYPE-REST

Description: *Transaction Type Rest*. This field is blank.

Valid Value(s):

Spaces

**Relationship Mode**

IGHT-USRPAR-USER-ID (Required)

Description: *User ID*

IGHT-USRPAR-PARTNER-ID (Required)

Description: *Partner ID*

IGHT-USRPAR-RECORD-TYPE (Required)

Description: The following fields are used to build this field:

- IGH-RECORD-TYPE-VAL-TC
- IGH-RECORD-TYPE-TRANS-ID
- IGH-TRANSACTION-TYPE-REST

See the definitions for the previous Partner/Qualifier mode section for more information about these fields.

IGHT-UNH-MESSAGE-REFERENCE-NBR

Description: *Message Reference Number*. Enter the message reference number to use when generating the UNH segment (UNH01).

## IGHT-UNH-VERSION-NUMBER

Description: *Version Number*. Enter the version number to use when generating the UNH segment (UNH02:02).

## IGHT-UNH-VERSION-RELEASE

Description: *Version Release*. Enter the version release to use when generating the UNH segment (UNH02:03).

## IGHT-UNH-VERSION-ASSOCIATN

Description: *Version Association*. Enter the version association assigned part of the version to use when generating the UNH segment (UNH02:05).

## IGHT-UNH-CONTROLLING-AGENCY

Description: *Controlling Agency*. Enter the controlling agency to use when generating the UNH segment (UNH02:04).

## IGHT-UNH-COMMON-ACCESS-REF

Description: *Common Access Reference*. Enter the access reference to use when generating the UNH segment (UNH03).

## IGHT-UNH-SEQUENCE-OF-TRANSFER

Description: *Sequence of Transfer*. Enter the sequence of the transfer to use when generating the UNH segment (UNH04:01).

## IGHT-UNH-FIRST-N-LAST-TRANSFER

Description: *First and Last Transfer*. Enter the first and last transfer to use when generating the UNH segment (UNH04:02).

## IGHT-VERSION

Description: *Version*. Associate the version/industry/association ID with this transaction or transaction by version (left-justified and space-filled).

## IGHT-UNH-CODE-LIST-VERSION

Description: *Code List Version*. Define the code list directory version number. This is valid for Syntax version 4 and greater.

## IGHT-UNH-MSG-SUB-FUNCTION

Description: *Message Sub-function*. Define the message type sub-function. This is valid for Syntax version 4+.

## IGHT-UNH-MSG-SUBSET-ID

Description: *Message Subset ID*. Define the Message Subset ID.

## IGHT-UNH-MSG-SUBSET-VERSION

Description: *Message Subset Version*. Define the message subset version number.

## IGHT-UNH-MSG-SUBSET-RELEASE

Description: *Message Subset Release*. Define the message subset release number.

## IGHT-UNH-MSG-SUBSET-AGENCY

Description: *Message Subset Agency*. Define the controlling agency for this message subset.

## IGHT-UNH-IMP-GUIDE-ID

Description: *Implementation Guideline ID*. Define the message implementation guideline ID.

## IGHT-UNH-IMP-GUIDE-VERSION

Description: *Implementation Guideline Version*. Define the message implementation guideline version number.

## IGHT-UNH-IMP-GUIDE-RELEASE

Description: *Implementation Guideline Release*. Define the message implementation guideline release number.

## IGHT-UNH-IMP-GUIDE-AGENCY

Description: *Implementation Guideline Agency*. Define the controlling agency for this message implementation guideline.

## IGHT-UNH-SCENARIO-ID

Description: *Scenario ID*. Define the Scenario ID.

## IGHT-UNH-SCENARIO-VERSION

Description: *Scenario Version*. Define the scenario version number.

## IGHT-UNH-SCENARIO-RELEASE

Description: *Scenario Release*. Define the scenario release number.

## IGHT-UNH-SCENARIO-AGENCY

Description: *Scenario Agency*. Define the controlling agency for this scenario.

## IGHT-MULTIPLE-ENVELOPE-ID

Description: *Multiple Envelope ID*. Determine which interchange envelope to generate for outbound transmissions.

Valid Value(s):

ISA	=	Generate ISA, IEA control envelopes
ICS	=	Generate ICS, ICE control envelopes
BG	=	Generate BG-EG control envelopes
GS	=	Do not generate transmission control envelopes; use GS, GE instead
SCH	=	Generate SCH, END
STX	=	Generate STX, END
UNA	=	Generate UNA, UNB, UNZ control envelopes
UNB	=	Generate UNB, UNZ control envelopes

**GENMHD Record***Description*

The GENMHD transaction contains the information to use to generate MHD control segments.

If the Transaction Control Segment is equal to spaces, and this transaction is submitted, MHD is forced into the Transaction Control Segment. If the Transaction Control Segment is equal to ST, the error 542 will result.

*Record Layout*

Name	Data Type	Position
01 IMHT-INPUT-TRAN-GEN-MHD-TRAN.		
05 IMHT-ACTION-CODE	X(01)	001 - 001
05 IMHT-PARTNER-KEY.		
10 IMHT-PARTNER-ID	X(35)	002 - 036
10 IMHT-PARTNER-ID-QUAL	X(04)	037 - 040
10 IMHT-RECORD-TYPE.		
15 IMHT-RECORD-TYPE-VAL-TC	X(06)	041 - 046
15 IMHT-RECORD-TYPE-TRANS-ID.		
20 IMHT-TRANS-ID-CHAR-1	X(01)	047 - 047
20 IMHT-REST-OF-ID	X(05)	048 - 052
10 IMHT-RECORD-TYPE-REL-NUM	X(01)	053 - 053
10 IMHT-TRANSACTION-TYPE-REST	X(03)	054 - 056
05 IMHT-USRPAR-KEY		
REDEFINES IMHT-PARTNER-KEY.		
10 IMHT-USRPAR-USER-ID	X(15)	002 - 016
10 IMHT-USRPAR-PARTNER-ID	X(15)	017 - 031
10 FILLER	X(09)	032 - 040
10 IMHT-USRPAR-RECORD-TYPE	X(16)	041 - 056
05 IMHT-MHD-MESSAGE-REFERENCE-NBR	X(12)	057 - 068
05 IMHT-MHD-TRANSACTION RELEASE	X(01)	069 - 069
05 FILLER	X(01)	070 - 070
05 IMHT-MHD-VERSION.		
10 IMHT-MHD-VERSION-NUMBER	X(03)	071 - 073
10 IMHT-MHD-VERSION-RELEASE	X(03)	074 - 076
10 IMHT-MHD-VERSION-ASSOCIATN	X(06)	077 - 082
05 IMHT-MHD-CONTROLLING-AGENCY	X(03)	083 - 085
05 IMHT-VERSION	X(12)	086 - 097
05 IMHT-MULTIPLE-ENVELOPE-ID	X(03)	098 - 100

*GENMHD FMS Transaction Definitions*

IMHT-ACTION-CODE (Required)

Description: *Transaction Action Code*

Valid Value(s):

A	=	Add
C	=	Change
D	=	Delete

**Partner/Qualifier Mode**

IMHT-PARTNER-ID (Required)

Description: *Partner ID*

IMHT-PARTNER-ID-QUAL (Required)

Description: *Partner ID Qualifier*

IMHT-RECORD-TYPE-VAL-TC (Required)

Description: *Transaction Type*

Valid Value(s):

Constant GENMHD (left-justified and space-filled).

IMHT-RECORD-TYPE-TRANS-ID

Description: *Record Type Transaction ID*. Enter the Transaction ID for the transaction controlled by this record (left-justified, space-filled).

IMHT-RECORD-TYPE-REL-NUM

Description: *Record Type Release Number*. Used for TRADACOMS partners. Enter the release number for the TRADACOMS Transaction ID.

IMHT-TRANSACTION-TYPE-REST

Description: *Transaction Type Rest*. This field is blank.

Valid Value(s):

Spaces

**Relationship Mode**

IMHT-USRPAR-USER-ID (Required)

Description: *User ID*

IMHT-USRPAR-PARTNER-ID (Required)

Description: *Partner ID*

IMHT-USRPAR-RECORD-TYPE (Required)

Description: The following fields are used to build this field:

- IMHT-RECORD-TYPE-VAL-TC
- IMHT-RECORD-TYPE-TRANS-ID
- IMHT-RECORD-TYPE-REL-NUM
- IMHT-TRANSACTION-TYPE-REST

See the definitions for the previous Partner/Qualifier mode section for more information about these fields.

IMHT-MHD-MESSAGE-REFERENCE-NBR

Description: *Message Reference Number*. Enter the unique reference number assigned by your division or department.

IMHT-MHD-TRANSACTION-RELEASE

Description: *Transaction Release*. Enter the transaction release number for TRADACOMS standard.

## IMHT-MHD-VERSION-NUMBER

Description: *Version Number*. Enter the version number for TRADACOMS.

## IMHT-MHD-VERSION-RELEASE

Description: *Version Release*. Enter the release number for TRADACOMS

## IMHT-MHD-VERSION-ASSOCIATN

Description: *Version Association*. Enter the version association for TRADACOMS.

## IMHT-MHD-CONTROLLING-AGENCY

Description: *Controlling Agency*. Define the controlling agency.

## IMHT-VERSION

Description: *Version*. Associate the version/industry/association ID with this transaction or transaction by version (left-justified and space-filled).

## IMHT-MULTIPLE-ENVELOPE-ID

Description: *Multiple Envelope ID*. Determine which interchange envelope to generate outbound transmissions.

## Valid Value(s):

ISA	=	Generate ISA, IEA control envelopes
ICS	=	Generate ICS, ICE control envelopes
BG	=	Generate BG-EG control envelopes
GS	=	Do not generate transmission control envelopes; use GS, GE instead
SCH	=	Generate SCH, END
STX	=	Generate STX, END
UNA	=	Generate UNA, UNB, UNZ control envelopes
UNB	=	Generate UNB, UNZ control envelopes

## Name and Address Records

### NAMEAD1 Record

#### Description

The NAMEAD1, NAMEAD2, NAMEAD3, NAMEAD4, and NAMEAD5 records process partner name and address information used in the Audit Print program.

#### Record Layout

Name	Data Type	Position
01 IN1T-INPUT-NAME-ADDR1-TRANSACTION.		
05 IN1T-ACTION-CODE	X(01)	001 - 001
05 IN1T-PARTNER-KEY.		
10 IN1T-PARTNER-ID	X(35)	002 - 036
10 IN1T-PARTNER-ID-QUAL	X(04)	037 - 040
10 IN1T-TRANSACTION-TYPE	X(16)	041 - 056
05 IN1T-USRPAR-KEY		
REDEFINES IN1T-PARTNER-KEY.		
10 IN1T-USRPAR-USER-ID	X(15)	002 - 016
10 IN1T-USRPAR-PARTNER ID	X(15)	017 - 031
10 FILLER	X(09)	032 - 040
10 IN1T-USRPAR-RECORD-TYPE	X(16)	041 - 056
05 IN1T-NAME	X(35)	057 - 091
05 IN1T-ADDRESS1	X(35)	092 - 126
05 IN1T-USER-PART-RCD-TYPE	X(04)	127 - 130

#### NAMEAD1 FMS Transaction Definitions

IN1T-ACTION-CODE (Required)

Description: *Transaction Action Code*

Valid Value(s):

A	=	Add
C	=	Change
D	=	Delete

#### Partner/Qualifier Mode

IN1T-PARTNER-ID (Required)

Description: *Partner ID*

IN1T-PARTNER-ID-QUAL (Required)

Description: *Partner ID Qualifier*

IN1T-TRANSACTION-TYPE (Required)

Description: *Transaction Type*

Default Value(s):

Constant NAMEAD1 (left-justified and space-filled).

**Relationship Mode**

INIT-USRPAR-USER-ID (Required)

Description: *User ID*

INIT-USRPAR-PARTNER-ID (Required)

Description: *Partner ID*

INIT-USRPAR-RECORD-TYPE (Required)

Description: *Transaction Type*

Valid Value(s):

Constant NAMEAD1 (left-justified and space-filled).

INIT-NAME

Description: *Name*. Enter the name of the partner.

**Note:** This field supports the use of underscore substitution characters (USCs). To use a USC, the substitution character must exist on the Partner Header record.

See the *Gentran:Basic for zSeries Release 6.4 User's Guide* for more information about this feature.

INIT-ADDRESS1

Description: *Address 1*. Enter the address of the partner.

INIT-USER-PART-RCD-TYPE (Required)

Description: *Record Type*

Valid Value(s):

USER PART

---



**NAMEAD2 Record***Description*

The NAMEAD1, NAMEAD2, NAMEAD3, NAMEAD4, and NAMEAD5 records process partner name and address information used in the Audit Print program.

*Record Layout*

Name	Data Type	Position
01 IN2T-INPUT-NAME-ADDR2-TRANSACTION-CODE		
05 IN2T-ACTION-CODE	X (01)	001 - 001
05 IN2T-PARTNER-KEY		
10 IN2T-PARTNER-ID	X (35)	002 - 036
10 IN2T-PARTNER-ID-QUAL	X (04)	037 - 040
10 IN2T-TRANSACTION-TYPE	X (16)	041 - 056
05 IN2T-USRPAR-KEY		
REDEFINES IN2T-PARTNER-KEY		
10 IN2T-USRPAR-USER-ID	X (15)	002 - 016
10 IN2T-USRPAR-PARTNER-ID	X (15)	017 - 031
10 FILLER	X (09)	032 - 040
10 IN2T-USRPAR-RECORD-TYPE	X (16)	041 - 056
05 IN2T-ADDRESS2	X (35)	057 - 091
05 IN2T-ADDRESS3	X (35)	092 - 126
05 IN2T-USER-PART-RCD-TYPE	X (04)	127 - 130

**NAMEAD2 FMS Transaction Definitions**

IN2T-ACTION-CODE (Required)

Description: *Transaction Action Code*

Valid Value(s):

A	=	Add
C	=	Change
D	=	Delete

**Partner/Qualifier Mode**

IN2T-PARTNER-ID (Required)

Description: *Partner ID*

IN2T-PARTNER-ID-QUAL (Required)

Description: *Partner ID Qualifier*

IN2T-TRANSACTION-TYPE

Description: *Transaction Type*

Default Value(s):

Constant NAMEAD2

**Relationship Mode**

IN2T-USRPAR-USER-ID (Required)

Description: *User ID*

IN2T-USRPAR-PARTNER-ID (Required)

Description: *Partner ID*

IN2T-USRPAR-RECORD-TYPE (Required)

Description: *Transaction Type*

Valid Value(s):

Constant NAMEAD2 (left-justified and space-filled).

IN2T-ADDRESS2

Description: *Address 2*. Additional (second) line for the partner address. Enter applicable address information.

**Note:** This field supports the use of underscore substitution characters (USCs). To use a USC, the substitution character must exist on the Partner Header record.

See the *Gentran:Basic for zSeries Release 6.4 User's Guide* for more information about this feature.

IN2T-ADDRESS3

Description: *Address 3*. Additional (third) line for the partner address. Enter applicable address information.

**Note:** This field supports the use of underscore substitution characters (USCs). To use a USC, the substitution character must exist on the Partner Header record.

See the *Gentran:Basic for zSeries Release 6.4 User's Guide* for more information about this feature.

IN2T-USER-PART-RCD-TYPE (Required)

Description: *Record Type*

Valid Value(s):

USER PART

**NAMEAD3 Record***Description*

The NAMEAD1, NAMEAD2, NAMEAD3, NAMEAD4, and NAMEAD5 records process partner name and address information used in the Audit Print program.

*Record Layout*

Name	Data Type	Position
01 IN3T-INPUT-NAME-ADDR3-TRANSACTION-KEY		
05 IN3T-ACTION-CODE	X(01)	001 - 001
05 IN3T-PARTNER-KEY		
10 IN3T-PARTNER-ID	X(35)	002 - 036
10 IN3T-PARTNER-ID-QUAL	X(04)	037 - 040
10 IN3T-TRANSACTION-TYPE	X(16)	041 - 056
05 IN3T-USRPAR-KEY		
REDEFINES IN3T-PARTNER-KEY		
10 IN3T-USRPAR-USER-ID	X(15)	002 - 016
10 IN3T-USRPAR-PARTNER-ID	X(15)	017 - 031
10 FILLER	X(09)	032 - 040
10 IN3T-USRPAR-RECORD-TYPE	X(16)	041 - 056
05 IN3T-ADDRESS4	X(35)	057 - 091
05 IN3T-ADDRESS5	X(35)	092 - 126
05 IN3T-USER-PART-RCD-TYPE	X(04)	127 - 130

**NAMEAD3 FMS Transaction Definitions**

IN3T-ACTION-CODE (Required)

Description: *Transaction Action Code*

Valid Value(s):

A	=	Add
C	=	Change
D	=	Delete

**Partner/Qualifier Mode**

IN3T-PARTNER-ID (Required)

Description: *Partner ID*

IN3T-PARTNER-ID-QUAL (Required)

Description: *Partner ID Qualifier*

IN3T-TRANSACTION-TYPE

Description: *Transaction Type*

Default Value(s):

Constant NAMEAD3

**Relationship Mode**

IN3T-USRPAR-USER-ID (Required)

Description: *User ID*

IN3T-USRPAR-PARTNER-ID (Required)

Description: *Partner ID*

IN3T-USRPAR-RECORD-TYPE (Required)

Description: *Transaction Type*

Valid Value(s):

Constant NAMEAD3 (left-justified and space-filled).

IN3T-ADDRESS4

Description: *Address 4*. Additional (fourth line) for the partner address. Enter applicable address information.

**Note:** This field supports the use of underscore substitution characters (USCs). To use a USC, the substitution character must exist on the Partner Header record.

See the *Gentran:Basic for zSeries Release 6.4 User's Guide* for more information about this feature.

IN3T-ADDRESS5

Description: *Address 5*. Additional (fifth) line for the partner address. Enter applicable address information.

**Note:** This field supports the use of underscore substitution characters (USCs). To use a USC, the substitution character must exist on the Partner Header record.

See the *Gentran:Basic for zSeries Release 6.4 User's Guide* for more information about this feature.

IN3T-USER-PART-RCD-TYPE (Required)

Description: *Record Type*

Valid Value(s):

USER PART

**NAMEAD4 Record***Description*

The NAMEAD1, NAMEAD2, NAMEAD3, NAMEAD4, and NAMEAD5 records process partner name and address information used in the Audit Print program.

*Record Layout*

Name	Data Type	Position
01 IN4T-INPUT-NAME-ADDR4-TRANSACTION.		
05 IN4T-ACTION-CODE	X(01)	001 - 001
05 IN4T-PARTNER-KEY.		
10 IN4T-PARTNER-ID	X(35)	002 - 036
10 IN4T-PARTNER-ID-QUAL	X(04)	037 - 040
10 IN4T-TRANSACTION-TYPE	X(16)	041 - 056
05 IN4T-USRPAR-KEY		
REDEFINES IN4T-PARTNER-KEY.		
10 IN4T-USRPAR-USER-ID	X(15)	002 - 016
10 IN4T-USRPAR-PARTNER-ID	X(15)	017 - 031
10 FILLER	X(09)	032 - 040
10 IN4T-USRPAR-RECORD-TYPE	X(16)	041 - 056
05 IN4T-CITY	X(30)	057 - 086
05 IN4T-STATE-CODE	X(04)	087 - 090
05 IN4T-ZIP-CODE	X(09)	091 - 099
05 IN4T-COUNTRY-CODE	X(02)	100 - 101
05 IN4T-USER-PART-RCD-TYPE	X(04)	102 - 105

**NAMEAD4 FMS Transaction Definitions**

IN4T-ACTION-CODE (Required)

Description: *Transaction Action Code*

Valid Value(s):

A	=	Add
C	=	Change
D	=	Delete

**Partner/Qualifier Mode**

IN4T-PARTNER-ID (Required)

Description: *Partner ID*

IN4T-PARTNER-ID-QUAL (Required)

Description: *Partner ID Qualifier*

IN4T-TRANSACTION-TYPE (Required)

Description: *Transaction Type*

Valid Value(s):

Constant NAMEAD4 (left-justified and space-filled).

**Relationship Mode**

IN4T-USRPAR-USER-ID (Required)

Description: *User ID*

IN4T-USRPAR-PARTNER-ID (Required)

Description: *Partner ID*

IN4T-USRPAR-RECORD-TYPE (Required)

Description: *Transaction Type*

Valid Value(s):

Constant NAMEAD4 (left-justified and space-filled).

IN4T-CITY

Description: *City*. Enter the city of the partner's address.

IN4T-STATE-CODE

Description: *State Code*. Enter the standard abbreviation for the state or province of partner's address.

IN4T-ZIP-CODE

Description: *Zip Code*. Enter the partner's postal zip code.

IN4T-COUNTRY-CODE

Description: *Country Code*. Enter the standard abbreviation for the country of partner's address.

IN4T-USER-PART-RCD-TYPE (Required)

Description: *Record Type*

Valid Value(s):

USER PART

**NAMEAD5 Record***Description*

The NAMEAD1, NAMEAD2, NAMEAD3, NAMEAD4, and NAMEAD5 records process partner name and address information used in the Audit Print program.

*Record Layout*

Name	Data Type	Position
01 IN5T-INPUT-NAME-ADDR5-TRANSACTION.		
05 IN5T-ACTION-CODE	X(01)	001 - 001
05 IN5T-PARTNER-KEY.		
10 IN5T-PARTNER-ID	X(35)	002 - 036
10 IN5T-PARTNER-ID-QUAL	X(04)	037 - 040
10 IN5T-TRANSACTION-TYPE	X(16)	041 - 056
05 IN5T-USRPAR-KEY		
REDEFINES IN5T-PARTNER-KEY.		
10 IN5T-USRPAR-USER-ID	X(15)	002 - 016
10 IN5T-USRPAR-PARTNER-ID	X(15)	017 - 031
10 FILLER	X(09)	032 - 040
10 IN5T-USRPAR-RECORD-TYPE	X(16)	041 - 056
05 IN5T-CONTACT-PERSON	X(30)	057 - 086
05 IN5T-CONTACT-PHONE-NUMBER.		
05 IN5T-INTL-DIAL-CODE	X(03)	087 - 089
05 IN5T-AREA-CODE	X(03)	090 - 092
05 IN5T-PREFIX	X(03)	093 - 095
05 IN5T-SUFFIX	X(04)	096 - 099
05 IN5T-CONTACT-PHONE-EXTENSION	X(04)	100 - 103
05 IN5T-USER-PART-RCD-TYPE	X(04)	104 - 107

**NAMEAD5 FMS Transaction Definitions****IN5T-ACTION-CODE (Required)**

Description: *Transaction Action Code*

Valid Value(s):

A	=	Add
C	=	Change
D	=	Delete

**Partner/Qualifier Mode****IN5T-PARTNER-ID (Required)**

Description: *Partner ID*

**IN5T-PARTNER-ID-QUAL (Required)**

Description: *Partner ID Qualifier*

**IN5T-TRANSACTION-TYPE (Required)**

Description: *Transaction Type*

Valid Value(s):

Constant NAMEAD5 (left-justified and space-filled).

**Relationship Mode**

IN5T-USRPAR-USER-ID (Required)

Description: *User ID*

IN5T-USRPAR-PARTNER-ID (Required)

Description: *Partner ID*

IN5T-USRPAR-RECORD-TYPE (Required)

Description: *Transaction Type*

Valid Value(s):

Constant NAMEAD5 (left-justified and space-filled).

IN5T-CONTACT-PERSON

Description: *Contact Person*. Enter the name of EDI primary contact for the trading partner.

**Note:** This field supports the use of underscore substitution characters (USCs). To use a USC, the substitution character must exist on the Partner Header record.

See the *Gentran:Basic for zSeries Release 6.4 User's Guide* for more information about this feature.

IN5T-CONTACT-PHONE-NUMBER.

Description: *Contact Phone Number*. Enter the telephone number of primary contact person.

IN5T-INTL-DIAL-CODE

Description: *International Dial Code*. The international dial code for telephone numbers identifies the country of the partner. Enter the alphanumeric code.

IN5T-AREA-CODE

Description: *Area Code*. The area code for telephone identifies the region of the partner. Enter the alphanumeric code.

IN5T-PREFIX

Description: *Prefix*. The prefix identifies the public exchange of the partner. Enter a 3-digit alphanumeric code.

IN5T-SUFFIX

Description: *Suffix*. The suffix identifies a party line for the partner. Enter the applicable 4-digit alphanumeric code.

IN5T-CONTACT-PHONE-EXTENSION

Description: *Contact Phone Extension*. Enter the telephone extension of the primary contact person for the partner, if applicable.

IN5T-USER-PART-RCD-TYPE (Required)

Description: *Record Type*

Valid Value(s):

USER PART



## User Defined Data Records

### USER Record

#### Definition

This record is user-defined and is used for any purpose.

#### Record Layout

Name	Data Type	Position
01 IUT-INPUT-USER-DEF-TRANSACTION.		
05 IUT-ACTION-CODE	X (01)	001 - 001
05 IUT-PARTNER-KEY.		
10 IUT-PARTNER-ID	X (35)	002 - 036
10 IUT-PARTNER-ID-QUAL	X (04)	037 - 040
10 IUT-TRANSACTION-TYPE	X (16)	041 - 056
05 IUT-USRPAR-KEY		
REDEFINES IUT-PARTNER-KEY.		
10 IUT-USRPAR-USER-ID	X (15)	002 - 016
10 IUT-USRPAR-PARTNER-ID	X (15)	017 - 031
10 FILLER	X (09)	032 - 040
10 IUT-USRPAR-RECORD-TYPE	X (16)	041 - 056
05 IUT-POSITION	X (02)	057 - 058
05 IUT-USER-DEFINED-DATA-AREA	X (40)	059 - 098
05 IUT-RECORD-TYPE	X (04)	099 - 102

#### USER FMS Transaction Definitions

##### IUT-ACTION-CODE (Required)

Description: *Transaction Action Code*

Valid Value(s):

A	=	Add
C	=	Change
D	=	Delete

##### Partner/Qualifier Mode

##### IUT-PARTNER-ID (Required)

Description: *Partner ID*

##### IUT-PARTNER-ID-QUAL (Required)

Description: *Partner ID Qualifier*

##### IUT-TRANSACTION-TYPE

Description: *Transaction Type*

Default Value(s):

Constant USER

##### Relationship Mode

##### IUT-USRPAR-USER-ID (Required)

Description: *User ID*

## IUT-USRPAR-PARTNER-ID (Required)

Description: *Partner ID*

## IUT-USRPAR-RECORD-TYPE

Description: *Record Type*

Valid value(s):

Constant USER

## IUT-POSITION

Description: *Position*. This record is divided into seven 40-byte areas in which you can enter up to seven user records to define the User Partner record. Define the record as needed.

## IUT-USER-DEFINED-DATA-AREA

Description: *User Defined Data Area*. This field provides additional space to maintain relevant partner information. Enter information applicable to the partner's requirements.

**Note:** This field supports the use of underscore substitution characters (USCs). To use a USC, the substitution character must exist on the Partner Header record.

See the *Gentran:Basic for zSeries Release 6.4 User's Guide* for more information about this feature.

## IUT-RECORD-TYPE (Required)

Description: *Transaction Type*. This field allows for multiple user-defined areas for the partner profile (left justified). Identify which user-defined area needs to be updated.

## Data Split Record

### DATA SPLIT Record

#### Description

The DATA SPLIT record allows inbound or outbound files to be split and routed to numerous partners or locations (maximum of 10 files).

#### Record Layout

Name	Data Type	Position
01 IDST-PARTNER-DATA-SPLIT-TRANS.		
05 IDST-ACTION-CODE	X(01)	001 - 001
05 IDST-PARTNER-KEY.		
10 IDST-PARTNER-ID	X(35)	002 - 036
10 IDST-PARTNER-ID-QUAL	X(04)	037 - 040
10 IDST-RECORD-TYPE.		
15 IDST-RECORD-TYPE-LEVEL	X(01)	041 - 041
15 IDST-RECORD-TYPE-DIR	X(01)	042 - 042
15 IDST-RECORD-TYPE-ID	X(06)	043 - 048
15 IDST-RECORD-TYPE-REL-NUM	X(01)	049 - 049
10 FILLER	X(07)	050 - 057
05 IDST-USRPAR-KEY		
REDEFINES IDST-PARTNER-KEY.		
10 IDST-USRPAR-USER-ID	X(15)	002 - 016
10 IDST-USRPAR-PARTNER-ID	X(15)	017 - 031
10 FILLER	X(09)	032 - 040
10 IDST-USRPAR-RECORD-TYPE	X(16)	041 - 056
05 IDST-FILE-ID-TYPE	X(01)	057 - 057
05 IDST-FILE-ID-LIST-C		
10 IDST-FILE-ID-LIST-C-ENTRY		
OCCURS 10 TIMES		
15 IDST-FILE-ID-NUMBER-X	X(04)	058 - 061
15 IDST-FILE-ID-NUMBER	X(04)	058 - 061
REDEFINES IDST-FILE-ID-NUMBER-X.		
05 FILLER	X(60)	098 - 157
05 IDST-DESTINATION	X(10)	158 - 167
05 IDST-TEST-TRANSACTION	X(01)	168 - 168
05 IDST-VERSION	X(12)	169 - 180
05 IDST-RTE-OPTION	X(03)	181 - 183
05 IDST-RTE-PRIORITY	X(02)	184 - 184

**Note:** “!” is not valid for any field on this transaction.

**DATA SPLIT FMS Transaction Definitions**

IDST-ACTION-CODE (Required)

Description: *Transaction Action Code*

Valid Value(s):

A	=	Add
C	=	Change
D	=	Delete

**Partner/Qualifier Mode**

IDST-PARTNER-ID (Required)

Description: *Partner ID*

IDST-PARTNER-ID-QUAL (Required)

Description: *Partner ID Qualifier*

IDST-RECORD-TYPE-LEVEL

Description: *Record Type Level*. Identify the split level.

Valid Value(s):

P	=	Partner
G	=	Group
T	=	Transaction

IDST-RECORD-TYPE-DIR

Description: *Record Type Direction*. Identify the direction of the data being split.

Valid Value(s):

I	=	Inbound Application
O	=	Outbound Network (used only with Partner record type)

IDST-RECORD-TYPE-ID

Description: *Record Type ID*. Enter the specific ID to use to split the data. The value depends on the level field.

Valid Value(s):

If Level field	=	G: 2-character functional group code, plus 4 spaces
If Group code	=	Blank: 390 (processing message code)
If Level field	=	T: 3-character transaction code, plus 3 spaces
If Tran code	=	Blank: 390 (processing message code)
If Level field	=	P: Blank

**Note:** If field is not blank, the processing message code 00401 is displayed.

IDST-RECORD-TYPE-REL-NUM

Description: *Record Type Release Number*. This field is used for TRADACOMS partners. Enter the release number for TRADACOMS Transactions ID.**Relationship Mode**

IDST-USRPAR-USER-ID (Required)

Description: *User ID*

IDST-USRPAR-PARTNER-ID (Required)

Description: *Partner ID*

## IDST-USRPAR-RECORD-TYPE (Required)

Description: The following fields are used to build this field:

- IDST-RECORD-TYPE-LEVEL
- IDST-RECORD-TYPE-DIR
- IDST-RECORD-TYPE-ID
- IDST-RECORD-TYPE-REL-NUM

See the definitions for the previous Partner/Qualifier mode section for more information about these fields.

## IDST-FILE-ID-TYPE

Description: *File ID Type*. Identify the type of File ID.

Valid Value(s):

C = COBOLFD

## IDST-FILE-ID-NUMBER

**Note:** Partner direction can support up to 10 File IDs, with 000 to 099 per direction.

Description: *File ID Number*. Enter a list that identifies files to which data identified by this record are written. This field is repeated 10 times, and each field is 4 positions in length.

## IDST-DESTINATION

Description: *Destination*. Identify which outbound network or inbound application this document is internally linked to by Gentran:Basic databanking (left-justified and space-filled).

## IDST-TEST-TRANSACTION

Description: *Test Transaction*. Used for transaction splitting only. Use in conjunction with the flag setting for the ITMT-TEST-TRANSACTION field in the TRANMISC input transaction.

Valid Value(s):

T	=	Test splitting
P	=	Production splitting
blank	=	No test/production splitting

## IDST-VERSION

Description: *Version*. Associate the version/industry/association ID with this transaction or transaction by version (left-justified and space-filled).

## IDST-RTE-OPTION

Description: *Route Option*. Identify the option to route and process the data.

## IDST-RTE-PRIORITY

Description: *Route Priority*. Indicate the relative priority (order) to process this set of documents.

Valid Value(s):

1 – 9 alphanumeric

## Error Rejection Record

### ERROR REJECTION Record

#### Description

The ERROR REJECTION record is used by the inbound and outbound Editors when the error rejection features of Gentran:Basic are used.

#### Record Layout

Name	Data Type	Position
01 IERT-PARTNER-ERROR-REJECT-TRAN.		
05 IERT-ACTION-CODE	X(01)	001 - 001
05 IERT-PARTNER-KEY.		
10 IERT-PARTNER-ID	X(35)	002 - 036
10 IERT-PARTNER-ID-QUAL	X(04)	037 - 040
10 IERT-RECORD-TYPE.		
15 IERT-RECORD-ERROR-IND	X(01)	041 - 041
15 IERT-RECORD-TYPE-LEVEL	X(01)	042 - 042
15 IERT-RECORD-TYPE-ID	X(06)	043 - 048
15 IERT-ERROR-NUMBER-X	X(04)	049 - 052
15 IERT-ERROR-NUMBER	X(04)	049 - 052
REDEFINES IERT-ERROR-NUMBER-X.		
15 IERT-RECORD-TYPE-REL-NUM	X(01)	053 - 053
15 FILLER	X(03)	054 - 056
05 IERT-USRPAR-KEY		
REDEFINES IERT-PARTNER-KEY.		
10 IERT-USRPAR-USER-ID	X(15)	002 - 016
10 IERT-USRPAR-PARTNER-ID	X(15)	017 - 031
10 FILLER	X(09)	032 - 040
10 IERT-USRPAR-RECORD-TYPE	X(16)	041 - 056
05 IERT-INBOUND-REJECTION	X(01)	057 - 057
05 IERT-OUTBOUND-REJECTION	X(01)	058 - 058
05 IERT-VERSION	X(12)	059 - 070

#### ERROR REJECTION FMS Transaction Definitions

##### IERT-ACTION-CODE (Required)

Description: *Transaction Action Code*

Valid Value(s):

A	=	Add
C	=	Change
D	=	Delete

##### Partner/Qualifier Mode

##### IERT-PARTNER-ID (Required)

Description: *Partner ID*

##### IERT-PARTNER-ID-QUAL (Required)

Description: *Partner ID Qualifier*

## IERT-RECORD-ERROR-IND

Description: *Record Error Indicator*

Default Value(s):

Constant E

## IERT-RECORD-TYPE-LEVEL

Description: *Record Type Level*. Identify the error level.

Valid Value(s):

P	=	Partner
G	=	Group
T	=	Transaction

## IERT-RECORD-TYPE-ID

Description: *Record Type ID*. Indicate the Group and Transaction ID to use to reject the associated level. The value depends on level field.

Valid Value(s):

If Level field	=	G: Functional group code
If Level field	=	T: Transaction code
If Level code	=	P: Blanks

## IERT-ERROR-NUMBER

Description: *Error Number*. Enter the specific error code that causes the level to be rejected. If not numeric, error message code is 403.

## IERT-RECORD-TYPE-REL-NUM

Description: *Record Type Release Number*. Used for TRADACOMS partners. Enter the release number for the TRADACOMS Transactions ID.

**Relationship Mode**

## IERT-USRPAR-USER-ID (Required)

Description: *User ID*

## IERT-USRPAR-PARTNER-ID (Required)

Description: *Partner ID*

## IERT-USRPAR-RECORD-TYPE (Required)

Description: The following fields are used to build this field:

- IERT-RECORD-ERROR-IND
- IERT-RECORD-TYPE-LEVEL
- IERT-ERROR-NUMBER
- IERT-RECORD-TYPE-REL-NUM

See the definitions for the previous Partner/Qualifier mode section for more information about these fields.

## IERT-INBOUND-REJECTION

Description: *Inbound Rejection*. Indicate that this error is to be rejected for inbound activity.

Valid Value(s):

Y	=	Error is to be rejected for inbound activity
N	=	Error is <i>not</i> to be rejected for inbound activity

## IERT-OUTBOUND-REJECTION

Description: *Outbound Rejection*. Indicate that this error is to be rejected for outbound activity.

Valid Value(s):

Y	=	Error is to be rejected for outbound activity
N	=	Error is <i>not</i> to be rejected for outbound activity

## IERT-VERSION

Description: *Version*. Associate the version/industry/association ID with this error rejection or error rejection by version (left-justified and space-filled).



## CROSSREF Record

### Description

This record maintains the Gentran:Basic cross-reference information on the Cross Reference file.

### Record Layout

Name	Data Type	Position
01 CREF-PARTNER-XREF-TRAN.		
05 CREF-ACTION-CODE	X(01)	001 - 001
05 CREF-PARTNER-KEY.		
10 CREF-PARTNER-ID	X(35)	002 - 036
10 CREF-PARTNER-ID-QUAL	X(04)	037 - 040
10 CREF-RECORD-TYPE	X(16)	041 - 056
05 CREF-ALIAS-ID	X(35)	057 - 091
05 CREF-ALIAS-ID-QUAL	X(04)	092 - 095

### CROSSREF FMS Transaction Definitions

#### CREF-ACTION-CODE (Required)

Description: *Action Code*

Valid Value(s):

A	=	Add
C	=	Change
D	=	Delete

#### CREF-PARTNER-KEY

Description: *Partner Key*. This field is the first resolution partner key of cross-reference (left-justified and space-filled). All maintenance is performed by the value(s) specified in this key. Enter the partner Cross-reference ID and Qualifier. This cannot be an existing profile.

#### CREF-PARTNER-ID (Required)

Description: *Partner ID*

#### CREF-PARTNER-ID-QUAL (Required)

Description: *Partner ID Qualifier*

#### CREF-RECORD-TYPE (Required)

Description: *Transaction Type*

Valid Value(s):

Constant CROSSREF (left-justified and space-filled).

## CREF-ALIAS-ID

Description: *Alias ID*. This field is a second resolution partner alias for an existing partner profile (left-justified and space-filled). Enter the Partner ID found in the partner profile. This cannot maintain nonexistent alias keys.

**Note:** This field supports the use of underscore substitution characters (USCs). To use a USC, the substitution character must exist on the Partner Header record.

See the *Gentran:Basic for zSeries Release 6.4 User's Guide* for more information about this feature.

## CREF-ALIAS-ID-QUAL

Description: *Alias ID and Qualifier*. This field is a second resolution Qualifier alias for an existing partner profile (left-justified and space-filled). Enter the Qualifier found in the partner profile. This cannot maintain nonexistent alias keys.

## RELPART Record

### Description

The RELPART record defines the relationship between trading partner in the partner profile.

### Record Layout

Name	Data Type	Position
01 PRL-PARTNER-REL-TRAN.		
05 PRL-ACTION-CODE	X(01)	001 - 001
05 PRL-USRPAR-KEY.		
10 PRL-USRPAR-USER-ID	X(15)	002 - 016
10 PRL-USRPAR-PART-ID	X(15)	017 - 031
10 FILLER	X(09)	032 - 040
10 PRL-RECORD-TYPE	X(16)	041 - 056
05 PRL-RELATIONSHIP-KEY		
10 PRL-USER-KEY.		
15 PRL-USER-ID	X(35)	057 - 091
15 PRL-USER-ID-QUAL	X(04)	092 - 095
10 PRL-PARTNER-KEY.		
15 PRL-PARTNER-ID	X(35)	096 - 130
15 PRL-USER-ID-QUAL	X(04)	131 - 134
05 PRL-DIVISION	X(03)	135 - 138

### RELPART FMS Transaction Definitions

#### PRL-ACTION-CODE (Required)

Description: *Action Code*

Valid Value(s):

A	=	Add
C	=	Change
D	=	Delete

#### PRL-USRPAR-USER-ID (Required)

Description: *User ID*

#### PRL-USRPAR-PART-ID (Required)

Description: *Partner ID*

#### PRL-RECORD-TYPE (Required)

Description: *Transaction Type*

Valid Value(s):

Constant RELPART (left-justified and space-filled).

#### PRL-USER-ID (Required)

Description: *User ID*. Specify the user EDI ID portion of the user/trading partner relationship for a specific User/Partner ID in the partner profile (left-justified and space-filled).

**Note:** This field supports the use of underscore substitution characters (USCs). To use a USC, the substitution

character must exist on the Partner Header record.

See the *Gentran:Basic for zSeries Release 6.4 User's Guide* for more information about this feature.

PRL-USER-ID-QUAL (Required)

Description: *User ID Qualifier*. Specify the Qualifier associated with the User EDI ID, if applicable (left-justified and space-filled).

PRL-PARTNER-ID (Required)

Description: *Partner ID*. Enter the trading partner EDI ID portion of the user/trading partner relationship for a specific User/Partner ID in the partner profile (left-justified and space-filled).

**Note:** This field supports the use of underscore substitution characters (USCs). To use a USC, the substitution character must exist on the Partner Header record.

See the *Gentran:Basic for zSeries Release 6.4 User's Guide* for more information about this feature.

PRL-PARTNER-ID-QUAL (Required)

Description: *Partner ID Qualifier*. Enter the Qualifier associated with the Partner EDI ID, if applicable (left-justified and space-filled).

PRL-DIVISION

Description: *Division Code*. Enter a user-specified code identifying the division.

## Processing Reports

The Gentran:Basic Batch Partner Profile Maintenance Program produces these two reports:

- Transaction Totals report
- Errors Encountered Processing Partner Data report

These reports are illustrated and described below.

### Transaction Totals Report

This Transaction Totals report provides the total number of submitted, accepted, and rejected transactions.

This report also indicates whether any default records were added. Remember that Group and Transaction default records are added every time a new Partner Control record is added.

### Report Example

Figure 9.1 is an example of the Transaction Totals report.

```

*****
PROGRAM  EBDI005  COMPILED 03/28/0116.23.49
VERSION  6.4  GENTRAN: BASIC 12/01/2005
CURRENT DATE IS 12/01/2005
TIME STARTED IS 16:30:23
*****

PRINT ALL TRANSACTIONS
***** NORMAL END OF JOB *****
NUMBER OF INPUT RECORDS READ.....:00000017
NUMBER OF TRANSACTIONS ACCEPTED.....:00000017
NUMBER OF TRANSACTIONS IN ERROR.....:00000000
NUMBER OF PARTNER RECORDS ADDED.....:00000007
NUMBER OF DEFAULT GROUP RECORDS ADDED.....:00000001
NUMBER OF DEFAULT TRANS RECORDS ADDED.....:00000001
NUMBER OF PARTNER RECORDS DELETED.....:00000000
NUMBER OF ERROR MESSAGE RECORD CHANGES....:00000000
NUMBER OF RELATIONSHIP RECORDS ADDED.....:00000000
NUMBER OF RELATIONSHIP RECORDS DELETED....:00000000
NUMBER OF RELATIONSHIP RECORDS CHANGED....:00000000
*****

```

**Figure 9.1 Transaction Totals Processing Report**

### Errors Encountered Processing Partner Data Report

This report details any errors encountered while processing of the Batch Partner profile maintenance.

The Errors Encountered Processing Partner Data report lists the following information:

- All errors
- Error number
- Transaction field value in error
- Error message text
- Entire transaction in error

### Report Example

Figure 9.2 is an example of the Errors Encountered Processing Partner Data report.

EBDI005	RUN 12/01/2005	TIME 16:30:23	ERRORS ENCOUNTERED PROCESSING PARTNER DATA			PAGE 1
ERROR VERSION	NUMBER	INFORMATION	ERROR MESSAGE			
TRANSACTION ACCEPTED:	AQAUSER8	QAPARTNER8	PARTHDR	BATCH 005 PROGRAM ADDED PARTNER 12/01/2005		
TRANSACTION ACCEPTED:	AQAUSER8	QAPARTNER8	PARTMISC	UNB	NNNI	UTE
TRANSACTION ACCEPTED:	AQAUSER8	QAPARTNER8	PARTIN	D		000000234
TRANSACTION ACCEPTED:	AQAUSER8	QAPARTNER8	PARTOUT			F UNB
TRANSACTION ACCEPTED:	AQAUSER8	QAPARTNER8	GENUNB1	UNOA400000000000019		YYFO
TRANSACTION ACCEPTED:	AQAUSER8	QAPARTNER8	GENUNB2	UN MICKEY		ZZ REVERSEROUTEAD
TRANSACTION ACCEPTED:	AQAUSER8	QAPARTNER8	GENUNB3	UN DONALD		ZZ RECEIVERROUTEA
TRANSACTION ACCEPTED:	AQAUSER8	QAPARTNER8	GENUNB4	1 MICKEY		MINNIE
TRANSACTION ACCEPTED:	AQAUSER8	QAPARTNER8	GRPMISCORDERS	UNGUNHY	CONTRL00000000000001	
TRANSACTION ACCEPTED:	AQAUSER8	QAPARTNER8	GRPINORDERS	NDD 99B	111111111	
TRANSACTION ACCEPTED:	AQAUSER8	QAPARTNER8	GRPOUTORDERS	YDD 99B		
TRANSACTION ACCEPTED:	AQAUSER8	QAPARTNER8	GENUNG1ORDERS	33333333333333UN D	99BU	
TRANSACTION ACCEPTED:	AQAUSER8	QAPARTNER8	GENUNG2ORDERS	UNMICKEY		UNDONALD
TRANSACTION ACCEPTED:	AQAUSER8	QAPARTNER8	TRANMISCORDERS	UNHY	NTORDERS	
TRANSACTION ACCEPTED:	AQAUSER8	QAPARTNER8	TRANINORDERS	N		NN
TRANSACTION ACCEPTED:	AQAUSER8	QAPARTNER8	TRANOUTORDERS	Y222222222		NN
TRANSACTION ACCEPTED:	AQAUSER8	QAPARTNER8	GENUNHORDERS	2222222222222D	99BU	
* * * * * END OF REPORT * * * * *						

**Figure 9.2 Errors Encountered Processing Partner Data Report**

# Global Parameters

## Overview

This appendix describes an introduction to global parameters and each Gentran:Basic System-level Override and Runtime Only global parameter. All of the record layouts included in this appendix are arranged by processing option type; the Global Parameter Reference tables provide a basic overview.

This appendix contains the following topics:

<b>Topic</b>	<b>Page</b>
<b>Introduction to Global Parameters</b> .....	<b>A-2</b>
Editor Processing .....	A-2
Partner Databank Level.....	A-3
Partner Acknowledgments .....	A-3
Global Parameter Reference Tables .....	A-5
Inbound and Outbound (Common) Parameter Record Layouts .....	A-13
Inbound (Only) Parameter Record Layouts .....	A-25
Outbound (Only) Parameter Record Layouts .....	A-49

## Introduction to Global Parameters

Global parameters are defined at two different levels in Gentran: System-level and Runtime. The Runtime global parameters offer more parameter options than the System-level. However, all System-level global parameters are contained within Runtime.

System-level global parameters permit users to set base requirements through the online system. Base requirements are the values that users use most of the time. Some examples are the trading profile mode, databank levels, code checking, partner verification, and group Qualifiers.

Runtime global parameters permit users to set requirements at actual execution time through instream parameter cards. The advantage of this parameter is that each run can be set up differently.

The purpose of this appendix is to explain the functionality of the System-level and Runtime parameters, as they relate to the Inbound and Outbound Editor process.

### Editor Processing

Both the Inbound and Outbound Editors use the same methodology when processing the global parameters. As part of the initialization process, the Editors resolve all global parameter requirements.

#### *Global parameter processing sequence:*

System-level global parameters are loaded into working storage tables.

Runtime global parameters are read and processed as follows:

1. Parameters ending in the word 'OFF' instruct the Editors to update their internal table settings.
2. Parameters corresponding with the System-level global parameters will overlay the system-level settings.
3. The Runtime only parameters are loaded into working storage tables.

#### *Using the Partner Profile*

There are two global parameters, PARTNER DATABANK LEVEL and PARTNER ACKNOWLEDGEMENTS, which direct the Editors to use the partner profile in conjunction with the other global parameters (System-level and Runtime). The following section describes these parameters.



### Partner Databank Level

The Partner Databank Level global parameter instructs the Editors to use the partner profile to determine what databank level to use. On the partner profile, the EDI databank level can be set at the interchange (Control Information screen – EDIM015), group (Group Information screen – EDIM030), and transaction (Transaction Information screen – EDIM040) level for both inbound and outbound processing.

For inbound processing, the partner databank level processing options are directory (D) or none (N). If you want to select databank message store, you need to use a global parameter that specifies the 'FULL' option. For outbound processing, you can select databank level of 'FULL' at the interchange level of the partner profile.

To help avoid confusion, it is better to use either the partner profile or the global parameters to control your databank processing options, but not both. When both are used, the Editors will treat them at equal levels; that is, the partner profile does not override the global settings.

For example, if we look at the group level, the following results by the Editor will occur if both the partner profile and global parameters are used to set the group databank processing options:

#### Group Databank Processing Options

Partner Setting	Global Parameter	Editor Results
D	N	D
D	D	D
N	D	D
N	N	N

**Note:** If both partner and global options are used together, then they both must be set to none (N) to turn off the option.

### Partner Acknowledgments

The Partner Acknowledgments global parameter instructs the inbound Editor to look at the partner profile, in addition to any acknowledgment global parameters, when determining the type and level of acknowledgment to generate.

To instruct the Editor to use the partner profile when determining acknowledgments to generate, either set the partner PARTNER ACKNOWLEDGMENT flag to **Y** on the online Global Parameter Maintenance screen or place PARTNER ACKNOWLEDGEMENT in your runtime parameters.

When the partner profile and global acknowledge parameters are used in conjunction, the Inbound Editor will select the one that specifies the most detail level. The following chart, using interchange acknowledgment as an illustration, demonstrates the option selected by the Editor when both the partner (PARTNER ACKNOWLEDGEMENT) and global parameters (System-level and Runtime) are utilized.

**Acknowledge Interchange Options**

System-Level	Runtime	Partner Setting	Editor Result
Y		N	Y
Y		Y	Y
Y	OFF	N	N
Y	OFF	Y	Y
N		N	N
N		Y	Y
N	ON	N	Y
N	ON	Y	Y

**Note:** The same type of chart can be set up for Acknowledge Group, Acknowledge Transaction, and Acknowledge Errors.

The previous table is the decision tree used by the Editor to determine whether to generate a specific acknowledgment. Now, let us suppose that the most detailed level for acknowledgment generation at the global parameter level is group, ACKNOWLEDGE GROUP ON. If the partner profile specified acknowledgments at the transaction level, the Editor would generate group and transaction level acknowledgments for that trading partner. This type of set up is ideal if you have selected trading partners that want errors acknowledged.

### *System and Runtime Global Parameters*

Determining whether to set a global parameter at the System-level or Runtime is a business decision. However, you should note that not all System-level global parameters can be overridden at run time. To assist you in your decision, this appendix has been updated to identify those global parameters which cannot be overridden and those global parameters which can be substituted but not inactivated (e.g., PARTNER VERSION ALWAYS INTERCHANGE can be substituted with PARTNER VERSION INTERCHANGE at run time).

### *Configuration File*

When looking up interchange, group, and transaction partner profile records, you have the option of using version as part of the partner key. As of release 6.1., this option is controlled through the Configuration file rather than through global parameters. To set the parameter for using version in the partner key, select Record Type 0 (On-line options) from the online Configuration Directory.

If you select the Version option for the partner key, it will affect partner lookups in the Outbound Mapper and the Outbound EDI Data Extract program, and cannot be overridden. However, you can override the Version option for the partner key in the Inbound and Outbound Editors using these global parameters: GROUP VERSION OFF, TRANSACTION VERSION OFF, GROUP VERSION ERROR REJECTION OFF, TRANSACTION VERSION ERROR REJECTION OFF, GROUP VERSION SPLIT OFF, TRANSACTION VERSION SPLIT OFF INTERCHANGE VERSION SPLIT OFF, and the INTERCHANGE VERSION ERROR REJECTION OFF.

### Global Parameter Reference Tables

The following tables list global parameters by processing option and describe the situations in which each can be used. Check marks in the System-level and Runtime Only columns indicate availability; chicaneries in the Outbound and Inbound columns indicate whether the global parameter is used by the Inbound Editor, Outbound Editor, or both. A check mark in the Editor Default column indicates the initial value set by the Editors before applying any System-level or Runtime values.

To override the System-level settings on the Configuration file Global Parameter record, add a Runtime global parameter to your JCL (SYS030) with the desired setting using the record layout defined in the next section.

---

*General Processing Options*

Global Parameter	Editor Default	System Level	Runtime Only	Outbound	Inbound
ICS TABLES ON		✓		✓	✓
ICS TABLES OFF	✓	✓		✓	✓
CODE CHECK ON	✓	✓		✓	✓
CODE CHECK OFF		✓		✓	✓
OUTPUT MESSAGE ON	✓	✓			✓
OUTPUT MESSAGE OFF		✓			✓
ERROR REPORT ALWAYS		✓**		✓	✓
ALL OR NOTHING			✓	✓	✓
DETAIL REPORT ALWAYS			✓	✓	✓
USE SUBELEMENT SEPARATOR FOR ANSI			✓	✓	✓
USE SUBELEMENT SEPARATOR FOR XXXXXXXXXXXXXbXX			✓	✓	✓
GENERATE 999 FOR XXXXXXXXXXXXXbXX			✓		✓
GENERATE 997 FOR XXXXXXXXXXXXXbXX			✓		✓
OUTPUT DATABANK ONLY			✓	✓	
VERSION CONVERSION			✓	✓	✓
IC CONVERSION			✓	✓	✓

✓\* = When enabled at the system-level, the parameter may be changed at run time to the alternate paired option (e.g., PARTNER VERSION ALWAYS GROUP can be changed to PARTNER VERSION GROUP or vice versa), but cannot be inactivated.

✓\*\* = When enabled at the system-level, the parameter cannot be overridden at run time.

*Databank Processing Options*

Global Parameter	Editor Default	System Level	Runtime Only	Outbound	Inbound
PARTNER DATABANK LEVEL		✓		✓	✓
PARTNER DATABANK LEVEL OFF	✓	✓		✓	✓
INBOUND EDI DATABANK REPROCESS			✓		✓
INBOUND EDI INTERCHANGE DIRECTORY		✓			✓
INBOUND EDI INTERCHANGE FULL		✓			✓
INBOUND EDI INTERCHANGE NONE	✓	✓			✓
INBOUND EDI GROUP DIRECTORY		✓			✓
INBOUND EDI GROUP NONE	✓	✓			✓
INBOUND EDI TRANSACTION DIRECTORY		✓			✓
INBOUND EDI TRANSACTION NONE	✓	✓			✓
OUTBOUND EDI INTERCHANGE DIRECTORY		✓		✓	
OUTBOUND EDI INTERCHANGE FULL		✓		✓	
OUTBOUND EDI INTERCHANGE NONE		✓		✓	
OUTBOUND EDI GROUP DIRECTORY		✓		✓	
OUTBOUND EDI GROUP NONE		✓		✓	
OUTBOUND EDI TRANSACTION DIRECTORY		✓		✓	
OUTBOUND EDI TRANSACTION NONE		✓		✓	

✓\* = When enabled at the system-level, the parameter can be changed at run time to the alternate paired option (e.g., PARTNER VERSION ALWAYS GROUP can be changed to PARTNER VERSION GROUP or vice versa), but cannot be inactivated.

✓\*\* = When enabled at the system-level, the parameter cannot be overridden at run time.

*Partner Processing Options*

Global Parameter	Editor Default	System Level	Runtime Only	Outbound	Inbound
BG PARTNER YES		✓**			✓
PARTNER VERSION ALWAYS INTERCHANGE		✓*		✓	✓
PARTNER VERSION INTERCHANGE		✓*		✓	✓
PARTNER VERSION ALWAYS GROUP		✓*		✓	✓
PARTNER VERSION GROUP		✓*		✓	✓
PARTNER SEQUENCE ERROR – INCREMENTAL		✓			✓
PARTNER SEQUENCE ERROR – CHRONOLOGICAL		✓			✓
PARTNER SEQUENCE OFF		✓			✓
VERIFY PARTNER INTERCHANGE		✓		✓	✓
VERIFY PARTNER INTERCHANGE OFF	✓	✓		✓	✓
VERIFY PARTNER GROUP		✓		✓	✓
VERIFY PARTNER GROUP OFF	✓	✓		✓	✓
VERIFY PARTNER TRANSACTION		✓		✓	✓
VERIFY PARTNER TRANSACTION OFF	✓	✓		✓	✓

✓\* = When enabled at the system-level, the parameter may be changed at run time to the alternate paired option (e.g., PARTNER VERSION ALWAYS GROUP can be changed to PARTNER VERSION GROUP or vice versa), but cannot be inactivated.

✓\*\* = When enabled at the system-level, the parameter cannot be overridden at run time.

*Receiver Processing Options*

Global Parameter	Editor Default	System Level	Runtime Only	Outbound	Inbound
VERIFY RECEIVER INTERCHANGE		✓			✓
VERIFY RECEIVER INTERCHANGE OFF	✓	✓			✓
VERIFY RECEIVER GROUP		✓			✓
VERIFY RECEIVER GROUP OFF	✓	✓			✓

*Partner Lookup Options*

Global Parameter	Editor Default	System Level	Runtime Only	Outbound	Inbound
PARTNER XREF FIRST		✓			✓
NO PARTNER XREF		✓			✓
BG SENDER ID/COMM ID	✓	✓*			✓
BG COMM ID/SENDER ID		✓*			✓
ISA SENDER ID/AUTHORIZATION	✓	✓*			✓
ISA AUTHORIZATION/SENDER ID		✓*			✓
GS SENDER/RECEIVER QUALIFIER SPACES	✓	✓		✓	✓
GS SENDER QUALIFIER SPACES	✓	✓		✓	✓
GS RECEIVER QUALIFIER SPACES	✓	✓		✓	✓
GS SENDER/RECEIVER QUALIFIER INTERCHANGE		✓		✓	✓
GS SENDER QUALIFIER INTERCHANGE		✓		✓	✓
GS RECEIVER QUALIFIER INTERCHANGE		✓		✓	✓
GROUP VERSION OFF			✓		✓
TRANSACTION VERSION OFF			✓		✓

✓\* = When enabled at the system-level, the parameter may be changed at run time to the alternate paired option (e.g., PARTNER VERSION ALWAYS GROUP can be changed to PARTNER VERSION GROUP or vice versa), but cannot be inactivated.

*Rejection Processing Options*

<b>Global Parameter</b>	<b>Editor Default</b>	<b>System Level</b>	<b>Runtime Only</b>	<b>Outbound</b>	<b>Inbound</b>
PARTNER ERROR CONTROL		✓**		✓	✓
GROUP VERSION ERROR REJECTION OFF			✓	✓	✓
TRANSACTION VERSION ERROR REJECTION OFF			✓	✓	✓
INTERCHANGE VERSION ERROR REJECTION OFF			✓	✓	

*Data Separation Options*

<b>Global Parameter</b>	<b>Editor Default</b>	<b>System Level</b>	<b>Runtime Only</b>	<b>Outbound</b>	<b>Inbound</b>
DIRECTED OUTPUT FILES		✓		✓	✓
DIRECTED OUTPUT FILES OFF			✓		✓
SENDER ID FOR DATA SEPARATION		✓**			✓
TRANSACTION TEST/ PRODUCTION SPLIT		✓**			✓
GROUP VERSION SPLIT OFF			✓	✓	✓
TRANSACTION VERSION SPLIT OFF			✓	✓	✓
INTERCHANGE VERSION SPLIT OFF			✓	✓	

✓\* = When enabled at the system-level, the parameter may be changed at run time to the alternate paired option (e.g., PARTNER VERSION ALWAYS GROUP can be changed to PARTNER VERSION GROUP or vice versa), but cannot be inactivated.

✓\*\* = When enabled at the system-level, the parameter cannot be overridden at run time.



*Acknowledgment Options*

Global Parameter	Editor Default	System Level	Runtime Only	Outbound	Inbound
PARTNER ACKNOWLEDGEMENT		✓			✓
PARTNER ACKNOWLEDGEMENT OFF	✓	✓			✓
ACKNOWLEDGE INTERCHANGE ON		✓			✓
ACKNOWLEDGE INTERCHANGE OFF	✓	✓			✓
ACKNOWLEDGE GROUP ON		✓			✓
ACKNOWLEDGE GROUP OFF	✓	✓			✓
ACKNOWLEDGE TRANSACTION ON		✓			✓
ACKNOWLEDGE TRANSACTION OFF	✓	✓			✓
ACKNOWLEDGE ERRORS ON		✓			✓
ACKNOWLEDGE ERRORS OFF	✓	✓			✓
USE CNTL FOR ACKNOWLEDGEMENTS		✓**			✓
USE TCR FOR ACKNOWLEDGEMENTS		✓**			✓
GENERATE A2 ACCEPTANCE		✓**			✓

✓\* = When enabled at the system-level, the parameter may be changed at run time to the alternate paired option (e.g., PARTNER VERSION ALWAYS GROUP can be changed to PARTNER VERSION GROUP or vice versa), but cannot be inactivated.

✓\*\* = When enabled at the system-level, the parameter cannot be overridden at run time.

*EDIFACT/TRADACOMS Options*

Global Parameter	Editor Default	System Level	Runtime Only	Outbound	Inbound
DELETE OPTIONAL BLANK ELEMENTS		✓**		✓	
DELETE ALL BLANK ELEMENTS		✓**		✓	
USE STX QUALIFIER		✓**		✓	✓
PARTNER ERROR CONTROL – TRADACOMS		✓**			✓

*Outbound Wrapping Options*

Global Parameter	Editor Default	System Level	Runtime Only	Outbound	Inbound
COMPRESS TRANSMISSION (Interchange)	✓	✓*		✓	
COMPRESS GROUP		✓*		✓	
COMPRESS TRANSACTION		✓*		✓	
COMPRESS SEGMENT		✓*		✓	

*Add-On Product Options*

Global Parameter	Editor Default	System Level	Runtime Only	Outbound	Inbound
PLUS WILL ASSIGN BATCH #		✓**		✓	
BUSINESS DOCUMENT TRACKING		✓**			✓
TRACK OUTPUT FILES		✓**		✓	

✓\* = When enabled at the system-level, the parameter may be changed at run time to the alternate paired option (e.g., PARTNER VERSION ALWAYS GROUP can be changed to PARTNER VERSION GROUP or vice versa), but cannot be inactivated.

✓\*\* = When enabled at the system-level, the parameter cannot be overridden at run time.

## Inbound and Outbound (Common) Parameter Record Layouts

### *TABLES Record*

The tables record is used only if you wish to send or receive the new ASC X12 ICS/ICE interchange envelopes. There are two formats for the ICS segment. The older format has a control standards identifier which is four bytes in length. The valid values for the four byte version are AIAG and ANSI. With the newer format, this field is now only one byte long. The valid value is U which stands for USA EDI. This record in the Global Parameters file is used to disallow the old format as follows:

#### **ICS TABLES ON –**

This parameter tells the Editors to only accept the new format.

#### **ICS TABLES OFF –**

This parameter tells the Editors to allow either format. This is the default.

Position	Field Content	COBOL Picture	Format	Length	Description
001 – 014	ICS Tables	X(14)	AN	14	Parameter instructing edit Use program to use or not use the tables in the Table file to parse the ICS record. Valid values are constants: <ul style="list-style-type: none"> <li>ICS TABLES ON</li> <li>ICS TABLES OFF</li> </ul>
015 – 080	Filler	X(66)	AN	66	Not used

### *CODE VERIFICATION Record*

#### **CODE CHECK OFF –**

This parameter is used in the Inbound or Outbound Editor to turn off the loading of the codes from the Table file. It also turns off the logic for verifying codes within the Editor. Set up this in the program to save run time.

#### **CODE CHECK ON –**

This parameter is used in the Inbound or Outbound Editors to turn on the loading of the codes from the Table file. It also turns on the logic for verifying codes within the Editor. This is the default setting.

Position	Field Content	COBOL Picture	Format	Length	Description
001 – 014	Code Option	X(14)	AN	14	Indicates to the Editor that standard codes are not to be loaded nor validated. Valid values: <ul style="list-style-type: none"> <li>CODE CHECK OFF</li> <li>CODE CHECK ON</li> </ul>
015 – 080	Filler	X(66)	AN	66	Not used

**ERROR REPORT Record**

This record is used by both the Inbound and Outbound Editors. If ERROR REPORT ALWAYS is present, the Editors generate an error report even when no errors are found in the data. If no data errors are found, a summary report with total errors is generated. If this parameter is not active and an error occurs, a report is generated with a condition code of 4. Do not generate a report when no errors are found is the default.

See the topic “Administrative Main Menu Error Message Maintenance” for details on changing return codes, print flags, and error message text.

Position	Field Content	COBOL Picture	Format	Length	Description
001 – 020	Error Report Parameter	X(20)	AN	20	Indicates error report headings and totals should always be printed even if no errors are found. Valid value: ERROR REPORT ALWAYS
021 – 080	Filler	X(60)	AN	60	Not used

**ALL OR NOTHING**

This record specifies that updates to the databank can be limited to *clean* data only. When active, this Runtime global parameter monitors all interchanges. It sets all databank directory and message store information written for this run as interrupted. It allows passage of the data again after corrections have been made by the inbound trading partner or through the outbound application.

**Prefix:** None

Position	Field Content	COBOL Picture	Format	Length	Description
001 – 014	All or Nothing	X(14)	AN	14	Instructs the Editor to place clean data to databank message store and directory. Valid constant: ALL OR NOTHING
028 – 080	Filler	X(66)	AN	66	Not used

***DETAIL REPORT Record***

This is used during both inbound and outbound processing. Reserve this parameter for use in problem resolution because it can generate a large number of report pages. The report provides detailed information on:

- Databank Directory
- Error Rejection
- Directed Output
- Sequence Checking
- Partner Verification
- Partner Acknowledgments
- GS Qualifier Processing

<b>Position</b>	<b>Field Content</b>	<b>COBOL Picture</b>	<b>Format</b>	<b>Length</b>	<b>Description</b>
001 – 020	Detail Report Parameter	X(20)	AN	20	Indicates detailed information should always be printed on error report. Valid value: <b>DETAIL REPORT ALWAYS</b>
021 – 080	Filler	X(60)	AN	60	Not used

***SUBELEMENT SEPARATOR Record***

This record instructs the Editor to validate the subelement separator. The value for the subelement separator is extracted from ISA16 or ICS02. Starting in ANSI version 003030, composite/group elements were introduced; the subelement separator is now required. The subelement separator as a delimiter for variable EDI data cannot equal the element separator, segment, spaces, period or comma.

This parameter should be used for ANSI data version 003030 and higher. Multiple subelement separator records with different standard version identifiers or with the value ANSI can be used for all X12 standards. Specifying ANSI and a specific version in the same global parameter file is permitted, but not recommended.

**USE SUBELEMENT SEPARATOR FOR ANSI**

This instructs the Editor to validate and use the subelement separator (ISA16).

**USE SUBELEMENT SEPARATOR FOR XXXXXXXXXXXXXXXbXX**

This instructs the Editor to validate and use the subelement separator (ISA16) for version XXXXXXXXXXXXXXX.

**Prefix:** None

Position	Field Content	COBOL Picture	Format	Length	Description
001 – 028	Subelement Separator Parameter	X(28)	AN	28	Indicates to the Editor that the subelement separator on ISA16 should be edited and used for compliance for the specified version. Valid value: USE SUBELEMENT SEPARATOR FOR
029 – 029	Filler	X(01)	AN	01	Not used
030 – 041	Version	X(12)	AN	12	The version to edit or the value ANSI.
042 – 042	Filler	X(01)	AN	01	Not used
043 – 044	Agency Code	X(02)	AN	02	The controlling agency code associated with the indicated version. This is <u>not</u> required if the value ANSI is used for version in position 030.
045 – 080	Filler	X(36)	AN	36	Not used

#### VERSION CONVERSION Record

This record is used by both the Inbound and Outbound Editors to convert unusual representations of the standard version identifier to a valid version representation which Gentran:Basic understands. It does not allow you to change from a valid version identifier (according to Gentran:Basic) to another valid version. If you or your trading partners choose to use a non-standard version identifier, this record identifies it as an alias to a known valid version in the TABLES file. For example, if you receive a group which specifies version X1/2 and Agency B, the valid batch table version identifier 002001 and Agency X is used, if you have specified: VERSION X1/2 B 002001 X.

**Note:** The Version Conversion record does not allow you to change from one valid standard version identifier to another.

Position	Field Content	COBOL Picture	Format	Length	Description
001 – 010	Version Conversion Indicator	X(10)	AN	10	Constant VERSION
011 – 022	Input Version	X(12)	AN	12	Version from input data to be converted to a different version for edit table loading.
023 – 023	Filler	X(01)	AN	1	Not used; recommended to be used for visual alignment of the next parameter in the record.

Position	Field Content	COBOL Picture	Format	Length	Description
024 – 026	Input Version Agency Code	X(03)	AN	3	Responsible Agency code associated with Input Version
027 – 027	Filler	X(01)	AN	1	Not used; recommended to be used for visual alignment of the next parameter in the record.
028 – 039	Version to Use	X(12)	AN	12	Version to be used (instead of the one in Input Version above) when loading edit tables.
040 – 040	Filler	X(01)	AN	1	Not used; recommended to be used for visual alignment of the next parameter in the record.
041 – 043	Version to Use Agency Code	X(03)	AN	3	Responsible Agency code associated with version in which to use.
044 – 080	Filler	X(37)	AN	37	Not used

**PARTNER DATABANK Record**

To set the databank to the 'FULL' option, the inbound and/or outbound global parameters must be used. However, you can control whether or not use the directory at the partner level.

**PARTNER DATABANK LEVEL –**

This parameter indicates to the Inbound and Outbound Editor to look at the partner profile for databank directory levels. Otherwise, only the global parameters are to be used.

**Note:** The partner databank level is used in conjunction with other global parameters. The most detailed level is used. If either one has the value D specified, the global parameters (if present) are used.

**PARTNER DATABANK LEVEL OFF –**

This parameter is an override for the system-level setting on the Configuration file. The parameter indicates to the Inbound and Outbound Editor that PARTNER DATABANK LEVEL should be turned off and only the System-level (or Runtime, if present) global databank parameters are to be used for this run.

Position	Field Content	COBOL Picture	Format	Length	Description
001 – 026	Partner Databank ID	X(26)	AN	26	Indicates to the Editor that the partner is to be used to determine databank level of processing. Valid values: <ul style="list-style-type: none"> <li>PARTNER DATABANK LEVEL</li> <li>PARTNER DATABANK LEVEL OFF</li> </ul>
027 – 080	Filler	X(54)	AN	54	Not used

### *PARTNER VERSION Record*

This record is used for both inbound and outbound processing to override the version sent on the control envelopes and use the version defined in the partner profile instead.

#### **PARTNER VERSION ALWAYS INTERCHANGE –**

Compliance checking always uses the version on the partner control record instead of the version on the actual interchange record. In this case, the PARTNER VERSION USE field (A, I, D) is not considered. If no partner record exists, create an error message record with the data. Data is suspended if the version on the control envelope is not found. If no version is found on the partner control record, the version on the control envelope is used. An error message is issued to indicate that the version is not coded on partner.

#### **PARTNER VERSION INTERCHANGE –**

Compliance checking uses either the version on the partner control record or the version from the actual interchange record. Which version to use is based on the Partner Control Information VERSION USE field.

- If interchange version use is A, always use the partner version.
- If interchange version use is I, always use the Interchange record version.
- If interchange version use is D, use this version as the default if the input's identified version cannot be located in the standards table.

If the version is not found on the standards table, the entire interchange is suspended.

#### **PARTNER VERSION ALWAYS GROUP –**

Compliance checking always uses the version on the partner group information record instead of the version on the actual Group record for segments from GS to GE. In this case, the PARTNER VERSION USE field (A, I, D) is not considered. If no partner group or control record exists create an error message record with the data. Data is suspended if the version on the control envelope is not found.

#### **PARTNER VERSION GROUP –**

Compliance checking uses either the version on the partner group information record or the version of the actual Group record to use for editing segments from GS to GE. Which version to use is based on the Partner Group Information VERSION USE field. If no partner record is found, do not issue an error message. If the partner group or control record does not exist, data is suspended.

If you are running EDIFACT data and no UNG segment exists in the interchange, compliance checking uses the version on the partner group information record instead of the version on the UNH record for segments from UNH to UNT. The Partner Group Information VERSION USE field is not used if the UNG segment is missing from the interchange.



Position	Field Content	COBOL Picture	Format	Length	Description
001 – 016	ID	X(16)	AN	16	Constant PARTNER VERSION
017 – 034	Version Control	X(18)	AN	18	Indicates when partner profile is to be used to determine version to be used. ALWAYS option suspends if not found; otherwise, secondary sources are used. Valid values: <ul style="list-style-type: none"> <li>• ALWAYS INTERCHANGE</li> <li>• INTERCHANGE</li> <li>• ALWAYS GROUP</li> <li>• GROUP</li> </ul>
035 – 080	Filler	X(46)	AN	46	Not used

***PARTNER ERROR Record***

This record is used by both the Inbound and Outbound Editors. This tells Gentran:Basic that you are using the Error Rejection feature. The default is no error rejection.

**PARTNER ERROR CONTROL –**

This record indicates to the Editors that the Partner Error rejection feature is utilized. If this parameter is not active, error rejection does not occur.

Position	Field Content	COBOL Picture	Format	Length	Description
001 – 021	Partner Error Control Wanted	X(21)	AN	21	Indicates that the partner profile is to be used to determine if errors identified during the editing process are to be rejected or sent along for further processing. Valid value: PARTNER ERROR CONTROL.
022 – 080	Filler	X(59)	AN	59	Not used

**PARTNER ERROR CONTROL – TRADACOMS**

This record tells the Inbound Editor to use error rejection at the partner or interchange level. Any error within the interchange rejects the entire interchange. The default is no error rejection.

Position	Field Content	COBOL Picture	Format	Length	Description
001 – 033	Partner Error Control Wanted	X(33)	AN	33	Indicates that the partner profile is to be used to determine if errors identified during the editing process are to be rejected or sent along for further processing. Valid value: PARTNER ERROR CONTROL – TRADACOMS.
034 – 080	Filler	X(47)	AN	47	Not used

### *VERIFY Record*

This record is used by Gentran:Basic when receiving or sending data to specify the level of checking done against the partner profile. The default is for no verification to be done. Data that does not pass verification is suspended.

#### **VERIFY PARTNER INTERCHANGE –**

Interchange Sender (inbound) and Receiver (outbound) IDs are to be validated by checking the partner profile for their existence. A partner must exist in the partner profile (Control record).

#### **VERIFY PARTNER INTERCHANGE OFF –**

This parameters overrides the System-level global parameter setting on the Configuration file and turns off the verify interchange feature. No verification is done.

#### **VERIFY PARTNER GROUP –**

Group Sender (inbound) and Receiver (outbound) IDs with their respective Functional Group IDs are to be validated by checking the Partner Group Information records for their existence. A group information record for this partner must exist in the partner profile for this functional group ID.

#### **VERIFY PARTNER GROUP OFF –**

This parameters overrides the System-level global parameter setting on the Configuration file and turns off the verify group feature. No verification is done.

#### **VERIFY PARTNER TRANSACTION –**

Group Sender ID (inbound) or Receiver ID (outbound) with Transaction Set ID is validated by checking the Partner Transaction Information records for their existence. A transaction record for this partner must exist in the partner profile for this transaction set.

#### **VERIFY PARTNER TRANSACTION OFF –**

This parameters overrides the System-level global parameter setting on the Configuration file and turns off the verify transaction feature. No verification is done.

#### **VERIFY RECEIVER INTERCHANGE –**

Interchange Receiver IDs (inbound) are validated by checking the partner profile for their existence. The receiver must exist in the partner profile (Control record).

#### **VERIFY RECEIVER INTERCHANGE OFF –**

This parameter overrides the system-level global parameter setting on the Configuration file and turns off the receiver interchange feature. No verification is done.

**VERIFY RECEIVER GROUP –**

Group Receiver IDs are validated by checking the partner profile for their existence. The Group Receiver ID must exist in the partner profile (Control Record).

**VERIFY RECEIVER GROUP OFF –**

This parameter overrides the system-level global parameter setting on the Configuration file and turns off the verify receiver group feature. No verification is done.

<b>Position</b>	<b>Field Content</b>	<b>COBOL Picture</b>	<b>Format</b>	<b>Length</b>	<b>Description</b>
001 – 030	Verification Level	X(30)	AN	30	Indicates level at which partner profile is to be used to verify data. Valid values: <ul style="list-style-type: none"> <li>• VERIFY PARTNER INTERCHANGE</li> <li>• VERIFY PARTNER INTERCHANGE OFF</li> <li>• VERIFY PARTNER GROUP</li> <li>• VERIFY PARTNER GROUP OFF</li> <li>• VERIFY PARTNER TRANSACTION</li> <li>• VERIFY PARTNER TRANSACTION OFF</li> <li>• VERIFY RECEIVER INTERCHANGE</li> <li>• VERIFY RECEIVER INTERCHANGE OFF</li> <li>• VERIFY RECEIVER GROUP</li> <li>• VERIFY RECEIVER GROUP OFF</li> </ul>
031 – 080	Filler	X(50)	AN	50	Not used

***GS PARTNER Record***

This record is used by the Inbound (EBDI001) and Outbound (EBDI002) Editors to control what the Group Qualifier contains.

**GS SENDER/RECEIVER QUALIFIER SPACES –**

When performing group and transaction level partner lookups, the resolved Partner ID/Qualifier is set to spaces.

**GS SENDER QUALIFIER SPACES –**

When performing group and transaction level partner lookups, the resolved Partner ID/Qualifier is set to spaces (Replaces GS ASSUME = N in Gentran:Basic for MVS version 4.2 for the Sender ID only).

**GS RECEIVER QUALIFIER SPACES –**

When performing group and transaction level partner lookups, the resolved Partner ID/Qualifier is set to spaces (Replaces GS ASSUME = N in Gentran:Basic for MVS version 4.2 for the Receiver ID only).

**GS SENDER/RECEIVER QUALIFIER INTERCHANGE –**

When performing group and transaction level partner lookups, the resolved Partner ID/Qualifier is set to its respective Interchange Qualifier (Replaces GS ASSUME = ‘ or I in Gentran:Basic for MVS version 4.2 for the Sender and Receiver IDs only).

**GS SENDER QUALIFIER INTERCHANGE –**

When performing group and transaction level partner lookups, the resolved Partner ID/Qualifier is set to the Interchange Sender ID Qualifier value (Replaces GS ASSUME = ‘ or I in Gentran:Basic for MVS version 4.2 for the Sender ID only).

**GS RECEIVER QUALIFIER INTERCHANGE –**

When performing group and transaction level partner lookups, the resolved Partner ID/Qualifier is set to the Interchange Receiver ID Qualifier value (Replaces GS ASSUME = ‘ or I in Gentran:Basic for MVS version 4.2 for the Receiver ID only).

**GS RECEIVER QUALIFIER INTERCHANGE/UN –**

When performing group and transaction level partner lookups, the resolved Partner ID/Qualifier is set to the unresolved Interchange Receiver ID Qualifier value.

Position	Field Content	COBOL Picture	Format	Length	Description
001 – 040	GS Lookup	X(40)	AN	40	Indicates to the Editor the value to use for group partner Qualifier when reading the partner profile. Valid values: <ul style="list-style-type: none"> <li>GS SENDER/RECEIVER QUALIFIER SPACES</li> <li>GS SENDER QUALIFIER SPACES</li> <li>GS RECEIVER QUALIFIER SPACES</li> <li>GS SENDER/RECEIVER QUALIFIER INTERCHANGE</li> <li>GS SENDER QUALIFIER INTERCHANGE</li> <li>GS RECEIVE QUALIFIER INTERCHANGE</li> <li>GS RECEIVER QUALIFIER INTERCHANGE/UN</li> </ul>
041 – 080	Filler	X(40)	AN	40	Not used

***DIRECTED OUTPUT Record***

This record is used by both the Inbound and Outbound Editors. This tells Gentran:Basic that you are using the Split feature. The default is no splitting.

**DIRECTED OUTPUT FILES –**

This record instructs the Editors to use the Data Separation feature of the partner profile to determine which split file is to receive this document. There are 99 split files from which to choose.

For TRADACOMS, only the header message in the logical document is used to build the partner profile key when looking up Data Separation records.

This option must be used to direct outbound databank interchanges by outbound network identification. The Directed Output Files global parameter places the outbound partner's network identification on the Outbound EDI Databank Directory (EDIOEA).

This option is required if passing \$TCR (Transmission Control record) free-form information through the Outbound Editor.

Position	Field Content	COBOL Picture	Format	Length	Description
001 – 021	Directed Output Files On	X(21)	AN	21	Indicates that the Editor program is to direct the output to multiple files determined by the Partner file. Valid value: DIRECTED OUTPUT FILES
022 – 080	Filler	X(59)	AN	59	Not used

**DIRECTED OUTPUT FILES OFF**

Position	Field Content	COBOL Picture	Format	Length	Description
001 – 025	Directed Output Files OFF	X(25)	AN	25	Indicates that the Editor program is to direct the output to a single file and that no data separation is to be performed. Valid value: DIRECTED OUTPUT FILES OFF
026 – 080	Filler	X(55)	AN	55	Not used

**STX QUALIFIER Record****USE STX QUALIFIER –**

This parameter is used by the Outbound Editor for TRADACOMS only. It indicates whether or not the first two positions of the sender control number SNRF are to be used as a Qualifier. Otherwise, the Qualifier for the STX segment is set to spaces.

Position	Field Content	COBOL Picture	Format	Length	Description
001 – 017	Use STX Qualifier	X(17)	AN	17	Indicates whether or not the first two positions of the sender control number SNRF is to be used as a Qualifier. Otherwise, the Qualifier for the STX segment is spaces.
018 – 080	Filler	X(63)	AN	63	Not used

**IC CONVERSION Record**

Both the Inbound and Outbound Editors use this record to convert an implementation convention identifier to a valid version representation, which Gentran:Basic understands. Transaction sets within a standard are normally defined for general use by more than one business. Many businesses actually use a subset of that transaction set. In addition, some businesses like to have the Editor evaluate based on their subset definition instead of the generally defined standard definition.

To identify the subset, an implementation identifier is provided on the document along with the transaction set. In Gentran, the Standards file permits only one definition of a transaction set within any given Version ID. Therefore, to define a unique subset of a standard transaction set, it must be defined under a separate and unique Version ID in the Gentran Standards file. An implementation convention can contain from 1 to 35 characters. Also, the Version ID on the Standards file is restricted to a maximum of 12 characters. Therefore, the IC CONVERSION record provides a 12 characters alias name for the Version ID that the Editors can use.

Position	Field Content	COBOL Picture	Format	Length	Description
001 – 003	IC Conversion Indicator	X(03)	AN	3	Constant IC
004 – 038	Input IC identifier	X(35)	AN	35	IC identifier from the input transaction record that is to be converted to a Version ID for edit table loading.
039 – 039	Filler	X(01)	AN	1	Not used
040 – 051	IC Version to use	X(12)	AN	12	Version ID to use instead of the IC identifier
052 – 052	Filler	X(01)	AN	1	Not used

Position	Field Content	COBOL Picture	Format	Length	Description
053 – 055	Version to use agency code	X(03)	AN	3	Responsible Agency code associated with the version in which to use. This is usually the agency code from the original general Version ID used to create the IC.
056 – 080	Filler	X(25)	AN	25	Not used

### Inbound (Only) Parameter Record Layouts

#### *OUTPUT FILE Record*

**OUTPUT MESSAGE ON –**

This parameter tells the Editor to write error messages to the output file. This is the default.

**OUTPUT MESSAGE OFF –**

This parameter tells the Editor not to write error messages to the output file.

Position	Field Content	COBOL Picture	Format	Length	Description
001 – 018	Output Message Indicator	X(18)	AN	18	Indicates whether or not error messages generated to the Editor error report are also to be written to the output file Valid values: <ul style="list-style-type: none"> <li>• OUTPUT MESSAGE ON</li> <li>• OUTPUT MESSAGE OFF</li> </ul>
019 – 080	Filler	X(62)	AN	62	Not used

#### *ACKNOWLEDGE Record*

This record tells the Inbound Editor which (if any) acknowledgments you want to generate for incoming interchanges, groups and transactions. The default for each of the records is OFF. Multiple records are allowed.

**ACKNOWLEDGE INTERCHANGE ON –**

After a UNB interchange for EDIFACT is received, the Editor generates a UCI segment. After an ISA interchange is received, the Editor generates a TA1 segment. After an ICS interchange is received the Editor generates an AC1 segment. There is no interchange acknowledgment for the BG interchange, so none would be generated. This global parameter setting overrides both the interchange and partner setting.

**ACKNOWLEDGE INTERCHANGE OFF –**

This is a Runtime override parameter for the global parameter setting on the Configuration file.

**ACKNOWLEDGE GROUP ON –**

After a UNG group envelope for EDIFACT is received, the Editor generates a UCF segment. Each GS (group) received is acknowledged with either a 997 or a 999 and contains the AK1 and AK9 segment for an 997 document, or a B5 segment for a 999 document. The choice of a 997 or a 999

is controlled by another Global Parameter record (Generate Version Records) and the partner profile, as specified.

**ACKNOWLEDGE GROUP OFF –**

This is a runtime override parameter for the global parameter setting on the Configuration file.

**ACKNOWLEDGE TRANSACTION ON –**

After a UNH transaction envelope for EDIFACT is received, the Editor generates a UCM segment. Each ST (transaction) group received is acknowledged with either a 997 or a 999 and contains the AK2, AK5, or B5 segments. Additionally, each GS (group) received is acknowledged with an AK1 and AK9. The choice of a 997 or a 999 is controlled by another Global Parameter record (Generate Version Records) and the partner profile, as specified.

**ACKNOWLEDGE TRANSACTION OFF –**

This is a Runtime override parameter for the global parameter setting on the Configuration file.

**ACKNOWLEDGE ERRORS ON –**

For EDIFACT messages, if an error in the transaction is detected in a segment other than the envelope segment, the Editor generates a UCS and a UCD segment. The UCS segment identifies the position of the segment in error relative to the UNH segment and the UCD identifies the data element (component element) in error relative to the Segment ID. For ASC X12, if an error in the transaction is detected in a segment other than the envelope segment and a 997 type of acknowledgment is requested, the Editor generates an AK3 and an AK4 segment.

**ACKNOWLEDGE ERRORS OFF –**

This is a Runtime override parameter for the global parameter setting on the Configuration file.

Position	Field Content	COBOL Picture	Format	Length	Description
001 – 011	Acknowledge ID	X(11)	AN	11	Constant ACKNOWLEDGE
012 – 012	Filler	X(01)	AN	1	Blank
013 – 027	Acknowledge	X(15)	AN	15	<p><b>Constants</b> which Setting determine the level of acknowledgment generated. Valid values:</p> <ul style="list-style-type: none"> <li>• INTERCHANGE ON</li> <li>• INTERCHANGE OFF</li> <li>• GROUP ON</li> <li>• GROUP OFF</li> <li>• TRANSACTION ON</li> <li>• TRANSACTION OFF</li> <li>• ERRORS ON</li> <li>• ERRORS OFF</li> </ul>
028 – 080	Filler	X(53)	AN	53	Not used



**SEQUENCE CHECK Record**

The Inbound Editor looks for this record in the Global Parameters file. If the record is present, the Editor checks the last incoming interchange, group and transaction control numbers on the partner. If the last incoming interchange control number on the partner control is numeric and the current interchange control number value on the interchange being validated is not greater than this field value, an error message is created. The Inbound Editor then updates this field on the partner control. The same test is performed for the group control number. If the last incoming group control number on the partner group is numeric, and the current group control number value on the group being validated is not one greater than this field value, an error message is created. The Inbound Editor then updates the field on the partner group.

If you want to perform selective sequence checking (with specific partners only), leave the last incoming interchange/group/transaction control number field blank for those partners you do not want to check. The default is no sequence checking.

**PARTNER SEQUENCE ERROR – CHRONOLOGICAL –**

The Inbound Editor checks the previous control number received against the current control number at the interchange, group, and transaction level. If the current control number is not greater than the previous control number received, an error message is generated.

**Note:** Whether or not an error occurs, the partner record is updated with the new control number.

**PARTNER SEQUENCE ERROR – INCREMENTAL**

The Inbound Editor checks the previous control number received against the current control number at the interchange, group, and transaction level. If the current control number is not exactly one greater than the previous control number received, an error message is generated.

**Note:** Whether or not an error occurs, the partner record is updated with the new control number.

Position	Field Content	COBOL Picture	Format	Length	Description
001 – 025	Sequence Check Parameter	X(24)	AN	25	Constant PARTNER SEQUENCE ERROR
026 – 037	Sequence Check Type	X(13)	AN	12	Indicates that partner checks the sequence of incoming interchanges, groups, and transactions. You can specify the input document to be 1 greater than last one received or just greater (i.e., 1 or more) than previous. Valid values: <ul style="list-style-type: none"> <li>• INCREMENTAL</li> <li>• CHRONOLOGICAL</li> <li>• OFF</li> </ul>
028 – 080	Filler	X(43)	AN	43	Not used

***PARTNER CROSS REFERENCE Record***

This feature is used by the Inbound Editor.

**PARTNER XREF FIRST –**

A majority (or all) of partners are set up using a partner cross-reference. Therefore, when looking up Sender or Receiver IDs for an interchange, group, and transaction, always check Partner Cross-Reference first to obtain the correct Partner ID/Qualifier.

**NO PARTNER XREF –**

Partner Cross-Reference is not being used, all partners have been set up using their EDI IDs. Therefore, a second level look-up is not needed.

Position	Field Content	COBOL Picture	Format	Length	Description
001 – 018	XREF Option	X(18)	AN	18	Indicates to the Editor that standard cross-referencing is used all the time or never. Valid values: <ul style="list-style-type: none"> <li>PARTNER XREF FIRST</li> <li>NO PARTNER XREF</li> </ul>
019 – 080	Filler	X(62)	AN	62	Not used

***PROCESSING SEQUENCE Record***

The Inbound Editor has a predefined sequence of processing EDI documents to find the correct partner record. This predefined sequence is outlined below.

ISA    Sender ID  
       Partner XREF  
       Authorization Info.  
       Partner XREF

BG     Sender ID  
       Partner XREF  
       Commid  
       Partner XREF

ICS/UNB/GS Sender ID  
       Partner XREF

STX    Sender Code  
       Partner XREF

The number of partner lookups will vary depending on how your trading partners are defined. However, global parameters enable the user to tell the Editors how the majority of the partners were established and potentially reduce the number of passes the Editors must make to find the interchange, group, and transaction records. This should help improve processing speed by reducing the partner profile I/O.

### **BG SENDER ID/COMMID –**

All or a majority of trading partners that use the BG interchange are set up on the partner profile using SENDER ID.

### **BG COMMID/SENDER ID –**

All or a majority of trading partners that use the BG interchange are set up on the partner profile using COMMID.

### **ISA SENDER ID/AUTHORIZATION –**

All or a majority of trading partners that use the ISA interchange are set up on the partner profile using Sender ID.

If the authorization information Qualifier contains 01, the authorization information is used as a second lookup should the Sender ID not be found on the partner profile.

### **ISA AUTHORIZATION/SENDER ID –**

All or a majority of trading partners that use the ISA interchange and have an Authorization Information Qualifier of 01, are set up on the partner profile using Authorization Information.

**Note:** When the Authorization Information Qualifier is equal to 01, the Authorization Information element contains UCS COMMID.

The Inbound Editor has a predefined sequence of processing EDI documents to find the correct partner profile. For each interchange enveloped there are two difference approaches described. The first approach occurs if you are processing in the *Partner/Qualifier* mode; the second occurs if you are processing the *Relationship* Mode. Processing modes are set in the Configuration file at the time of installation.

Refer to the *Gentran:Basic for zSeries Release 6.4 User's Guide* for more information on processing modes.

The partner profile has a new Header record and the Control, Group, and Transaction keys now include an Interchange Header Option. The Interchange Header Option (ISA, BG, ICS, GS, UNB, and STX) permits multiple standards to be defined in one profile. You can have more than one Control record per partner. In addition, the version has been added to the key. This Version field, when used, contains the group compliance version and ties the Control record to its corresponding Group and Transaction records on the partner profile.

The predefined sequence for reading the partner profile is outlined in the following tables.

### **Header (AH) Record**

<b>ISA</b>	<b>Sender ID + Qualifier + Record Type</b> The key used to read the Partner Profile Header record.
	<b>Sender ID + Qualifier</b> If not found on the partner profile, use the same key to read the Partner Cross-Reference file.
	<b>Alias Sender ID + Qualifier</b> If found on the Partner Cross-Reference file, substitute the alias Sender ID and Qualifier for the ISA Sender ID and Qualifier and read the partner profile with this new key. If the read is not successful, continue.

**ISA Authorization + Qualifier + Record Type + Header Option**

The alternate key used to read the Partner Profile Header record. In this scenario, the Qualifier is always set to spaces.

**ISA Authorization + Qualifier**

If not found on the partner profile, use the same key to read the Partner Cross-Reference file.

**Alias Sender ID + Qualifier + Record Type + Header Option**

If found on the Partner Cross-Reference file, substitute the alias Sender ID and Qualifier for ISA Authorization and Qualifier. Read the partner profile with this new key.

**BG****Sender ID + Qualifier + Record Type + Header Option**

The key used to read the Partner Profile Header record. In this scenario, the Qualifier field is always spaces, and the Interchange Header Option contains the value BG.

**Sender ID + Qualifier**

If not found on the partner profile, use the same key to read the Partner Cross-Reference file.

**Alias Sender ID + Qualifier + Record Type + Header Option**

If found on the Partner Cross-Reference file, substitute the alias Sender ID and Qualifier for the BG Sender ID and Qualifier and read the partner profile with this new key. If the read is not successful, continue.

**BG COMMID + Qualifier + Record Type + Header Option**

The key used to read the Partner Profile Header record. In this scenario, the Qualifier value is always spaces.

**BG COMMID + Qualifier**

If the Partner Header record is not found, use the same key to read the Partner Cross-Reference file.

**Alias Sender ID + Qualifier + Record Type + Header Option**

If found on the Partner Cross-reference file, substitute the alias Sender ID and Qualifier for BG COMMID and Qualifier. Read the partner profile with this new key.

**ICS/GS****Sender ID + Qualifier + Record Type + Header Option**

The key used to read the Partner Profile Header record. In this scenario, the Qualifier field is always spaces and the Interchange Header Option contains either the value ICS or GS.

**Sender ID + Qualifier**

If the partner read is unsuccessful, use the same key to read the Partner Cross-reference file.

**Alias Sender ID + Qualifier + Record Type + Header Option**

If found on the Partner Cross-Reference file, substitute the alias Sender ID and Qualifier for ICS or GS Sender ID and Qualifier and read the partner profile with this new key.

**UNB****Sender ID + Qualifier + Record Type + Header Option**

The key used to read the Partner Profile Header record.

**Sender ID + Qualifier**

If not found on the partner profile, use the same key to read the Partner Cross-Reference file.

**Alias Sender ID + Qualifier + Record Type + Header Option**

If found on the Partner Cross-Reference file, substitute the alias Sender ID and Qualifier for the UNB Sender ID and Qualifier and read the partner profile with this new key.

**STX Sender ID + Qualifier + Record Type + Header Option**

The key used to read the Partner Profile Header record.

**Sender ID + Qualifier**

If not found on the partner profile, use the same key to read the Partner Cross-Reference file.

**Alias Sender ID + Qualifier + Record Type + Header Option**

If found on the Partner Cross-Reference file, substitute the alias Sender ID and Qualifier for UNB Sender ID and Qualifier and read the partner profile with this new key.

If the partner is still not found, the Editor checks the global parameters for an active VERIFY PARTNER INTERCHANGE. If this global parameter is active, suspend the interchange. Otherwise, a return code of 4 is issued and processing continues.

### *Control (CO) Record – ISA/BG/ICS/GS/UNB/STX*

**ISA Sender ID + Qualifier + Record Type + Header Option + GS Version**

The key used to read the Partner Control record if the Configuration file (Record Type 0) has the Interchange Version Flag set to 'Y.' In this scenario, the Version field contains the group compliance version, the Interchange Header Option contains ISA, and the record type contains 'CO.'

**Sender ID + Qualifier + Record Type + Header Option**

The key used to read the Partner Profile Control record, if the Configuration file (Record Type 0) has the Interchange Version Flag set to 'N' or the previous read with the version was not successful.

**Sender ID + Qualifier**

If the partner is not found, use the same key to read the Partner Cross-Reference file.

**Alias Sender ID + Qualifier + Record Type + Header Option + GS Version**

If found on the Partner Cross-Reference file, substitute the alias Sender ID and Qualifier for ISA Sender ID and Qualifier and read the partner profile with this new key. If the read is not successful, continue.

**Alias Sender ID + Qualifier + Record Type + Header Option**

If not found on the partner profile, remove the Version field, and try the partner again. If the read is not successful, continue.

**ISA Authorization + Qualifier + Record Type + Header Option + GS Version**

The key used to read the Partner Profile Control record if configuration has the Interchange Version Flag set to 'Y' and the previous reads had no hit. In this scenario, the Qualifier is set to spaces and the Version field contains the group compliance version.

**ISA Authorization + Qualifier + Record Type + Header Option**

The key used to read the Partner Profile Control record if the Configuration file (Record Type 0) has the Interchange Version Flag set to 'N' or the previous read was unsuccessful with GS Version. In this scenario, the Qualifier field is always set to spaces.

**ISA Authorization + Qualifier**

If the partner is not found, use the same key to read the Partner Cross-Reference file.

**Alias Sender ID + Qualifier + Record Type + Header Option + GS Version**

If found on the Partner Cross-Reference file, substitute the alias Sender ID and Qualifier for ISA Authorization and Qualifier. If configuration has the Interchange Version Flag set to 'Y' and the previous reads had no hit, read the partner profile with this new key. In this scenario, the Version field contains the group compliance version.

**Alias Sender ID + Qualifier + Record Type + Header Option**

If found on the Partner Cross-Reference file, substitute the alias Sender ID and Qualifier for ISA Authorization and Qualifier. If configuration has the Interchange Version Flag set to 'N' and the previous reads had no hit, read the partner profile with this new key.

If the partner is still not found, the Editor checks the global parameters for an active VERIFY PARTNER INTERCHANGE. If this global parameter is active, suspend the interchange. Otherwise, a return code of 4 is issued and processing continues.

**BG****Sender ID + Qualifier + Record Type + Header Option + GS Version**

The key used to read the Partner Profile Control record if the Configuration file (Record Type 0) has the Interchange Version Flag set to 'N' or the previous read with the version was not successful.

**Sender ID + Qualifier + Record Type + Header Option**

The key used to read the Partner Profile Control record, if the Configuration file (Record Type 0) has the Interchange Version Flag set to 'N' or the previous read with the version was not successful.

**Sender ID + Qualifier**

If the partner is not found, use same key to read the Partner Cross-Reference file.

**Alias Sender ID + Qualifier + Record Type + Header Option + GS Version**

If found on the Partner Cross-Reference field, substitute the alias Sender ID and Qualifier for BG Sender ID and Qualifier. If configuration has the Interchange Version Flag set to 'Y' and the previous reads had no hit, read the partner profile with this new key, If the read is not successful, continue.

**Alias Sender ID + Qualifier + Record Type + Header Option**

If found on the Partner Cross-Reference field, and the Configuration file (Record Type 0) has the Interchange Version Flag set to 'N' or the previous read was unsuccessful with the version, use this key to read the partner profile. If the read is not successful, continue.

**BG COMMID + Qualifier + Record Type + Header Option + GS Version**

The key used to read the partner profile Control record if the Configuration file (Record Type 0) has the Interchange Version Flag set to 'Y' and the previous reads had no hit. In this scenario, the Qualifier field contains spaces, the Interchange Header Option contains BG, and the Version field contains the group compliance version.

**BG COMMID + Qualifier + Record Type + Header Option**

The key used to read the partner profile Control record if the Configuration file (Record Type 0) has the Interchange Version Flag set to 'N' or the previous read was unsuccessful with Version.

**BG COMMID + Qualifier**

If the partner is not found, use the same key to read the Partner Cross-Reference file.

**Alias Sender ID + Qualifier + Record Type + Header Option + GS Version**

If found on the Partner Cross-Reference file, substitute the alias Sender ID and Qualifier for BG COMMID. Read the partner profile with this new key, if the Configuration file (Record Type 0) has the Interchange Version Flag set to 'Y' and the previous reads were unsuccessful.

**Alias Sender ID + Qualifier + Record Type + Header Option**

If found on the Partner Cross-Reference file, substitute the alias Sender ID and Qualifier for BG COMMID and Qualifier. Read the partner profile with this new key, if the Configuration file (Record Type 0) has the Interchange Version Flag set to 'N' or the previous read was unsuccessful.

If the partner is still not found, the Editor checks the global parameters for an active VERIFY PARTNER INTERCHANGE. If this global parameter is active, suspend the interchange. Otherwise, a return code of 4 is issued and processing continues.

**ICS/GS      Sender ID + Qualifier + Record Type + Header Option + GS Version**

The key used to read the Partner Profile Control record if the Configuration file (Record Type 0) has the Interchange Version Flag set to 'Y.' In this scenario, the Interchange Header Option contains ICS or GS, and Version contains the group compliance version.

**Sender ID + Qualifier + Record Type + Header Option**

The key used to read the Partner Profile Control record, if the Configuration file (Record Type 0) has the Interchange Version Flag set to 'N' or the previous read with the version was not successful.

**Sender ID + Qualifier**

If the partner is not found, use the same key to read the Partner Cross-Reference file.

**Alias Sender ID + Qualifier + Record Type + Header Option + GS Version**

If found on the Partner Cross-Reference file, substitute the alias Sender ID and Qualifier for ICS or GS Sender ID and Qualifier. Read the partner profile with this new key, if the Configuration file (Record Type 0) has the Interchange Version Flag set to 'Y' and the previous reads were unsuccessful.

**Alias Sender ID + Qualifier + Record Type + Header Option**

If found on the Partner Cross-Reference file, substitute the alias Sender ID and Qualifier for ICS or GS Sender ID and Qualifier. Read the partner profile with this new key, if the Configuration file (Record Type 0) has the Interchange Version Flag set to 'N' and the previous reads were unsuccessful.

If the partner is still not found, the Editor checks the global parameters for an active VERIFY PARTNER INTERCHANGE. If this global parameter is active, suspend the interchange. Otherwise, a return code of 4 is issued and processing continues.

**UNB      Sender ID + Qualifier + Record Type + Header Option + UNG (UNH) Version**

The key used to read the Partner Profile Control record if the Configuration file (Record Type 0) has the Interchange Version Flag set to 'Y.' In this scenario, the Interchange Header Option contains UNB and Version contains the group or transaction compliance version.

**Sender ID + Qualifier + Record Type + Header Option**

The key used to read the Partner Profile Control record, if the Configuration file (Record Type 0) has the Interchange Version Flag set to 'N' or the previous read with the UNH Version was not successful.

**Sender ID + Qualifier**

If the partner is not found, use same key to read the Partner Cross-Reference file.

**Alias Sender ID + Qualifier + Record Type + Header Option + UNG (UNH) Version**

If found on the Partner Cross-Reference file, substitute the alias Sender ID and Qualifier for UNB Sender ID and Qualifier. Read the partner profile with this new key, if the Configuration file (Record Type 0) has the Interchange Version Flag set to 'Y' and the previous read was unsuccessful.

**Alias Sender ID + Qualifier + Record Type + Header Option**

If found on the Partner Cross-Reference file, substitute the alias Sender ID and Qualifier for UNB Sender ID and Qualifier. Read the partner profile with this new key, if the Configuration file (Record Type 0) has the Interchange Version Flag set to 'N' and the previous read was unsuccessful.

If the partner is still not found, the Editor checks the global parameters for an active VERIFY PARTNER INTERCHANGE. If this global parameter is active, suspend the interchange. Otherwise, a return code of 4 is issued and processing continues.

**STX      Sender ID + Qualifier + Record Type + Header Option + STX Version**

The key used to read the Partner Profile Control record if the Configuration file (Record Type 0) has the Interchange Version Flag set to 'Y.' In this scenario, the Header Option contains the value STX, and Version field contains the STX compliance version.

**Sender ID + Qualifier + Record Type + Header Option**

The key used to read the Partner Profile Control record, if the Configuration file (Record Type 0) has the Interchange Version Flag set to 'N' or the previous read with the version was not successful.

**Sender ID + Qualifier**

If the partner is not found, use the same key to read the Partner Cross-Reference file.



### **Alias Sender ID + Qualifier + Record Type + Header Option + STX Version**

If found on the Partner Cross-Reference file, substitute the alias Sender ID and Qualifier for STX Sender ID and Qualifier. Read the partner profile with this new key, if the Configuration file (Record Type 0) has the Interchange Version Flag set to 'Y' and the previous reads were unsuccessful.

### **Alias Sender ID + Qualifier + Record Type + Header Option**

If found on the Partner Cross-Reference file, substitute the alias Sender ID and Qualifier for STX Sender ID and Qualifier. Read the partner profile with this new key. If the Configuration file (Record Type 0) has the Interchange Version Flag set to 'N' or the previous read with the version was unsuccessful.

If the partner is still not found, the Editor checks the global parameters for an active VERIFY PARTNER INTERCHANGE. If this global parameter is active, suspend the interchange. Otherwise, a return code of 4 is issued and processing continues.

## *Group (GP) Record – GS/UNG/BAT*

### **Sender ID + Qualifier + Record Type + Header Option + Version**

The newly constructed User/Partner key from above, plus the group (compliance) version, if the Configuration file (Record Type 0) has the Group Version Flag set to 'Y,' is used by the Editor to read the Partner Profile Group record. If the Partner Group Record is not found, the Editor removes the GS Version and tries again.

### **Sender ID + Qualifier + Record Type + Header Option**

The key used to read the Partner Profile Control record, if the Configuration file (Record Type 0) has the Interchange Version Flag set to 'N' or the previous read with the version was not successful.

### **Sender ID + Qualifier**

If the partner is not found, use the same key to read the Partner Cross-Reference file.

### **Alias Sender ID + Qualifier + Record Type + Header Option + Version**

If found on the Partner Cross-Reference file, substitute the alias Sender ID and Qualifier for Sender ID and Qualifier. If the Configuration file (Record Type 0) has Group Version Flag set to 'Y' and the previous read was unsuccessful, read the partner profile with this new key. If this read is not successful, continue.

### **Alias Sender ID + Qualifier + Record Type + Header Option**

If found on the Partner Cross-Reference file, substitute the alias Sender ID and Qualifier for Sender ID and Qualifier. If the Configuration file (Record Type 0) has Group Version Flag set to 'N' and the previous read was unsuccessful, read the partner profile with this new key.

If the partner is still not found, the Editor checks the global parameters for an active VERIFY PARTNER GROUP. If this global parameter is active, suspend the Group. Otherwise, use the !!!DFT Group record.

*Transaction (TC) Record – ST/UNH/MHD***Sender ID + Qualifier + Record Type + Header Record + Version**

The key used to read the Partner Profile Transaction record. In this scenario, the Version field contains the group compliance version, if the Configuration file (Record Type 0) has the Transaction Version Flag set to 'Y.' If the partner Transaction record is not found, the Editor removes the Version field and continues the process.

**Sender ID + Qualifier + Record Type + Header Option**

The key used to read the Partner Profile Transaction record, if the Configuration file (Record Type 0) has Transaction Version Flag set to 'N' or the previous read with the version was not successful.

If the partner is still not found, the Editor checks the global parameters for an active VERIFY PARTNER GROUP. If this global parameter is active, suspend the Group. Otherwise, use the !!!DFT Group record.

Key search sequence to determine a partner profile record when processing in Relationship mode requires two steps:

**Step 1: Build the User/Partner Key and Read the Partner Profile.****ISA****Sender ID + Qualifier + Receiver ID + Qualifier**

The Editor builds an EDI relationship key using the ISA Sender ID, Sender ID Qualifier, Receiver ID and Receiver ID Qualifier. The EDI relationship key is used by the Editor to read the Partner Relationship file to get the User/Partner key needed for the partner profile.

**Authorization ID + Receiver ID + Receiver ID Qualifier**

If the EDI relationship key is not found on the Partner Relationship file, the Editor substitutes the ISA Authorization for Sender ID, moves spaces to the Qualifier, and attempts another read against the Partner Relationship file.

**Sender ID (15-bytes) + Receiver ID (15 bytes)**

If the EDI Relationship record is not found on the Partner Relationship file, the Editor builds a User/Partner key by extracting the first 15 characters of the ISA Sender ID (Partner) and the first 15 characters of the ISA Receiver ID (User).

**BG****Sender ID + Receiver ID (set Qualifiers = spaces)**

The Editor builds an EDI relationship key using the BG Sender ID, Receiver ID, and spaces for Qualifiers. The EDI relationship key is used by the Editor to read the Partner Relationship file to get the User/Partner key needed for the partner profile.

**Comm ID + Receiver ID (set Qualifiers = spaces)**

If the EDI relationship key is not found on the Partner Relationship file, the Editor substitutes the BG Comm ID for Sender ID and attempts another read of the Partner Relationship file.

**Sender ID (15 bytes) + Receiver ID (15 bytes)**

If the EDI relationship key is not found on the Partner Relationship file, the Editor builds a User/Partner key by extracting the first 15 characters of the BG Sender ID (Partner) and the first 15 characters of the BG Receiver ID (User).

- ICS/GS**      **Sender ID + Qualifier + Receiver ID + Qualifier**  
The Editor builds an EDI relationship key using the Sender ID, Sender ID Qualifier, Receiver ID, and Receiver ID Qualifier. The EDI Relationship key is used by the Editor to read the Partner Relationship file to get the User/Partner key needed for the partner profile.
- Sender ID (15-bytes) + Receiver ID (15-bytes)**  
If the EDI Relationship record is not found on the Partner Relationship file, the Editor builds a User/Partner key by extracting the first 15 characters of the Sender ID (Partner) and the first 15 characters of the Receiver ID (User).
- UNB**            **Sender ID + Qualifier + Receiver ID + Qualifier**  
The Editor builds an EDI relationship key using the Sender ID, Sender ID Qualifier, Receiver ID, and Receiver ID Qualifier. The EDI relationship key is used by the Editor to read the Partner Relationship file to get the User/Partner key needed for the partner profile.
- Sender ID (15-bytes) + Receiver ID (15-bytes)**  
If the EDI Relationship record is not found on the Partner Relationship file, the Editor builds a User/Partner key by extracting the first 15 characters of the Sender ID (Partner) and the first 15 characters of the Receiver ID (User).
- STX**            **Sender ID (or Sender ID Qualifier) + STX Modify Qualifier + Receiver ID Qualifier**  
The Editor first checks STX Sender ID Qualifier for spaces. If true, the Editor uses the STX Sender ID as Partner ID. If false, the Editor uses the STX Sender ID Qualifier as Partner ID. Next the Editor checks to see if the global parameter, USE-STX-QUAL, is active. If it is active, it then checks to see if the STX Modify Qualifier contains a value. If true, the Editor uses the STX Modify Qualifier for the Partner ID/Qualifier. Otherwise, the Editor sets the Partner ID/Qualifier to spaces.
- After determining the partner portion of the EDI relationship key, the Editor finds the receiver portion of the EDI relationship key. The Editor checks the STX Receiver ID Qualifier to see if it contains spaces. If true, the Editor uses the STX Receiver ID as the Partner ID. Otherwise, the STX Receiver ID Qualifier is used as the Partner ID. In all cases, the Partner ID/Qualifier is set to spaces by the Editor.
- The Editor builds an EDI relationship key using the Sender Partner ID, Sender Partner ID Qualifier, Receiver Partner ID and Receiver Partner ID Qualifier determined above. The EDI relationship key is used by the Editor to read the Partner Relationship file to get the User/Partner key needed for the partner profile.
- Sender ID (15-bytes) + Receiver ID (15-bytes)**  
If the EDI relationship key is not found on the Partner Relationship file, the Editor builds a User/Partner key by extracting the first 15 characters of the Sender Partner ID (Partner) and the first 15 characters of the Receiver Partner ID (User).

**Step 2: Read the Partner Profile***Header (AH) Record***User ID + Partner ID + Record Type**

The newly constructed User/Partner key, from above, is used by the Editor to read the partner profile. If the partner is still not found, the Editor checks the global parameters for an active VERIFY PARTNER INTERCHANGE. If this global parameter is active, suspend the interchange. Otherwise, a return code of 4 is issued and processing continues.

*Control (CO) Record***User ID + Partner ID + Record Type + Header Option + Version**

The newly constructed User/Partner key, from above, plus the group (compliance) version, if the Configuration file (Record Type 0) has the Interchange Version Flag set to 'Y,' is used by the EDI to read the partner profile Control record. If the Partner Control record is not found, continue.

**User ID + Partner ID + Record Type + Header Option**

The new constructed User/Partner key, from above, is used by the Editor to read the Partner Control record when the Configuration file (Record Type 0) has the Interchange Version Flag set to 'N' or the initial read above failed. If the partner is still not found, the Editor checks the global parameters for an active VERIFY PARTNER INTERCHANGE. If this global parameter is active, suspend the interchange. Otherwise, a return code of 4 is issued and processing continues.

*Group (GP) Record – GS/UNG/BAT***User ID + Partner ID + Record Type + Header Option + VERSION**

The newly constructed User/Partner key, from above, plus the group (compliance) version, if the Configuration file (Record Type 0) has the Group Version Flag set to 'Y,' is used by the Editor to read the Partner Profile Group record. If the partner Group record is not found, continue to next option.

**User ID + Partner ID + Record Type + Header Option**

The newly constructed User/Partner key, from above, is used by the Editor to read the Partner Profile Group record when the Configuration file (Record Type 0) has the Group Version Flag set to 'N' or the initial read above fails. If the partner is still not found, the Editor checks the global parameters for an active VERIFY PARTNER TRANSACTION. If this global parameter is active, suspend the Group. Otherwise, the !!!DFT Group record is issued and processing continues.

*Transaction (TC) Record – ST/UNH/MHD***User ID + Partner ID + Record Type + Header Option + VERSION**

The newly constructed User/Partner key, from above, plus the group (compliance) version, if the Configuration file (Record Type 0) has the Transaction Version Flag set to 'Y,' is used by the Editor to read the Partner Profile Transaction record. If the partner Transaction record is not found, the Editor removes the GS Version and tries again.

### User ID + Partner ID + Record Type + Header Option

The newly constructed User/Partner key, from above, is used by the Editor to read the Partner Profile Transaction record when the Configuration file (Record Type 0) has the Transaction Version Flag set to 'N' or the initial read above fails. If the partner is still not found, the Editor checks the global parameters for an active VERIFY PARTNER TRANSACTION. If this global parameter is active, suspend the Transaction. Otherwise, the !!!DFT transaction record is issued and processing continues.

Position	Field Content	COBOL Picture	Format	Length	Description
001 – 027	Process Option	X(27)	AN	27	Indicates to the Editor the sequence to process GS and ISA interchanges. Valid values: <ul style="list-style-type: none"><li>• BG SENDER IF/COMMID</li><li>• BG COMMID/SENDER ID</li><li>• ISA SENDER ID/ AUTHORIZATION</li><li>• ISA AUTHORIZATION/ SENDER ID</li></ul>
028 – 080	Filler	X(53)	AN	53	Not used

**DATABANK Record****INBOUND EDI INTERCHANGE FULL –**

For each interchange, the Inbound Editor generates both directory (audit) and message store entries for the databank.

**INBOUND EDI INTERCHANGE DIRECTORY –**

For each interchange, the Inbound Editor generates directory (audit) entries for the databank.

**INBOUND EDI INTERCHANGE NONE –**

For each interchange, the Inbound Editor does not use the databank feature.

**INBOUND EDI GROUP DIRECTORY –**

At the group level, the Inbound Editor generates directory (Audit) only entries for the databank.

**INBOUND EDI GROUP NONE –**

At the group level, the Inbound Editor does not use the databank feature.

**INBOUND EDI TRANSACTION DIRECTORY –**

For each transaction, the Inbound Editor generates directory (Audit) only entries for the databank.

**INBOUND EDI TRANSACTION NONE –**

For each transaction, the Inbound Editor does not use the databank feature.

**INBOUND EDI DATABANK REPROCESS –**

After selecting the interchange to be reprocessed through the online databank, use this parameter to instruct the Editor to pick them up for reprocessing. If parameter is not active, the Editor does not pick up any databank interchanges.

Position	Field Content	COBOL Picture	Format	Length	Description
001 – 011	Databank Indicator	X(11)	AN	11	Constant INBOUND EDI
012 – 012	Filler	X(01)	AN	01	Constant: space
013 – 033	Databank Option	X(21)	AN	21	Indicates to the Editor the level of processing to get done for databank. Valid values: <ul style="list-style-type: none"> <li>• DATABANK REPROCESS</li> <li>• INTERCHANGE FULL</li> <li>• INTERCHANGE DIRECTORY</li> <li>• INTERCHANGE NONE</li> <li>• GROUP DIRECTORY</li> <li>• GROUP NONE</li> <li>• TRANSACTION DIRECTORY</li> <li>• TRANSACTION NONE</li> </ul>
034 – 080	Filler	X(47)	AN	47	Not used

*GENERATE A2 Record*

**GENERATE A2 ACCEPTANCE –**

This parameter instructs the Inbound Editor to generate an A2 segment for each accepted transaction. If a transaction is rejected, an A1 segment is generated (if acknowledge errors is active).

Position	Field Content	COBOL Picture	Format	Length	Description
001-022	Generate A2	X(22)	AN	22	Indicates to the Editor that when 999 type of acknowledgments are generated. Also indicates to the Editor to generate A2 acceptance records. Valid value: GENERATE A2 ACCEPTANCE
023-080	Filler	X(58)	AN	58	Not used

*TCR ACKNOWLEDGE Record*

Transmission Control Records are pre-defined. Each outbound partner must have a defined \$TCR user record in the partner profile. The Inbound Editor generates an outbound \$TCR for each acknowledgment interchange. A third party communication package uses the \$TCR information to load and transmit acknowledgments.

See Chapter 2 in the *Gentran:Basic for zSeries Release 6.4 User's Guide* for additional information about Record Maintenance.

Position	Field Content	COBOL Picture	Format	Length	Description
001-031	Use TCR for acknowledgments	X(31)	AN	31	Instructs the Editor to generate \$TCR user-defined information. Valid value: USE TCR FOR ACKNOWLEDGEMENTS
032-080	Filler	X(49)	AN	49	Not used

**CNTL ACKNOWLEDGE Record****USE CNTL FOR ACKNOWLEDGEMENTS –**

Parameter indicates to the Editor that CNTL records are to be used when building acknowledgments. The acknowledgments must go through the envelope generation process before being processed by the Outbound Editor. Use this parameter with program EBDI011A. With this option, the partner file must be set up to handle outbound acknowledgments because EBDI011A relies on the partner file to build the envelopes.

Position	Field Content	COBOL Picture	Format	Length	Description
001-028	Generate CNTL Records	X(28)	AN	28	Indicates to the Editor that acknowledgments being generated are to contain CNTL control records instead of envelopes. Valid value: USE CNTL FOR ACKNOWLEDGMENTS
029-080	Filler	X(52)	AN	52	Not used

**SPLIT BY SENDER Record****SENDER ID FOR DATA SEPARATION –**

Data separation is normally based on the Receiver ID. This parameter instructs the Inbound Editor to use the Sender ID instead.

Position	Field Content	COBOL Picture	Format	Length	Description
001-029	Split by Sender	X(29)	AN	29	Indicates to the Editor that Sender ID and Qualifier are to be used when reading the partner Data Separation (Split) record. Valid value: SENDER ID FOR DATA SEPARATION
030-080	Filler	X(51)	AN	51	Not used



**INTERCHANGE VERSION Record**

This parameter is used by the Outbound Editor to indicate whether the optional Version field key should be used when reading the partner profile.

**Note:** The INTERCHANGE VERSION feature is activated through the online Configuration file, not through the online Global Parameter file or runtime parameters. If the Interchange Version field on the Record Type 0 screen of the Configuration file is set to a 'Y,' the parameters INTERCHANGE, INTERCHANGE VERSION SPLIT and INTERCHANGE VERSION ERROR REJECTION will be activated. If these parameters are activated, the summary report generated by the Outbound Editor will indicate this. Please review the Configuration Maintenance section of your *Gentran:Basic for zSeries Release 6.4 User's Guide* for more information on updating the Record Type 0 screen of the Configuration file.

**INTERCHANGE VERSION SPLIT OFF-**

This parameter indicates to the Editor that the partner data separation (split) record was not set up using the optional Version field. When this parameter is not active, the Editor does not use the version from interchange compliance checking as part of the key when looking up the interchange level data separation record. Please note that the version to be used by the Editor may be different from the version on the interchange envelope if the global parameters PARTNER VERSION (ALWAYS) INTERCHANGE or VERSION conversion is utilized.

**INTERCHANGE VERSION ERROR REJECTION OFF-**

This parameter indicates to the Editor that the partner error rejection record was not set up by using the optional Version field. When this parameter is not active, the Editor does not use the version from interchange compliance checking as part of the key when looking up the interchange level error rejection record. Please note that the version to be used by the Editor may be different from the version on the interchange envelope if the global parameters PARTNER VERSION (ALWAYS) INTERCHANGE or VERSION conversion is utilized.

Position	Field Content	COBOL Picture	Format	Length	Description
001-033	Interchange Version Option	X(33)	AN	33	Indicates to the Editor that the optional Version field has not been utilized. Valid values: <ul style="list-style-type: none"> <li>• INTERCHANGE OFF</li> <li>• INTERCHANGE VERSION SPLIT OFF</li> <li>• INTERCHANGE VERSION ERROR REJECTION OFF</li> </ul>
034-080	Filler	X(47)	AN	47	Not used

### *GROUP VERSION Record*

This parameter is used by the Inbound and Outbound Editors to indicate whether the optional Version field key should be used when reading the partner profile.

**Note:** The GROUP VERSION feature is activated through the online Configuration file, not through the online Global Parameter file or runtime parameters. If the Group Version field on the Record Type 0 screen of the Configuration file is set to a 'Y,' the parameters GROUP VERSION, GROUP VERSION SPLIT and GROUP VERSION ERROR REJECTION will be activated. If these parameters are activated, the summary report generated by the Inbound and Outbound Editors will indicate this. Please review the Configuration Maintenance section of your *Gentran:Basic for zSeries Release 6.4 User's Guide* for more information on updating the Record Type 0 screen of the Configuration file.

#### **GROUP VERSION OFF –**

This parameter indicates to the Inbound Editor that version should not be used for looking up the group information record on the partner profile. If the Group record was not created using the optional Version field, use this parameter. If the Group record is not found, the Editor checks to see if global parameter VERIFY PARTNER GROUP is active. If it is active, the group is suspended.

#### **GROUP VERSION SPLIT OFF-**

This parameter indicates to the Editor that the partner data separation (split) record was not set up using the optional Version field. When this parameter is not active, the Editor does not use the version from group compliance checking as part of the key when looking up the group level data separation record. Please note that the version to be used by the Editor may be different from the version on the group envelope if the global parameters PARTNER VERSION (ALWAYS) GROUP or VERSION conversion is utilized.

#### **GROUP VERSION ERROR REJECTION OFF-**

This parameter indicates to the Editor that the partner error rejection record was not set up by using the optional Version field. When this parameter is not active, the Editor does not use the version from group compliance checking as part of the key when looking up the group level error rejection record. Please note that the version to be used by the Editor may be different from the version on the group envelope if the global parameters PARTNER VERSION (ALWAYS) GROUP or VERSION conversion is utilized.

Position	Field Content	COBOL Picture	Format	Length	Description
001-033	Group Version Option	X(33)	AN	33	Indicates to the Editor that the optional Version field has not been utilized. Valid values: <ul style="list-style-type: none"> <li>GROUP VERSION OFF</li> <li>GROUP VERSION SPLIT OFF</li> <li>GROUP VERSION ERROR REJECTION OFF</li> </ul>
034-080	Filler	X(47)	AN	47	Not used

**TRANSACTION VERSION Record**

This parameter is used by the Inbound and Outbound Editors to indicate whether the optional Version field key should be used when reading the partner profile.

**Note:** The TRANSACTION VERSION feature is activated through the online Configuration file, not through the online Global Parameter file or runtime parameters. If the Transaction Version field on the Record Type 0 screen of the Configuration file is set to 'Y,' the parameters TRANSACTION VERSION, TRANSACTION VERSION SPLIT and TRANSACTION VERSION ERROR REJECTION will be activated. If these parameters are activated, the summary report generated by the Inbound and Outbound Editors will indicate this.

**TRANSACTION VERSION OFF-**

This parameter indicates to the Inbound Editor that version should not be used when looking up the transaction information record on the partner profile. If the transaction record was not created using the optional Version field, use this parameter. If the transaction record is not found, the Editor checks to see whether the global parameter VERIFY PARTNER TRANSACTION is active. If it is, the transaction is suspended.

**TRANSACTION VERSION SPLIT OFF –**

This parameter indicates to the Editor that the partner data separation (split) record was not set up using the optional Version field. When this parameter is not active, the Editor does not use the version from group compliance checking as part of the key when looking up the transaction level data separation record. Please note that the version on the group envelope if the global parameters PARTNER VERSION (ALWAYS) GROUP or VERSION conversion is utilized.

**TRANSACTION VERSION ERROR REJECTION OFF-**

This parameter indicates to the Editor that the partner error rejection record was not set up by using the optional Version field. When this parameter is not active, the Editor does not use the version from group compliance checking as part of the key when looking up the transaction level error rejection record. The version to be used by the Editor may be different from the version on the group envelope if the global parameters PARTNER VERSION (ALWAYS) GROUP or VERSION conversion is used.

Position	Field Content	COBOL Picture	Format	Length	Description
001-039	Transaction Version Option	X(39)	AN	39	Indicates that the optional Version field has not been used for transaction level processing. Valid values: <ul style="list-style-type: none"> <li>TRANSACTION VERSION OFF</li> <li>TRANSACTION VERSION SPLIT OFF</li> <li>TRANSACTION VERSION ERROR REJECTION OFF</li> </ul>
040-080	Filler	X(41)	AN	41	Not used

**TRANSACTION TEST/PRODUCTION Record**

This parameter is used by the Inbound Editor to identify groups and transactions in test/production sessions.

**TRANSACTION TEST/PRODUCTION SPLIT –**

This parameter indicates to the Editor that the test/production flag from the transaction record on the partner profile is to be used in the partner data separation (split) lookup.

**Note:** This parameter only uses the transaction record on the partner profile to determine test or production status and does not consider the test or production status found on the envelopes of the input documents.

The test/production flag on the partner data separation record is an optional field that may be used in conjunction with the global parameters TRANSACTION VERSION SPLIT and TRANSACTION VERSION. The number of lookups performed by the Editor is based on the various transaction globals activated. The Editor starts with all options selected and proceeds until either it finds the data separation record or all combinations (including no Test/Production and no version) have been completed.

Position	Field Content	COBOL Picture	Format	Length	Description
001-033	Split by Test/ Prod	X(33)	AN	33	Indicates that splitting is to be done using the test/production flag from the transaction partner record. Valid value: TRANSACTION TEST/ PRODUCTION SPLIT
034-080	Filler	X(47)	AN	47	Not used

### *PARTNER ACKNOWLEDGE Record*

#### **PARTNER ACKNOWLEDGEMENT –**

This parameter is used by the Inbound Editor (EBDI001) to instruct Gentran:Basic to look at the partner profile for information on acknowledgment generation. The Global Parameters Acknowledgments records are looked at in conjunction with the partner profile if the partner acknowledge record is present. If the partner acknowledge record is omitted, only the Global Parameter Acknowledge records are used to determine the types of acknowledgments generated. The level of acknowledgment generated is present at the most detailed level in either of these files. For example, if the partner profile says acknowledge groups and the Global Parameters file only has an acknowledge interchange record, groups is acknowledged along with interchanges. This is useful if you want more detailed acknowledgments generated for selected trading partners.

#### **PARTNER ACKNOWLEDGEMENT OFF –**

This runtime global parameter overrides the Configuration Global Parameter record value and turns it off for this run.

Position	Field Content	COBOL Picture	Format	Length	Description
001-027	Partner Acknowledge Request	X(27)	AN	27	Indicates if the partner profile is to be used to determine the generation of acknowledgments. Valid values: <ul style="list-style-type: none"><li>• PARTNER ACKNOWLEDGEMENT</li><li>• PARTNER ACKNOWLEDGEMENT OFF</li></ul>
028-080	Filler	X(53)	AN	53	Not used

### *BG PARTNER Record*

This record is used by the Inbound Editor (EBDI001) to control what the communication ID and password contain on BG segments surrounding acknowledgments. This record indicates to the Inbound Editor that the BG PASSWORD on the incoming document (BG02) must be matched against the BG PASSWORD field on the partner control record. If the BG PASSWORD equals spaces, this parameter is ignored. If the BG PASSWORD is not equal to the password of the incoming BG document, an error message (#380) is generated. This is required to observe UCS/WINS identification conventions, which are normally used with the BG.

**BG PARTNER YES**

Position	Field Content	COBOL Picture	Format	Length	Description
001-011	ID	X(11)	AN	11	Constant BG PARTNER.
012-014	Partner Indicator	X(03)	AN	3	Indicates that partner profile is to be used to retrieve data to build BG segment for acknowledgments. Valid value: YES
015-080	Filler	X(66)	AN	66	Not used

**GENERATE Record**

This record tells the Inbound Editor which type of functional acknowledgment to generate. This is done on a per version level. It is the method used for Gentran:Basic to generate a 999 type acknowledgment (acceptance/rejection). The default is 997.

**GENERATE 999 FOR VERSION-**

The version is changed to a valid version identifier (per Gentran:Basic Tables file). This would generate a 999 type of functional acknowledgment (acceptance/rejection) whenever this version is the version of the inbound data for all partners using this version. Otherwise, the Editor determines the acknowledgment type through the transaction record. If no acknowledgment is found, the Editor defaults to a 997 type of acknowledgment. The GENERATE 999 parameter can be overridden now by the partner profile if PARTNER ACKNOWLEDGMENT is on and group level acknowledgments are being generated. If the Partner Transaction record specifies a 997 for a partner, a 997 is generated.

Position	Field Content	COBOL Picture	Format	Length	Description
001-008	ID	X(08)	AN	8	Constant GENERATE
009-009	Filler	X(01)	AN	1	Not used
010-012	Output Trans	X(03)	AN	3	Transaction number for acknowledgment transaction. Valid values: 997, 999
013-013	Filler	X(01)	AN	1	Not used
014-016	ID	X(03)	AN	3	Constant FOR
017-017	Filler	X(01)	AN	1	Not used
018-029	Version	X(12)	AN	12	Version for which to generate 997 or 999.
030-030	Filler	X(01)	AN	01	Not used
031-033	Agency Code	X(03)	AN	03	The controlling agency code associated with the indicated version.
034-080	Filler	X(47)	AN	47	Not used

### *BUSINESS DOCUMENT TRACKING Record*

#### **BUSINESS DOCUMENT TRACKING –**

This record tells the inbound Editor to pass the Message Store location of each claim on the CLM segment to the add-on product, Examiner.

<b>Position</b>	<b>Field Content</b>	<b>COBOL Picture</b>	<b>Format</b>	<b>Length</b>	<b>Description</b>
001-026	Business Document Tracking	X(26)	AN	26	Indicates that the add-on product, Examiner, is present and that claim tracking is to be performed on the Databank Message Store. Valid value: BUSINESS DOCUMENT TRACKING
027-080	Filler	X(54)	AN	54	Not used

### **Outbound (Only) Parameter Record Layouts**

#### *Databank Record*

This record instructs the Outbound Editor on which, if any, databank features you wish to use.

#### **OUTBOUND EDI INTERCHANGE FULL**

This parameter specifies that for each interchange, the Outbound Editor is to generate both directory (audit) and message store entries for the databank.

#### **OUTBOUND EDI INTERCHANGE DIRECTORY**

This parameter specifies that for each interchange, the Outbound Editor is to generate directory (audit) entries for the databank.

#### **OUTBOUND EDI INTERCHANGE NONE**

This parameter specifies that for each interchange, the Outbound Editor does not use the databank feature.

#### **OUTBOUND EDI GROUP DIRECTORY**

This parameter specifies that for each group level, the Outbound Editor is to generate directory (audit) entries for the databank.

#### **OUTBOUND EDI GROUP NONE**

This parameter specifies that for each group level, the Outbound Editor does not use the databank feature.

#### **OUTBOUND EDI TRANSACTION DIRECTORY**

This parameter specifies that for each transaction, the Outbound Editor is to generate directory (audit) entries for the databank.

**OUTBOUND EDI TRANSACTION NONE**

This parameter specifies that for each transaction, the Outbound Editor does not use the databank feature.

Position	Field Content	COBOL Picture	Format	Length	Description
001-034	Setting	X(34)	AN	34	Parameter indicating level of auditing to be done. Valid values are constants: <ul style="list-style-type: none"> <li>• OUTBOUND EDI INTERCHANGE FULL</li> <li>• OUTBOUND EDI INTERCHANGE DIRECTORY</li> <li>• OUTBOUND EDI INTERCHANGE NONE</li> <li>• OUTBOUND EDI GROUP DIRECTORY</li> <li>• OUTBOUND EDI GROUP NONE</li> <li>• OUTBOUND EDI TRANSACTION DIRECTORY</li> <li>• OUTBOUND EDI TRANSACTION NONE</li> </ul>
035-080	Filler	X(46)	AN	46	Not used

**Output Compression (Wrap) Records**

These records are used by the Outbound Editor (EBDI002) to tell Gentran:Basic if it should wrap the data; if so, the record also specifies at what level the data should be wrapped. Wrapping can improve transmission efficiency by eliminating trailing spaces. The default value is COMPRESS TRANSMISSION. All plurals are used in the same manner as their singular counterpart (e.g., compress transmission is the same as compress transmissions). If there is more than one compression record in the Global Parameters file, the last one is used and all previous records are ignored.

**COMPRESS TRANSMISSION –**

This parameter indicates to the Outbound Editor that it should compress from the interchange beginning segment identifier to the ending interchange segment identifier (ISA/IEA, ICS/ICE, BG/EG, UNB/UNZ). All interchange segments are to start in position 1 of a new record when writing the output file. All other segments are wrapped.

**COMPRESS GROUP –**

This parameter indicates to the Outbound Editor that it should compress from the GS segment to the GE segment. All interchange and group segments are to start in position 1 of a new record when writing the output file. All other segments are wrapped.



**COMPRESS TRANSACTION –**

This parameter indicates to the Outbound Editor that it should compress from the ST segment to the SE segment. All interchange, group, and transaction segments are to start in position 1 of a new record when writing the output file. All other segments are wrapped.

**COMPRESS SEGMENT –**

This indicates to the Outbound Editor that it should compress individual segments. All segments start in position 1 of new records. Therefore, no segments are wrapped.

Position	Field Content	COBOL Picture	Format	Length	Description
001-010	Compress ID	X(10)	AN	10	Constant: COMPRESS
011-023	Compress Level	X(12)	AN	12	Indicates the level of compression or wrapping. Valid values: <ul style="list-style-type: none"> <li>• TRANSMISSION</li> <li>• GROUP</li> <li>• TRANSACTION</li> <li>• SEGMENT</li> </ul>
024-080	Filler	X(57)	AN	57	Not used

**Delete Blank Element Record**

**DELETE OPTIONAL BLANK ELEMENTS –**

This parameter is used by the Outbound Editor for only EDIFACT and ANA TRADACOMS documents. Elements which contain spaces and are defined in the table file as optional are removed. Example: \*~~bbbb~~\* becomes \*\*.

**DELETE ALL BLANK ELEMENTS –**

This parameter is used by the Outbound Editor for only EDIFACT and ANA TRADACOMS documents. Elements which contain spaces are removed. Example: \*~~bbbb~~\* becomes \*\*.

Position	Field Content	COBOL Picture	Format	Length	Description
001-030	TRADACOMS Compression	X(30)	AN	30	Indicates for TRADACOMS documents whether or not additional compression is requested. Valid values: <ul style="list-style-type: none"> <li>• DELETE OPTIONAL BLANK ELEMENTS</li> <li>• DELETE ALL BLANK ELEMENTS</li> </ul>
031-080	Filler	X(50)	AN	50	Not used

### Output Databank Record

Databank offers an extract feature for outbound EDI documents. If you want to use the databank Extract feature, you may not want an output file created from the Outbound Editor.

See Chapter 5 for more information about the EDID205 program.

#### OUTPUT DATABANK ONLY

This global parameter instructs the Outbound Editor to verify that the message store feature of databanking is active. If message store is active, do not create an output file. Otherwise, ignore this global parameter and create an output file from the Editor.

This parameter only applies if used with the System-level global parameter OUTBOUND EDI INTERCHANGE FULL.

**Note:** With OUTPUT DATABANK ONLY, the \$TCR free-form 80-byte information is placed only on the Outbound EDI Databank Message Store (EDIOES).

Position	Field Content	COBOL Picture	Format	Length	Description
001-020	Output Databank On	X(20)	AN	20	Indicates to the Editor that if Databank Message Store is being performed, do not create an Output file for the Outbound Split Program. Valid value: OUTPUT DATABANK ONLY
021-080	Filler	X(60)	AN	60	Not used

### Viewpoint User Tracking Record

This record informs the Outbound Editor whether or not Gentran:Viewpoint Tracking Management monitoring continues after the outbound flow. To continue tracking the EDI data, a user written tracking program is required. The user tracking program must call the appropriate Gentran:Viewpoint User Gateway using the information supplied by the Outbound Editor on the Viewpoint User Tracking record. The default is that no tracking information will be performed for communications.

### TRACK OUTPUT FILES

This parameter indicates to the Editor that the Viewpoint User Tracking records should be written to the output file. This parameter works in association with the Output Databank record and the Directed Output record. Viewpoint User Tracking records are created when the Editor is writing data to the sequential EDI output file. When using 'FULL' Databank and the Outbound EDI Databank Extract, the Extract program creates the Gentran:Viewpoint User Tracking records. If data separation is requested, the Editor includes a User Tracking record for each unique split file that is specified on the Partner File Data Separation records during the current run.

Position	Field Content	COBOL Picture	Format	Length	Description
001-018	Track Output Files	X(18)	AN	18	Indicates whether or not Viewpoint Tracking is required after the Editor. Valid value: TRACK OUTPUT FILES
019-080	Filler	X(62)	AN	62	Not used

### PLUS BATCH NUMBER OPTIONS

Under normal processing, the Outbound Editor assigns the communications batch number. This record allows the batch number to be posted on the Outbound EDI Databank. The user can see the communication status from the databank screens.

### PLUS WILL ASSIGN BATCH NUMBER

This parameter instructs the Outbound Editor to not assign a communications batch number or post one to the databank. The STOUTL ADD utility assigns the batch communications batch number after the Editor has run. If you use this option, you cannot view communications status from the databank screens.

Position	Field Content	COBOL Picture	Format	Length	Description
001-029	Gentran:Plus assigns a batch number.	X(29)	AN	29	Indicates that the STOUTL ADD utility assigns the communications batch number. Valid value: PLUS WILL ASSIGN BATCH NUMBER.
030-080	Filler	X(51)	AN	51	Not used



# Determining DASD Standards Space Requirement

## Overview

This appendix contains the following topics:

<b>Topic</b>	<b>Page</b>
<b>Calculating DASD Requirements.....</b>	<b>B-2</b>
Sample Report.....	B-3
Sequential Calculation .....	B-4
VSAM Calculation.....	B-4
Sample Calculation .....	B-5

## Calculating DASD Requirements

The following table contains information that can be used to calculate DASD space requirements for sequential and VSAM Standards file. These calculated space allocations can be used for DEFSTD and UPDSTD2 processes.

<b>File</b>	<b>Rec Size</b>	<b>Block Size</b>	<b>Recs/ Block</b>	<b>CI Size</b>	<b>Recs/ CI</b>
Transaction	160	32640	204	2048	12
Segment	100	32700	327	2048	20
Segment Desc	120	32760	273	2048	16
Element	130	32760	252	2048	15
Element Desc.	130	32760	252	2048	15
Activity	170	32640	192	2048	11
Dictionary	70	32760	468	2048	29
Code 1	140	32760	234	2048	14
Code 2	140	32760	234	2048	14
Code 3	150	32700	218	2048	13
Code 4	170	32640	192	2048	11
Association	80	32720	409	2048	25

## Calculating DASD Requirements

### Sample Report

The following is a sample of the Standards Extract report that is located in the STDINCL.TXT file that is part of the Standards package. This is a detailed list of the versions that are included in the Standards package. It will be used for DASD space allocation calculations when installing or updating Standards.

EBDI038 12/01/2005 12:00			STANDARDS EXTRACT REPORT										PAGE 1
VERSION	AGENCY	TRANS	SEGMENT	SEG DESC	ELEMENT	ELE DESC	ACTIVITY	DICT	CODE 1	CODE 2	CODE 3	CODE 4	
ACHG02	A	2	37	37	211	153	17	153	621	4197	0	0	
ADJL01	A	2	101	77	588	388	17	388	4197	21058	0	0	
ADJL04	A	2	101	92	819	406	17	406	4560	21469	0	0	
ADJL05	A	2	111	99	852	412	17	412	4906	21058	0	0	
ADJL06	A	2	121	100	818	444	17	444	5626	21469	0	0	
AISI50	D	10	84	18	191	238	0	238	1493	0	0	0	
ANAO01	A	182	1020	647	3996	2271	18	2271	5513	936	30	0	
ATRM01	A	2	31	31	176	118	17	118	388	4197	0	0	
ATRM02	A	2	43	39	202	136	17	136	615	4197	0	0	
ATRM04	A	2	43	39	207	145	17	145	689	4745	0	0	
AZ/4	T	16	218	87	536	326	17	326	2664	239	0	0	
AZ/5	T	16	218	87	536	326	17	326	2704	349	0	0	
AZ/6	T	16	233	88	577	336	17	336	3147	365	0	0	
AZ/7	T	15	215	84	533	323	17	323	3301	322	0	0	
AZ/8	T	15	248	95	574	313	20	313	3711	80	0	0	
CLNM01	A	2	51	48	328	238	17	238	872	21058	0	0	
CLNM04	A	2	51	48	328	238	17	238	918	21655	0	0	
CLPM01	A	2	35	35	210	147	17	147	434	4197	0	0	
CLPM04	A	2	35	35	210	145	17	145	453	4745	0	0	
CLUF01	A	2	30	30	159	102	17	102	498	0	0	0	
CMPR01	A	2	77	77	529	311	17	311	2947	5007	0	0	
CRED01	A	2	38	38	226	163	17	163	510	6677	0	0	
CVNO03	A	2	46	44	258	189	17	189	606	21606	0	0	
CVNO04	A	2	46	44	268	191	17	191	715	21868	0	0	
CVNO05	A	2	47	45	271	192	17	192	671	21058	0	0	
CVNO06	A	2	47	45	279	206	17	206	847	21868	0	0	
CVNT02	A	2	37	37	217	156	17	156	545	21058	0	0	
C2/4	T	26	471	108	681	360	17	360	2608	58	0	0	
C2/5	T	27	514	117	732	369	17	369	2604	58	0	0	
C2/6	T	31	563	127	799	380	17	380	7689	73	0	0	
C2/7	T	31	575	129	827	389	17	389	3501	9	0	0	
C2/8	T	31	576	128	838	385	17	385	4024	66	0	0	
C2/9	T	34	547	136	909	394	17	394	4745	73	0	0	
D 00A	UN	187	10700	198	1659	761	27	761	9514	165	109	23	
D 00B	UN	191	11007	199	1663	763	27	763	9585	165	362	23	
D 01A	UN	194	11313	199	1663	763	27	763	9691	166	368	23	
D 01B	UN	194	11320	199	1663	763	27	763	9739	166	370	24	
D 01C	UN	194	11339	199	1663	763	27	763	9858	167	372	24	
D 02A	UN	194	11345	199	1663	763	27	763	9897	167	372	24	
D 02B	UN	194	11360	197	1647	763	27	763	9939	57	372	24	
D 03A	UN	192	11331	197	1647	762	27	762	9839	57	118	24	
D 03B	UN	192	11341	197	1647	762	27	762	9844	57	118	24	
D 93A	UN	57	3948	130	1016	532	27	532	5357	0	0	0	
D 94A	UN	70	4330	133	1052	538	27	538	5904	64	0	0	
D 94B	IA	4	73	50	350	226	27	226	2703	0	0	0	
D 94B	UN	79	4616	136	1064	544	27	544	5953	64	0	0	
D 95A	UN	102	5454	140	1121	569	27	569	6090	64	0	0	
D 95B	UN	116	6020	159	1270	628	27	628	6435	67	41	0	
D 96A	UN	126	6581	159	1302	607	17	607	6791	76	41	0	
D 96B	UN	137	7318	177	1482	712	27	712	6886	76	83	0	
D 97A	UN	144	7556	177	1498	701	27	701	7219	83	83	0	
D 97B	UN	152	8079	179	1512	707	27	707	7393	83	83	0	
D 98A	UN	158	8422	180	1525	713	27	713	7710	83	73	21	
D 98B	UN	166	8975	190	1613	740	27	740	8129	101	79	21	
D 99A	UN	170	9274	194	1656	750	27	750	9089	104	87	21	
D 99B	UN	181	10201	199	1721	745	27	745	9599	130	107	23	
DCHG02	A	2	51	51	286	197	17	197	1903	4197	0	0	
DDOC02	A	2	33	33	167	109	17	109	315	4197	0	0	
DEFAULT	SC	4	45	42	224	133	17	133	550	0	0	0	

The following calculations are based on 3390 information.

## Sequential Calculation

To calculate in TRACKS:

$$\text{TRACKS} = \text{TOTAL \# OF RECORDS} / \text{RECORDS PER BLOCK}$$

(round up to next integer)

Total # of records can be determined by using the Standards list located in file 1 of the Standards media.

Records per block can be found on the Calculating DASD Requirements table on page B-2.

**Note:** Block sizes have been adjusted to allow one block per track.

To convert to CYLINDERS:

$$\text{CYLINDERS} = \text{TRACKS} / 15 \text{ (round up to next integer)}$$

## VSAM Calculation

1. Calculate FREE SPACE = 'Free Space definition % control interval' X 2048

**Note:** VSAM Definitions shipped with Gentran:Basic do not contain free space. If free space is not used, this value is zero.

2048 represents the CISIZE in the VSAM Definitions shipped with the product. If this value changes, that number must be reflected here.

2. Determine the RECORDS PER CI. If you are not using free space, select this value from the Calculating DASD Requirements table on page B-2.

If using free space, use this calculation:

$$\text{RECORDS PER CI} = ( 2038 - \text{FREE SPACE} ) / \text{Record size}$$

2038 represents the CISIZE used in the VSAM Definitions shipped with the product minus 10 bytes for 2 RDFs and 1 CIDF.

You calculated FREE SPACE in Step 1.

See the Calculating DASD Requirements table on page B-2 for record size.

3. Calculate # LOADED CI PER CA = 315 - ( 'Free space definition % control area' X 315 ).

315 represents the number of CI per CA. This is based on the default CISIZE of 2048 that is shipped with the product. If CISIZE has changed, refer to DFSMS documentation to determine a new value for number of CI per CA.

**Note:** VSAM Definitions shipped with Gentran:Basic do not contain free space. If free space is not used, the # LOADED CI PER CA is 315.



## Calculating DASD Requirements

---

- Calculate RECORDS PER CYLINDER = RECORDS PER CI X # LOADED CI PER CA.

You calculated RECORDS PER CI in Step 2.

You calculated # LOADED CI PER CA in Step 3.

- Calculate TOTAL ALLOCATION = Number of records / RECORDS PER CYLINDER

Number of records represents the total number of records to be loaded to this VSAM file. If the calculation is for the UPDSTD2 process, this includes the number of new records to be added, plus the number of existing records already on your Standards files.

You calculated RECORDS PER CYLINDER in Step 4.

### Sample Calculation

Versions D 96A and DEFAULT are to be unloaded from the Standards by means of the UPDSTD2 process and added to existing Standards files.

The following record counts are based on the report in the STDINCL.TXT file.

	Trans	Seg	Segdesc	Elem	El Desc	Dict	Activity	Code 1	Code 2	Code 3	Code 4
<b>D 96A</b>	126	6581	159	1302	607	607	17	6791	76	41	0
<b>DEFAULT</b>	4	45	42	224	133	133	17	550	0	0	0
<b>Subtotal</b>	130	6626	201	1526	740	740	34	7341	76	41	0
<b>Existing</b>	292	4460	1283	8401	3534	3534	205	16788	1098	26	0
<b>Total</b>	422	11086	1484	9927	4274	4274	239	24129	1174	67	0

- To Calculate required space in tracks for sequential extract files (GENTRAN.V6X4.EXTRACT.xxxxxxxx) for STEP16 of UPDSTD2 process:

**Calculation:**

TRACKS = TOTAL # OF RECORDS / RECORDS PER BLOCK (rounded)

*TOTAL # OF RECORDS* - Extract files contain only the new Standard information  
Use the "Subtotal" line above for counts.

*RECORDS PER BLOCK* - This information from the Calculating DASD Requirements table on page B-2.

File	TRACKS =	Round Up To
EXTVER (Version)	This file is very small. Normally 1 – 5 tracks is sufficient.	
EXTTRN (Transaction)	130 / 204 = .64	1 track
EXTSEG (Segment)	6626 / 327 = 20.26	21 tracks
EXTSGD (Segment Desc)	201 / 273 = .74	1 track
EXTELE (Element)	1526 / 252 = 6.06	7 tracks
EXTELD (Element Desc)	740 / 252 = 2.94	3 tracks
EXTDIC (Dictionary)	740 / 468 = 1.58	2 tracks
EXTACT (Activity)	34 / 192 = .18	1 track
EXTCD1 (Code 1)	7341 / 234 = 31.37	32 tracks
EXTCD2 (Code 2)	74 / 234 = .32	1 track
EXTCD3 (Code 3)	41 / 218 = .19	1 track
EXTCD4 (Code 4)	0 / 192 = 0	1 track

2. To Calculate required space for sequential output files (GENTRAN.V6X4.SEQ.STD.xxxxxxxx) for STEP7 of UPDSTD2 process:

Calculation TRACKS = TOTAL # OF RECORDS / RECORDS PER BLOCK (rounded)

*TOTAL # OF RECORDS* - Output files contain the new Standard information and existing Standards information; therefore, use the “Total” line above for counts.

*RECORDS PER BLOCK* - This information is from the Calculating DASD Requirements table on page B-2.

File	TRACKS =	Round Up To
NEWVER (Version)	This file is very small. Normally 1 – 5 tracks is sufficient.	
NEWTRN (Transaction)	422 / 204 = 2.07	3 tracks
NEWSEG (Segment)	11086 / 327 = 33.9	34 tracks
NEWSGD (Segment Desc)	1484 / 273 = 5.4	6 tracks
NEWELE (Element)	9927 / 252 = 39.39	40 tracks
NEWELD (Element Desc)	4274 / 252 = 16.96	17 tracks
NEWDIC (Dictionary)	4274 / 468 = 9.31	10 tracks
NEWACT (Activity)	239 / 192 = 1.24	2 tracks
NEWCD1 (Code 1)	24129 / 234 = 103.12	104 tracks
NEWCD2 (Code 2)	1174 / 234 = 5.02	6 track

## Calculating DASD Requirements

---

File	TRACKS =	Round Up To
NEWCD3 (Code 3)	$67 / 218 = .31$	1 track
NEWCD4 (Code 4)	$0 / 192 = 0$	1 track

3. To Calculate required space for VSAM files (GENTRAN.V6X4.VSAM.STD.xxxxxxxx) for STEP8 through STEP30 of UPDSTD2 process:

Calculation  $TOTAL\ ALLOCATION = NUMBER\ OF\ RECORDS / (RECORDS\ PER\ CI \times 315)$

*NUMBER OF RECORDS* - VSAM files contain the new Standard information and existing Standards information; therefore, use the "Total" line above for counts.

*RECORDS PER CI* - This information is from the Calculating DASD Requirements table on page B-2.

File	ALLOC =	Round Up To
VERSION	This file is very small. Normally 5 – 10 tracks is sufficient.	
TRANSACTION	$422 / (12 \times 315) = .11$	1 cylinder
SEGMENT	$11086 / (20 \times 315) = 1.76$	2 cylinders
SEGMENT DESC.	$1484 / (16 \times 315) = .29$	1 cylinder
ELEMENT	$9927 / (15 \times 315) = 2.10$	3 cylinders
ELEMENT DESC.	$4274 / (15 \times 315) = .90$	1 cylinder
DICTIONARY	$4274 / (29 \times 315) = .47$	1 cylinder
ACTIVITY	$239 / (11 \times 315) = .07$	Use 1 track
CODE 1	$24129 / (14 \times 315) = 5.47$	6 cylinders
CODE 2	$1174 / (14 \times 315) = .26$	1 cylinder
CODE 3	$67 / (13 \times 315) = .02$	Use 1 track
CODE 4	$0 / (11 \times 315) = 0$	Use 1 track



# Glossary

## Definition of Terms

### Acknowledgment

A special transaction set (e.g., 997, 999, or CONTRL) that is transmitted to the original sender to indicate the status of a transmission with respect to standards adherence. There are indicators that imply acceptance of the contents of the associated transaction sets.

Acknowledgments are also known as Functional Acknowledgments (the 997 transaction set), Accept/Reject Advices (the 999 transaction set), and Syntax and Service Report Message (the CONTRL message).

Gentran:Basic differentiates between inbound acknowledgments (those from your trading partner) and outbound acknowledgments (those to go to your trading partner). Gentran:Basic also deals with *expected acknowledgments*, which should be satisfied by inbound acknowledgments.

See also *Functional Acknowledgment*.

### Addressing Method

A mean of identifying the sender and intended receiver of an incoming EDI message.

Gentran:Basic supports several addressing methods to enable you to configure the system to your current technique, or your industry's preferences.

### Application

An in-house program or series of programs designed to process specific jobs. These programs are formatted by the customer for their business needs (e.g., payroll).

### Application Databank

One of two application databanks available: Inbound (incoming) and Outbound (outgoing). The databank contains application documents as well as control and routing information. Inquiry capabilities are available both online and in batch processing to provide fast answers to the operation of your EDI system. Document archive, restore, and reset capabilities complete the picture to address audit, re-transmit, and operational requirements.

### Application Data ID

An identification of the application definition as defined by the user via the Mapping subsystem. This definition is used by the Mappers to process a user's application file.

## Definition of Terms

---

### Application Definition

The definition of the application interface file used during the mapping process.

### Application ID

See *Application Data ID*.

### Archive

A permanent record of messages transmitted or received.

### Batch Processing

A type of data processing operation and data communications transmission during which related transactions are grouped together and transmitted for processing. Usually batch processing occurs by the same computer and under the same application. It is generally regarded as non-realtime data traffic consisting of large files: batch processing is used for the type of data traffic for which network response time is not critical, compared with interactive processing.

### Compress

An automatic process in the outbound flow of Gentran that remove all leading zeroes and trailing spaces from an EDI data element. Empty elements and subelements (those containing all spaces) are set to a null or zero length. All trailing data elements (those data elements between the last element containing information and the segment terminator) are removed, along with their respective element separators. The segment terminator is then shifted to the end of the last element. For real type data elements, the zero significant digit just before and just after a decimal point is not removed.

### Conditional Code

The relational definition of data elements within an EDI segment. There are these five conditional codes: P, R, E, C, and L. These codes are defined through the Segment Element Conditional Code Maintenance screen (EDIM141) and are used by the Inbound Editor program (EBDI001) and the Outbound Editor program (EBDI002). The five conditional codes are listed and described below:

- |          |   |  |
|----------|---|--|
| <b>P</b> | = | <i>(Paired)</i> If any of the referenced data elements is present, they all must be present.   |
| <b>R</b> | = | <i>(Required)</i> At least one of the referenced data elements must be present.  |
| <b>E</b> | = | <i>(Exclusive)</i> Only one of the referenced data elements may be present.  |
| <b>C</b> | = | <i>(Conditional)</i> If the first referenced data element is used, then all remaining referenced data elements must be used.                             |
| <b>L</b> | = | <i>(List Conditional Paired)</i> If the first referenced data element is used, then at least one of the remaining referenced data elements must be used. |

### Condition Code

A numeric value issued by a program to indicate the status of its execution. This status value is also called a *return code*. A condition code can range in value from 0 to 4096. Testing a condition code in a JCL job stream requires a COND parameter.

All main processing programs (inbound and outbound) issue condition codes in Gentran. However, the key programs are Inbound Editor (EBDI001), Inbound Mapper (EBDI041), Outbound Mapper (EBDI042), and Outbound Editor (EBDI002).

### Control Number

See *Databank Envelope Reference ID*.

### Databank

A repository for storing document/transaction data and any related control information. Using the Databank subsystem, you can control and monitor the document/transaction flow through the Gentran system. The four databanks are the following: outbound application databank, outbound EDI databank, inbound EDI databank, inbound application databank.

### Databank Change Audit

The file that stores detailed information about all user updates performed against the databanks using the Databank subsystem. There is one Change Audit file for each databank.

### Databank Configuration

Identification of how you have configured a databank. For each of the databanks, the following configurations are valid:

Directory=	Only Directory data is maintained.
Full =	Both Directory and Message Store data are maintained.
None =	Neither Directory nor Message Store data is maintained.

**Note:** Full databanking is not available for Gentran:Structure customers' fixed-format data.

### Databank Directory

The area of a databank that stores control and status information (e.g., interchange envelope reference ID, acknowledgment status, or document mapped status). There is one directory for each databank.

### Databank Envelope Reference ID

An ID used to uniquely identify an EDI envelope for a trading partner. Formerly called *control number*. The envelope reference ID's used in the databank are the following: Interchange Envelope Reference ID, Group Envelope Reference ID, Transaction Envelope Reference ID.

### Databank Message Store

The area of a databank in which the actual application records or EDI segments are stored. There is one message store for each databank.

## Definition of Terms

---

### Databank Run Number

An 8-digit number assigned by Gentran to identify the run that processed the document for a databank.

### Data Dictionary

The publication that defines all of the data elements that exist for a standard. The data dictionary describes data types (e.g., length and usage) and all code values for an element, if applicable.

### Data Element

See *Element*.

### Data Element Dictionary

See *Data Dictionary*.

### Data Element Separator

See *Delimiter*.

### Data Segment

See *Segment*.

### Decompress

An automatic process in the Inbound Editor that expands each data element to its maximum length as defined by the online EDI standards. Decompression of data elements is accomplished by inserting trailing spaces or leading zeroes into each element until the element is at its maximum length. All numeric data elements automatically have three bytes added to their maximum length. These three bytes contain a sign (space or -), along with the number of decimal positions for the data element.

### Delimiter

A special character used to separate and mark the end of data segments, data elements, and data subelements within an EDI standard. The data segment terminator, data element separator, and data subelement separator must each contain unique special characters.

### Distribution

The process of controlling and reporting the flow of documents to and from user applications and outside networks/partners. This distribution function is supported by the EDI Databank application. See also *EDI Databank*.

### Document

See *Message*.



### **Document Status**

A code that describes the current status of a document on the databanks. For example, an EDI document placed on the databank would have a status of acknowledged after the acknowledgment is received from the trading partner.

### **DUNS Number**

Dun & Bradstreet's assigned number used to distinguish a corporate entity. Often used in EDI documents to determine sender and receiver identification.

### **EDI (Electronic Data Interchange)**

The exchange of routine business transactions in a computer-processible format, covering such traditional applications as inquiries, planning, purchasing, acknowledgments, pricing, order status, scheduling, test results, shipping and receiving, invoices, payments, and financial reporting.

### **EDI Databank**

One of two EDI databanks: inbound (incoming) and outbound (outgoing). The databank contains EDI documents, as well as control and routing information. Inquiry capabilities are available both online and in batch processing to provide quick answers to the operation of your EDI system. Document archive, restore, and reset capabilities complete the picture to address audit, re-transmit, and operational requirements.

### **EDI Databank Inquiry**

An online reporting tool used to determine the status and other significant information about EDI messages. Gentran:Basic also provides this function in batch mode. A multitude of selection criteria can be specified to quickly locate any documents of interest.

### **Editor**

One of two editors: the Inbound Editor (EBDI001) and the Outbound Editor (EBDI002). The Inbound Editor uses predetermined instructions to verify, check, and possibly reject the EDI standard data you have received from a trading partner, and moves the edited data to split files and then to the appropriate inbound application files via Mapping. The Outbound Editor prepares your data for transmission to your trading partners. The Editor checks all data to ensure compliance with the selected public standard and wraps the data based on compression method.

### **Element**

The smallest named item in a segment that can convey data. For example, Bill To City is one data element within a segment in an EDI standard.

### **Envelope**

The transmission header and trailer enclosing an EDI document. The envelope is automatically generated by Gentran:Basic using the information on the Partner file. Envelopes exist at the interchange, group, and transaction levels.

## Definition of Terms

---

### **EOT**

End of Transmission.

### **Field**

The smallest item of information in a record in a defined application.

### **Functional Acknowledgment**

A special transaction set transmitted by a receiver as an affirmative response to the sender. Gentran automatically generates a functional acknowledgment for documents received. There is never an acknowledgment generated for an acknowledgment.

### **Gentran:Control**

An add-on product to Gentran:Basic that enables you to automate the EDI processing stream and prioritize processing by trading partner and/or type of document. Gentran:Control enables you to organize processing and responsiveness based upon business criteria, not system limitations. In addition, the application and communications gateways expedite the movement of documents across the batch/CICS barrier.

### **Gentran:Plus**

A comprehensive software system that provides a seamless application-to-EDI interface. Gentran:Plus can work in conjunction with Gentran:Basic, or as a stand-alone product. Gentran:Plus contains these three subsystems: Communications, Translation, and Communications Setup and Management. Gentran:Plus can accommodate an unlimited number of transactions while using an unlimited number of EDI standards versions. It supports multiple lines with different protocols for concurrent EDI sessions. This combination of translation, communications, and mailboxing represents a highly efficient EDI resource that can be used to automate your communications needs.

### **Gentran:Realtime**

An add-on product to Gentran:Basic that facilitates the transmission of data processing in an operating mode when data is entered in an interactive session, as opposed to a batch session. Response time is drastically reduced over batch processing, providing the competitive benefits of online speed, flexible control, and universal EDI standards to many business applications.

### **Gentran:Structure**

An add-on product to Gentran:Basic that enables you to define proprietary fixed-format standards using the Standards subsystem, and map to and from these standards using the Inbound and Outbound Mapping programs.

### **Gentran:Viewpoint**

A Gentran product that consists of these two modules: Exception Management and Tracking Management. Exception Management enables you to capture exception information, notify the appropriate user, and optionally take automated action for exceptions that are specific to their EDI business needs. Tracking Management is an optional module that enables you to query on the status and history of a document as it moves through the entire end-to-end EDI environment.

## Header

The portion of a message that precedes the body and trailer of the business transaction.

## In-house Interface

Routine(s) developed by application personnel that connect application systems to Gentran:Basic. These routines either create Application Interface files for outbound transactions or process an Application Interface file containing inbound transactions.

## Logical Document

The treating of several related ANA TRADACOMS messages as one document. A basic TRADACOMS document contains a header message, a detail message, and a summary message. Each message has its own envelope. Individually, they are meaningless, but together they represent a whole (complete) document.

## Mandatory

A requirement designator value indicating that a segment, element, or component data element *must* be used within an EDI standard.

## Mapping

Map processing that uses user-defined transaction maps to translate business documents from the internal application data format to the selected EDI standard, or from the EDI standard to an internal application format. Transaction maps are defined using the Mapping subsystem.

## Message

An identified and structured set of elements and segments covering the requirements for a specific transaction.

## Message Status Level

A status level given to messages. The three status levels are listed and described below:

- Status 0 = Draft document for information only.
- Status 1 = Draft document approved for formal trial.
- Status 2 = Recommended approved by the UN/EDIFACT governing body WP.4.

UN/EDIFACT messages need to be submitted, reviewed, and added to by regional boards.

## Network

A third-party EDI facilitator (EDI mailbox). Networks are defined to Gentran:Basic for reconciliation purposes only.

## Partner Profile

A collection of records in Gentran that contain information about a trading partner.

## Definition of Terms

---

### Partner/Qualifier Mode

This is trading partner processing method which defines your organization and each trading partner as separate and distinct entities on the partner profile. This mode permits each trading partner to be used in multiple relationships.

### Record

A logical collection of data elements (fields) grouped together.

### Relationship Mode

This is trading partner processing method provides uniqueness for identification purposes that would not be possible if the trading partner was defined using the Partner/Qualifier method. This mode requires you to first identify the inter- or intra- business relationships. Each identified relationship is then defined separately and distinctly on the partner profile. This method is ideal for defining multiple entities within your organization with shared (common) trading partners.

### Release Character

A syntax character defined in the ANA TRADACOMS and EDIFACT standards to override a special character used in the text of a document differently than its system definition. For example, if a quote is defined as a segment terminator and is used in the content of a text message as a quote, the release character must precede the quote in the text to indicate to the system that the character is only a character and not a segment terminator.

### Release Indicator

See *Release Character*.

### Return Code

See *Condition Code*.

### Segment

A logical group of data elements. The data elements are sequentially placed within the group. A segment starts with a segment code and ends with a segment terminator within an EDI standard.

### Segment Separator

A syntax character used in the ANA TRADACOMS standards to isolate the segment ID from the rest of the data elements in a segment. The only valid value to be used as a segment separator under current TRADACOMS standards is an equals sign (=).

### Segment Terminator

The delimiter that identifies the end of a segment. This delimiter must be a special character (e.g., no alphabetic or numerics or space permitted).

### **Selection Criteria**

Criteria used to control document processing and reporting functions. Examples of some of the fields available for use as selection criteria are the following: User Document Reference (P.O. Number, Invoice Number), User, Partner, Network, Application, Transaction Type, Date, Time, Document Status, Document Acknowledgment Status, Document Validation Status, Duplicate Status, Test/Production Status, Interchange Control Numbers, and other EDI envelope information.

### **Simple Data Element**

A data element whose data item representation embodies a single concept (i.e., a data element that is not made up of component data elements).

### **Subelement Separator**

A syntax character that is used to separate the components of an element. The syntax character must be a special character (alphabetic and numeric not allowed) that is not displayed in any of the text for a document. It must also be different from the segment terminator and data element separator.

### **Summary Area**

The portion of a message that follows the body of the message and contains summary information relating to the entire message.

### **Trading Partner**

Any organization with whom you do business. Typically, trading partners are customers, suppliers, and intra-company divisions.

### **Trading Partner Processing Mode**

A system processing methodology that enables you to define partner profiles. The Gentran:Basic system provides three processing modes from which to choose: Partner/Qualifier mode, Relationship mode, and Mixed mode.

### **Transaction Map**

A set of instructions detailing the correspondence between the EDI message standard and your application transaction. The transaction map is also used to document your application's use of EDI and to document your data requirements to your partners.

### **Transaction Set**

See *Message*.

### **Turnaround Time**

The maximum allowable time between the sending of a document and the return of its acknowledgment. If the acknowledgment is not received within this time, an alert is raised and the originating document is flagged as Overdue.

### Unwrap

The process of identifying each EDI segment of a wrapped document and spreading them so that each segment begins its own record. If the record is defined as fixed length, it is padded with spaces. The Gentran programs that unwrap segments are the Inbound Unwrap (EBDI010 and EBDI010A) and Inbound Editor (EBDI001).

### User

The term used to identify your organization to Gentran:Basic. You may define one or more users for your organization, with each user having its own unique EDI identifier. This is used when defining a User/Partner relationship providing enhanced auditing capabilities.

### User Document Reference

A user-defined field that provides a way of referring to EDI documents by your existing methods (e.g., documents can be referred to by your Purchase Order Number rather than by referring to an EDI control number). The user document reference is supported for both inbound and outbound documents.

### User Reference

User-defined information that can provide a relationship link between application data and EDI data. When you need to refer to databank records, user reference information will be helpful in locating a specific record.

### Wrap

The process of removing all trailing spaces between EDI segments. That is, the start of each new segment immediately follows the terminator of the previous segment. The Gentran programs that wrap segments are the Outbound Editor (EBDI002) and the Outbound Databank Extract (EDID205).

## Reserved Word Constants

### **Inbound and Outbound:**

The following reserved word constants are available for use in both inbound and outbound map fields:

### **CONSTANT-01 through CONSTANT-99**

These constants can be used on maps to hold alphanumeric information that is needed later in the map, either to be moved to an output field or used in a conditional statement. Generally, the constants are used for:

1. Loading common data into each application record during an inbound map (e.g., a sort key).
2. Loading input header information into output detail or summary areas.

The following descriptions discuss unique characteristics about the constant field ranges:

**CONSTANT-01 through CONSTANT-40**

These constants are initialized to spaces at the beginning of each document generated (outbound) or document read (inbound).

**CONSTANT-41 through CONSTANT-45**

These constants are initialized to spaces at the beginning of each functional group generated (outbound) or functional group read (inbound).

**CONSTANT-46 through CONSTANT-50**

These constants are initialized to spaces at the beginning of each interchange generated (outbound) or interchange read (inbound).

**CONSTANT-51 through CONSTANT-55**

These constants are initialized to spaces at the beginning of the mapping run only.

**CONSTANT-56 through CONSTANT-99**

These constants are initialized to spaces at the beginning of each document generated (outbound) or document read (inbound).

**HASH-TOTAL-01 through HASH-TOTAL-20**

Hash totals can be used on maps to accumulate predefined input fields that are needed later in the map, either to be moved to an output field or used in a conditional statement. Generally, hash totals are used in maps to accumulate numeric field values (e.g., quantity or dollars), and map the accumulated total into a control total field.

The following descriptions discuss unique characteristics about the hash total field ranges:

**HASH-TOTAL-01 through HASH-TOTAL-05**

These hash totals are initialized to zeros at the beginning of each document that is generated (outbound), or each document that is read (inbound).

**HASH-TOTAL-06 through HASH-TOTAL-10**

These hash totals are initialized to zeros at the beginning of each functional group that is generated (outbound), or each functional group that is read (inbound).

**HASH-TOTAL-11 through HASH-TOTAL-15**

These hash totals are initialized to zeros at the beginning of each interchange that is generated (outbound), or each interchange that is read (inbound).

**HASH-TOTAL-16 through HASH-TOTAL-20**

These hash totals are initialized to zeros only at the beginning of the mapping run.

**ACCUMULATOR-01 through ACCUMULATOR-20**

Accumulators can be used on maps to maintain counters that are incremented by one (1) that are needed later in the map, either to be moved to an output field or used in a conditional statement. Generally, accumulators are used to:

- (1) Count the occurrence of a specific entity such as a line item.
- (2) Generate increasing/sequential record/line item numbers.

The following descriptions discuss unique characteristics about the accumulator field ranges:

### **ACCUMULATOR-01 through ACCUMULATOR-05**

These accumulators are initialized to zeros at the beginning of each document that is generated (outbound), or each document that is read (inbound).

### **ACCUMULATOR-06 through ACCUMULATOR-10**

These accumulators are initialized to zeros at the beginning of each functional group that is generated (outbound), or each functional group that is read (inbound).

### **ACCUMULATOR-11 through ACCUMULATOR-15**

These accumulators are initialized to zeros at the beginning of each interchange that is generated (outbound), or each interchange that is read (inbound).

### **ACCUMULATOR-16 through ACCUMULATOR-20**

These accumulators are initialized to zeros only at the beginning of the mapping run.

## **CURRENT-DATE**

This constant contains the current system date. This field may be mapped to any of the valid date formats (both six and eight character).

- If the application field is defined as 'AN', then only 6 bytes are mapped, regardless of the length of the field.
- If the application field is a valid 8-digit date type (CD, CM, CY, J8, ZD, ZJ, ZM, or ZY), the full 8 digits are mapped.
- If the application field is a valid 6-digit date type (DD, JD, MM, PD, PJ, PM, PY, or YY) only 6 digits are mapped.  
In this case, to get a YYYYMMDD format from CURRENT-DATE, the application field must be defined as 'CY'.

## **CURRENT-TIME**

This constant contains the current system time.

## **TOTAL-SEGMENTS**

This constant contains either the number of segments read (inbound) or the number of segments written (outbound). It is available to be moved into a numeric application field (inbound) or a numeric EDI element (outbound).

**Caution:** On inbound maps only, segments defined to the map are counted and the count is only valid after all segments have been read. For this reason, this constant should only load an application field defined in the summary section of the application. On outbound maps this constant does not contain a valid value if more than one (1) segment is generated in the summary section of the EDI document. Envelopes are not counted in either direction.



### **USER-FIELD**

#### **USER-FIELD-MAP**

These constants contain application link information. These alphanumeric 80-byte application fields can be used to pass application information to the Outbound EDI Databank Statistics file. Application link information is identified in the application with an application field type of UF. The USER-FIELD reserved word constant creates application link information only. The USER-FIELD-MAP reserved word constant creates application link information and maps the application field to the corresponding outbound EDI element.

### **TOTAL-ITEMS**

This constant contains the number of line items generated for an application (inbound) or an EDI document (outbound). A line item is defined as the segments or records in the detail section of the application or EDI document. It is available to be moved into a numeric application field (inbound) or a numeric EDI element (outbound). This constant should only be used in the summary section of the EDI document or application.

#### **Inbound Only:**

The following reserved word constants are all preset by the inbound mapping program. They are available to be moved into an appropriate application field.

### **SENDER-ID**

This constant contains the resolved sender ID (partner cross reference is used) as determined by the inbound editor.

### **SENDER-ID-Q**

This constant contains the resolved sender ID qualifier, if used, (cross reference is used) as determined by the inbound editor.

### **RECEIVER-ID**

This constant contains the resolved receiver ID (partner cross reference is used) as determined by the inbound editor.

### **RECEIVER-ID-Q**

This constant contains the resolved receiver ID qualifier, if used, (cross reference is used) as determined by the inbound editor.

### **STATUS-ACKNOWLEDG**

This constant contains the two-character status code from the Inbound Editor (EBDI001). This constant shows the status of the generated outbound acknowledgement associated with the current incoming document. This constant declares the generation status for the outbound acknowledgement.

### **STATUS-DOCUMENT**

This constant contains the two-character status code from the Inbound Editor (EBDI001). This constant shows the status of the incoming document after compliance checking is complete.

### **STATUS-ENVELOPE**

This constant contains the two-character status code from the Inbound Editor (EBDI001). This constant shows the status of the envelope compliance for incoming control header and trailers associated with the incoming document.

### **STATUS-MAPPING**

This constant contains the highest mapping status value from the Inbound Mapper. This constant is set by the Inbound Mapper within internal error processing. This constant is updated as errors are generated by the Inbound Mapper (the highest error level is indicated).

### **STATUS-SECURITY**

This constant contains the two-character status code from the Inbound Editor (EBDI001). This constant shows the status of the BG PARTNER communication verification. The sender's communication identification is used.

### **STATUS-SEQUENCE**

This constant contains the two-character status code from the Inbound Editor (EBDI001). This constant shows the status of PARTNER SEQUENCING with incoming control numbers.

### **PART-TEST-FLAG**

This constant contains a 'P' if the transaction record on the partner file for the sender has a Production status. If the trading partner is not on file or the transaction record found is not set to production status, this constant contains a value of 'T.'

### **PROD-TEST-FLAG**

This constant contains a P if the transaction status (in Mapping Maintenance) is set to Production. This constant contains a T if the transaction status (in Mapping Maintenance) is set to Test.

### **ISA01 through ISA16**

The constants contain the values of each element received in the ISA segment.

### **ICS01 through ICS12**

The constants contain the values of each element received in the ICS segment.

### **BG01 through BG07**

The constants contain the values of each element received in the BG segment.

### **UNA01 through UNA06**

The constants contain the values of each element received in the UNA segment.

### **UNB01 through UNB18**

The constants contain the values of each element received in the UNB segment.

### **GS01 through GS08**

The constants contain the values of each element received in the GS segment.

### **UNG01 through UNG13**

The constants contain the values of each element received in the UNG segment.

**ST01 through ST02**

The constants contain the values of each element received in the ST segment.

**UNH01 through UNH09**

The constants contain the values of each element received in the UNH segment.

**SCH01 through SCH05**

The constants contain the values of each element received in the SCH segment.

**STX01 through STX12**

The constants contain the values of each element received in the STX segment.

**BAT01 through BAT02**

The constants contain the values of each element received in the BAT segment.

**MHD01 through MHD03**

The constants contain the values of each element received in the MHD segment.

## Application Data Types

When defining an application field, the length of the field and the field data type must be specified. The length of the field is the physical field length or the number of bytes of storage the data field occupied in the file. The data type of the field depends on the use and format of the field.

**Note:** The value is not always the same as its logical length.

### Alphanumeric Data Types

The alphanumeric data types are designed to describe fields that hold free-form data that contains any characters (letters, numbers, and special characters) including spaces. Data is moved to and from these fields, character-by-character, without editing. The data in the receiving field is left-justified. During inbound processing, these fields are initialized to spaces. Fields are truncated because of field/element length considerations without causing an error condition.

If data is moved from a numeric EDI element to an alphanumeric field during an inbound map process, the numeric EDI element is moved to the application field starting with the first significant digit of the number. Any real or implied decimal points are shown in the data as a real decimal point.

For example:

- (1) EDI alphanumeric (length 35) to Application alphanumeric (length 20)  
AB CORPORATION - BILLING DEPARTMENT =====>  
AB CORPORATION - BIL
- (2) EDI numeric (type N2, length 6) to Application alphanumeric (length 8)

123456 (logically 1234.56) =====>

1234.56<space><space>

### **AN (Alphanumeric)**

This is the data type used to describe 99.9 percent of the alphanumeric fields in applications.

For AN data types, the length of the field is the number of characters to be written in the field.

### **AS (Application Select)**

This data type describes a field with the same data characteristics as AN. The difference is that this field also tells the mapper to select appropriate maps based on the value of the data contained in this field. Only one application select field should be coded for an application definition.

## Numeric Data Types

The numeric data types are designed to describe application fields that contain numeric data. Unlike the alphanumeric data types, which are interchangeable, the numeric types have very specific formats that must be adhered to. The numeric application data can be mapped to and from an alphanumeric EDI element.

### **B (Binary)**

This data type supports full word, positive, binary numbers for outbound applications only. The field lengths specified must be in full-word increments (e.g., 2 bytes, 4 bytes, 6 bytes, and 8 bytes). These data types are represented in COBOL as PIC 9(#) COMP, where # is a multiple of 4.

### **N# (Signed Numeric; # = number of implied decimal positions. Valid values are N0 through N9)**

This data type is similar to the EDI N data type except that the entire length of the field must be filled with numeric values. Its length is defined as the number of digits in the number. Negative numbers have an explicit minus sign (-) in the first byte. It is recommended that this data type be used for only data values that is always positive. For example, a 6-position number with two implied decimal positions is defined to the application as data type N2 with a length of 6. It is represented in a COBOL program as PIC 9(4)V9(2).

### **ND (Numeric Display – Outbound Applications Only)**

This data type is similar to the R data type except that any special characters encountered when translating from this field are ignored. This data type was developed to allow limited numeric processing capabilities when the application file is in report format. These fields are converted as shown in the following example. (Assume moving to R type EDI element.)

(1) \$ 120.00=====>120

(2) \$\*\*\*\*\*1.00=====>1

(3) - 1.2=====>-1.2

(4) 1.2 CR=====>-1.2

Remember, this data type can be used only in outbound application definitions.

**P# (Signed Packed; # = number of implied decimal positions. Valid values are P0 through P9)**

This data type is commonly called packed decimal. Its length is defined as the number of digits in the number divided by 2, plus 1. The sign is represented in the last half-byte of the number (D-negative; C or F-positive). A 6-position number with two implied decimal positions would be defined to the application as data type P2 with a length of 4. It is represented in a COBOL program as PIC S9(4)V9(2) COMP-3.

**R Real**

This data type is similar to the EDI R data type except that the entire length of the field must be filled with numeric values. Its length is defined as the number of digits in the number. Negative numbers have an explicit minus sign (-) in the first byte. Decimal positions are indicated with a real decimal point. It is recommended that this data type be used only when absolutely necessary because there is a considerable amount of programming required to generate or interpret this data type compared to the other types. A 6-position number with two positions after the decimal is displayed as 1234.56. For inbound, the field is padded on the right with trailing zeros. As a result, the number one (1) is displayed in a 6-position application field as 1.0000.

**S# (Signed Numeric; # = number of implied decimal positions. Valid values are S0 through S9)**

This data type is commonly called zoned decimal. Its length is defined as the number of digits in the number. The sign is represented by an overpunch in the last byte of the number. A six-position number with two implied decimal positions is defined to the application as data type S2, with a length of 6. It is represented in a COBOL program as PIC S9(4)V9(2).

**U# (Unsigned Packed; # = number of implied decimal positions. Valid values are U0 through U9)**

This data type is the same as the packed data type (P#) above except that the last half-byte of the number that contains the sign is always an F. A 6-position number with two implied decimal positions is defined to the application as data type U2 with a length of 4. It is represented in a COBOL program as PIC 9(4)V9(2) COMP-3.

**Date Data Types:**

<b>CD</b>	Date is in DDMMYYYY format. Field is not packed. The length is 8 bytes.
<b>CM</b>	Date is in MMDDYYYY format. Field is not packed. The length is 8 bytes.
<b>CY</b>	Date is in YYYYMMDD format. Field is not packed. The length is 8 bytes.
<b>DD</b>	Date is in DDMMYY format. Field is not packed. The length is 6 bytes.
<b>J8</b>	Date is in YYYYDDD format. Field is not packed. The length is 7 bytes.
<b>JD</b>	Date is in YYDDD (Julian) format. Field is not packed. The length is 5 bytes.
<b>MM</b>	Date is in MMDDYY format. Field is not packed. The length is 6 bytes.
<b>PD</b>	Date is in DDMMYY format. Field is packed. The length is 4 bytes.
<b>PJ</b>	Date is in YYDDD (Julian) format. Field is packed. The length is 3 bytes.
<b>PM</b>	Date is in MMDDYY format. Field is packed. The length is 4 bytes.
<b>PY</b>	Date is in YYMMDD format. Field is packed. The length is 4 bytes.
<b>T6</b>	Time is in HHMMSS format. Field is not packed. The length is 6 bytes.
<b>T8</b>	Time is in HHMMSSHH format. Field is not packed. The length is 8 bytes.

<b>TM</b>	Time is in HHMM format. Field is not packed. The length is 4 bytes.
<b>YY</b>	Date is in YYMMDD format. Field is not packed. The length is 6 bytes.
<b>ZD</b>	Date is in DDMMYYYY format. Field is packed. The length is 5 bytes.
<b>ZJ</b>	Date is in YYYYDDD format. Field is packed. The length is 4 bytes.
<b>ZM</b>	Date is in MMDDYYYY format. Field is packed. The length is 5 bytes.
<b>ZY</b>	Date is in YYYYMMDD format. Field is packed. The length is 5 bytes.

## EDI Data Types

The following data types are used to describe all of the elements in an EDI document.

### Alphanumeric Data Types

#### **AN    Alphanumeric**

The Alphanumeric data type is used to describe most of the alphanumeric elements in EDI documents. The data in AN field types is free-form text.

#### **ID    Identification**

The Identification data type describes a data field that is alphanumeric, but may contain only specified code values. If the number of these code values has a realistic, finite limit (e.g., unit of measure), then the values in the fields are validated during compliance checking. If the number of valid values has no real defined limit (e.g., DUNS number), then the value is not validated.

### Numeric Data Types

#### **N#    Signed Numeric(# = number of implied decimal positions. Valid values are N0 through N9)**

The Numeric data type is defined as having an implied decimal point and an explicit minus (-) sign, if the value is negative. The value of 1.23 in an N3 element, with a minimum length of one, is 1230.

#### **R    Real**

#### **R# (Real; # = number of decimal positions. Valid values are R0 through R9)**

This data type is defined as having a real decimal point and an explicit minus sign, if the value is negative. The value of 1.23 in an "R" element, with a minimum length of one, is 1.23.

### Date Data Types

#### **D8    Date**

Date is in YYYYMMDD format. Field is not packed. The length is eight bytes.

#### **DT    Date**

Date is in YYYYMMDD or YYMMDD depending on standard definition.

#### **T6    Time**

Time is in HHMMSS format. Field is not packed. The length is six bytes.

**T8 Time**

Time is in HHMMSSHS format. Field is not packed. The length is eight bytes.

**TM Time**

Time is in HHMM format. The length is four bytes.





# Index

## **Symbols**

---

(EBDI032) 4-126

## **Numerics**

---

5/3 NUMBER Record 3-34

5/4 NUMBER Record 3-35

6/3 NUMBER Record 3-34

## **A**

---

ACCUMULATOR-## G-12

acknowledgment

*see* Glossary

Acknowledgment Reconciliation file (EDIDAR)  
8-3

Acknowledgment Reconciliation Monitor  
program (EDID510) 5-5

acknowledgment, functional

*see* Glossary

add-on products

*see* Gentran:Control, Gentran:Dataguard,  
Gentran:Examiner, Gentran:Plus,  
Gentran:Realtime, Gentran:Structure  
Gentran:Viewpoint in Glossary

addressing method

*see* Glossary

alphanumeric data types G-16, G-19

AN data type G-16, G-19

application

*see* Glossary

application data ID

*see* Glossary

application data types G-16 to G-18

application databank

*see* Glossary

Application Databank Inquiry report (EDID551)  
5-8

application definition

*see* Glossary

Application Definition Maintenance program  
(EDIG100) 4-12

application ID

*see* Glossary

archive

*see* Glossary

AS data type G-16

## **B**

---

B data type G-17

BAT## G-15

Batch Application Sequence Renumbering  
program (EBDI090) 4-18

Batch Mapping Report Summary (EBDI061)  
7-59

Batch Partner Profile file maintenance

Batch Partner File Maintenance (EBDI005)  
parameters 9-12

Batch Partner Profile maintenance

batch partner cross reference table 9-3  
Batch Partner Profile file (EBDI005) 9-1  
Control records 9-18  
defined 9-3  
files used 9-14  
Group records 9-67  
Header record 9-16  
PARTIN record 9-23  
PARTMISC record 9-16, 9-18  
partner profile setup 9-3  
PARTOUT record 9-27  
Transaction records 9-88

batch processing

*see* Glossary

Batch Standards Optimization program  
(EBDI004) 4-151

Batch Transaction Mapping Definition Re-Synch  
program (EBDI089) 4-151

BG## G-15

**C**

---

CD data type G-18  
Change Audit Databank Archive file  
(EDICAAR) 8-4  
change audit, databank  
*see* Glossary  
CM data type G-18  
COBOL Copybook Generator program  
(EDIG110) 4-30  
Commerce:Network Response Processor  
program (EDID800) 6-2  
compress  
*see* Glossary  
condition code  
*see* Glossary  
conditional code  
*see* Glossary  
configuration, databank  
*see* Glossary  
CONSTANT-## G-11  
control number  
*see* Glossary  
Control records 9-18  
Copy Application Definition Program  
(EBDI060) 4-7  
Copy Partners program (EBDI087) 4-110  
Copy Transaction Definition program (EBDI080)  
4-146  
Copy Transaction Table program (EBDI076)  
4-36  
CROSSREF (Cross-Reference) record 9-127  
CURRENT-DATE G-12  
CURRENT-TIME G-13  
CY data type G-18

**D**

---

D8 data type G-19  
data dictionary  
*see* Glossary  
data element  
*see* Glossary  
data element dictionary  
*see* Glossary

data element separator  
*see* Glossary  
data element, simple  
*see* Glossary  
data ID, application  
*see* Glossary  
Data Router (EBDI095) program 4-49  
data segment  
*see* Glossary  
DATA SPLIT record  
record layout 9-121  
data types  
alphanumeric G-16, G-19  
application G-16 to G-18  
date G-18, G-19  
EDI G-19  
numeric G-16 to G-18, G-19  
databank  
Acknowledgment Reconciliation file (EDIDAR)  
8-3  
Acknowledgment Reconciliation/Monitor  
program (EDID510)  
Application Databank Inquiry report (EDID551)  
5-8  
Change Audit Databank Archive file (EDICAAR)  
Databank Change Audit Inquiry program  
(EDID500) 5-27  
Databank Change Audit Maintenance program  
(EDID502) 5-31  
databank files 8-1  
EDI Archive Inquiry report (EDID555) 5-19  
EDI Databank Inquiry report (EDID550) 5-37  
EDI Databank Orphan Record Cleanup program  
(EDIG300) 5-109  
EDI Databank Statistics program (EDISTX)  
error rejection 2-22, 3-88  
files used in inbound processing 2-73  
Formatted Network file (EDIFNF) 8-5  
Inbound Application Databank Archive file  
(EDIIAAR) 8-6  
Inbound Application Databank Change Audit file  
(EDIACA) 8-7  
Inbound Application Databank Directory file  
(EDIIAA) 8-9  
Inbound Application Databank Extract program  
(EDID405) 5-59  
Inbound Application Databank Maintenance  
program (EDID401) 5-66  
Inbound Application Databank Message Store file  
(EDIIAAS) 8-11  
Inbound Application Databank Pending file  
(EDIIAAP) 8-13

- Inbound Application Databank Transaction Queue file (EDIQ095) 8-15
- Inbound EDI Databank Archive file (EDIIEAR) 8-17
- Inbound EDI Databank Change Audit file (EDIIECA) 8-18
- Inbound EDI Databank Directory file (EDIIEA) 8-20
- Inbound EDI Databank Link file (EDIIEL) 8-22
- Inbound EDI Databank Maintenance program (EDID301) 5-73
- Inbound EDI Databank Message Store file (EDIIES) 8-24
- Inbound EDI Databank Pending file (EDIIEP) 8-27
- Inbound EDI Databank Transaction Queue file (EDIQ097) 8-28
- Inbound Processing 2-36
- Network Reconciliation file (EDINRC) 8-30
- Network Response file (EDINR) 8-31
- online databank maintenance 5-4
- Outbound Application Databank Archive file (EDIOAAR) 8-32
- Outbound Application Databank Change Audit file (EDIOACA) 8-33
- Outbound Application Databank Directory file (EDIOAA) 8-35
- Outbound Application Databank Link file (EDIOAL) 8-35
- Outbound Application Databank Maintenance program (EDID101) 5-80
- Outbound Application Databank Message Store file (EDIOAS) 8-39
- Outbound Application Databank Pending file (EDIOAP) 8-41
- Outbound Application Databank Transaction Queue file (EDIQ093) 8-42
- Outbound EDI Databank Archive file (EDIOEAR) 8-44
- Outbound EDI Databank Change Audit file (EDIOECA) 8-45
- Outbound EDI Databank Directory file (EDIOEA) 8-47
- Outbound EDI Databank Extract program (EDID205) 5-87
- Outbound EDI Databank Maintenance program (EDID201) 5-99
- Outbound EDI Databank Message Store file (EDIOES) 8-50
- Outbound EDI Databank Pending file (EDIOEP) 8-54
- Outbound EDI Databank Transaction Queue file (EDIQ091) 8-54
- Outbound Processing 3-18
- processing 3-98
- processing option - databanking 2-66
- Reprocess Work file (EDIRW) 8-56
- Transaction Queue Initialize program (EDID520) 5-107
- utilities and reports 5-1
- databank change audit
  - see* Glossary
- Databank Change Audit Inquiry (EDID500) program 5-27
- Databank Change Audit Maintenance (EDID502) program 5-31
- databank configuration
  - see* Glossary
- databank directory
  - see* Glossary
- databank envelope reference ID
  - see* Glossary
- databank inquiry, EDI
  - see* Glossary
- databank message store
  - see* Glossary
- databank run number
  - see* Glossary
- databank, application
  - see* Glossary
- databank, definition
  - see* Glossary
- databank, EDI
  - see* Glossary
- date data types G-18, G-19
- DD data type G-18
- decompress
  - see* Glossary
- deferred enveloping
  - calculating space requirements 8-48, 8-51
  - Control record processing 3-96
  - data compression 3-101
  - databank processing 3-98
  - input files 5-88
  - Outbound EDI Databank Extract (EDID205) program 5-87
  - output files 5-89
  - output processing 3-99
  - parameters 3-98
  - processing method 3-91
  - restrictions 3-91
  - segment processing 3-97
  - valid criteria values 5-94, 5-96, 5-97, 5-103

definition, application

*see* Glossary

delimiter

*see* Glossary

Determining B-1

Determining your DASD Requirement B-1

directory, databank

*see* Glossary

distribution

*see* Glossary

document

*see* Glossary

Document Compliance 3-84

document reference, user

*see* Glossary

document status

*see* Glossary

document, logical

*see* Glossary

DT data type G-19

## E

---

EBDI001 2-9

*see also* Inbound Editor program

EBDI002 3-77

*see also* Outbound Editor program

EBDI004 4-151

*see also* Batch Standards Optimization program

EBDI005 9-1

*see also* Batch Partner Profile file

EBDI006 7-82

*see also* Partner File Print Report program

EBDI010 2-7

*see also* Inbound Unwrap Processing program

EBDI010A 2-7

*see also* Inbound Unwrap Processing program

EBDI011A 3-63

*see also* Outbound Envelope Generation program

EBDI011L 3-63

*see also* Outbound Envelope Generation program

EBDI011M 3-63

*see also* Outbound Envelope Generation program

EBDI015 2-55

*see also* Inbound Split program

EBDI016 3-119

*see also* Outbound Split program

EBDI017 7-2

*see also* Received Acknowledgment Report program

EBDI018 7-2

*see also* Received Acknowledgment Report program

EBDI019 7-19

*see also* EDI Data Print Report program

EBDI032 4-126

*see also* Standards Copy Utility

EBDI036 7-116

*see also* EDI Standard Report in X12 Format program

EBDI038 4-135

*see also* Standards Update program

EBDI039 4-135

*see also* Standards Maintenance program

EBDI041 2-62

Parameters File Layout - Inbound 2-74

*see also* Inbound Data Mapping program

EBDI042 3-5

CNTL Record Layout 3-39, 3-40

Enveloping Parameter Records (SYS000) 3-33

Instream Parameter Record Layout 3-37

Mapping Parameter Records (SYS001) 3-29

Outbound Control Totals File (SYS007) 3-56

*see also* Outbound Data Mapping program

User Exit Call Blocks 3-42

EBDI053 7-42

*see also* Mapping Report (Application Sequence) program

EBDI060 4-7

*see also* Copy Application Definition From One Set of VSAM Files to Another program

EBDI076 4-36

*see also* Copy Transaction Table From File to File program

EBDI080 4-146

*see also* Copy Transaction From One Set of VSAM Files to Another program

EBDI087 4-110

*see also* Copy Partners From One File to Another program

EBDI088 4-80

*see also* Version Comparison and Conversion Report program

- 
- EBDI089 4-151  
*see also* Batch Transaction Mapping Definition Re-Synch program
- EBDI091 7-32  
*see also* Mapping Cross-Reference Report program
- EBDI095 4-49  
*see also* Data Router program
- EBDI110 4-122  
*see also* Sort Generated Acknowledgments program
- EDELMT  
*see also* Online Log File Utility program
- EDI  
*see* Glossary
- EDI Archive Inquiry report (EDID555) 5-19
- EDI Data Print Report (EBDI019) 7-19
- EDI Data Print report (EBDI019) 7-19
- EDI data types G-19
- EDI databank  
*see* Glossary
- EDI databank inquiry  
*see* Glossary
- EDI Databank Inquiry report (EDID550) 5-37
- EDI Databank Orphan Record Cleanup program (EDIG300) 5-109
- EDI Databank Statistics file (EDISTX) 5-53
- EDI Standard Report (EBDI036) program 7-116
- EDIACA 8-33  
*see also* Outbound Application Databank Change Audit file
- EDICAAR 8-4  
*see also* Change Audit Databank Archive file
- EDID101 5-80  
*see also* Outbound Application Databank Maintenance program
- EDID201 5-99  
*see also* Outbound EDI Databank Maintenance program
- EDID205 5-87  
*see also* Outbound EDI Databank Extract program
- EDID301 5-73  
*see also* Inbound EDI Databank Maintenance program
- EDID401 5-66  
*see also* Inbound Application Databank Maintenance program
- EDID405 5-59  
*see also* Inbound Application Databank Extract program
- EDID500 5-27  
*see also* Databank Change Audit Inquiry program
- EDID502 5-31  
*see also* Databank Change Audit Maintenance program
- EDID510 5-5  
*see also* Acknowledgment Reconciliation/Monitor program
- EDID520 5-107  
*see also* Transaction Queue Initialize program
- EDID550 5-37  
*see also* EDI Databank Inquiry Report program
- EDID551 5-8, 5-19  
*see also* Application Databank Inquiry Report program
- EDID800 6-2  
*see also* COMMERCE:Network Response Processor program
- EDID805 6-6  
*see also* GEIS Response Processor program
- EDID810 6-7  
*see also* IBM IN Response Processor program
- EDID825 6-14  
*see also* Universal Response Processor program
- EDID850 6-9  
*see also* Network Reconciliation program
- EDID860 6-11  
*see also* Network Reconciliation Maintenance program
- EDIDAR 8-3  
*see also* Acknowledgment Reconciliation file
- EDIELMT 4-102
- EDIFNF 8-5  
*see also* Formatted Network file
- EDIG100 4-12  
*see also* Application Definition Maintenance program
- EDIG110 4-30  
*see also* COBOL Copybook Generator program
- EDIIAA 8-9  
*see also* Inbound Application Databank Directory file
- EDIIAAP 8-13  
*see also* Inbound Application Databank Pending file
-

## Index

---

- EDIIAAR 8-6
  - see also* Inbound Application Databank Archive file
- EDIIAAS 8-11
  - see also* Inbound Application Databank Message Store file
- EDIIACA 8-7
  - see also* Inbound Application Databank Change Audit file
- EDIIEA 8-20
  - see also* Inbound EDI Databank Directory file
- EDIIEAR 8-17
  - see also* Inbound EDI Databank Archive file
- EDIIECA 8-18
  - see also* Inbound EDI Databank Change Audit file
- EDIIEL 8-22
  - see also* Inbound EDI Databank Link file
- EDIIEP 8-27
  - see also* Inbound EDI Databank Pending file
- EDIIES 8-24
  - see also* Inbound EDI Databank Message Store file
- EDINR 8-31
  - see also* Network Response file
- EDINRC 8-30
  - see also* Network Reconciliation file
- EDIOAA 8-35
  - see also* Outbound Application Databank Directory file
- EDIOAAR 8-32
  - see also* Outbound Application Databank Archive file
- EDIOAL 8-37
  - see also* Outbound Application Databank Link file
- EDIOAP 8-41
  - see also* Outbound Application Databank Pending file
- EDIOAS 8-39
  - see also* Outbound Application Databank Message Store file
- EDIOEA 8-47
  - see also* Outbound EDI Databank Directory file
- EDIOEAR 8-44
  - see also* Outbound EDI Databank Archive file
- EDIOECA 8-45
  - see also* Outbound EDI Databank Change Audit file
- EDIOEP 8-53
  - see also* Outbound EDI Databank Pending file
- EDIOES 8-50
  - see also* Outbound EDI Databank Message Store file
- EDIQ091 8-54
  - see also* Outbound EDI Databank Transaction Queue file
- EDIQ093 8-42
  - see also* Outbound Application Databank Transaction Queue file
- EDIQ095 8-15
  - see also* Inbound Application Databank Transaction Queue file
- EDIQ097 8-28
  - see also* Inbound EDI Databank Transaction Queue file
- EDIRW 8-56
  - see also* Reprocess Work file
- EDISTX 5-53
  - see also* EDI Databank Statistics program
- Editor
  - see* Glossary
- electronic data interchange
  - see* EDI
- element
  - see* Glossary
- End of Transmission
  - see* EOT
- envelope
  - see* Glossary
- Envelope Generation program 3-6
- envelope reference ID, databank
  - see* Glossary
- EOT
  - see* Glossary
- Error Message and Control File 3-85
- Error Message Maintenance 4-68
- Error Rejection
  - error numbers 2-22, 3-89
  - error rejection numbers, inbound 2-22
  - error rejection numbers, outbound 3-89
  - error rejection record 9-124
  - Error Rejection with the Databank global parameter 3-88
  - partner error control flag, global parameters file 3-86
  - processing, inbound 2-19
  - processing, outbound 3-85

Errors Encountered Processing Partner Data  
report 9-132

## F

---

field

*see* Glossary

files, databank 8-1

Formatted Network file (EDIFNF) 8-5

functional acknowledgment

*see* Glossary

## G

---

GEIS Response Processor (EDID805) program  
6-6

Gentran:Control

files updated by Inbound Application Databank

Extract program (EDID405) 5-59, 5-63

files written to by Outbound EDI Databank

Extract program (EDID205) 5-89, 5-94

Online Log File Utility program usage 4-102

*see* Glossary

Gentran:Plus

associating partner with communications profile  
9-21

associating partner with network tracking 9-21

CONTROL record 3-74, 3-75

field on CNTL record layout 3-39, 3-41

field on user exit call block 2-84

Inbound Split program processing control

parameter 2-56

input from Outbound Mapper 3-24

INS Relationship Control 9-9

INS-TRADANET network status information  
5-37

output from Outbound Split program 3-121

parameters for Outbound EDI Databank Extract  
program (EDID205) 5-93, 5-94

PLUS record 3-35, 3-72

processing option for Outbound Mapper 3-23

processing options for Outbound Mapper 3-23,  
3-27

*see* Glossary

VSAM Batch Status flags 5-53

Gentran:Realtime

Online Log File Utility program usage 4-102

*see* Glossary

Gentran:Structure

data routing with 4-50

Envelope Format field 7-46, 7-55, 7-70

envelope format field 7-63

*see* Glossary

Segment ID Length field 7-47, 7-56, 7-64, 7-71

Segment ID Start field 7-46, 7-55, 7-64, 7-70

user-defined envelope type 7-46, 7-55, 7-63,  
7-70

Gentran:Viewpoint

Exception Management performed at group level  
9-70, 9-91

Exception Management performed at interchange  
level 9-21

Exception Management performed at transaction  
level 9-90

field on Parameters File (Inbound) 2-76

files for Inbound Mapper 2-74

files for Outbound Mapper 3-27

files used with Outbound Editor 3-107

Inbound Application Databank Extract program  
(EDID405) 5-65

Outbound EDI Databank Extract program  
(EDID205) 5-96

processing option for Inbound Mapper 2-71

processing option for Outbound Mapper 3-23

*see* Glossary

Tracking Management performed for partner  
9-22

Viewpoint processing with Outbound Editor  
3-99

Viewpoint tracking of output 3-102

Viewpoint User Tracking record 3-120

Gentran:Viewpoint Exception Management

files used with Outbound Editor 3-107

performed at group level 9-70, 9-91

performed at interchange level 9-21

performed at transaction level 9-90

Gentran:Viewpoint Tracking Management

file for Outbound Mapper 3-27

files used with Outbound Editor 3-107

performed for partner 9-22

processing option for Inbound Mapper 2-71

processing option for Outbound Mapper 3-23

Viewpoint tracking of output 3-102

Viewpoint User Tracking record 3-120, A-52

GENUNH record 9-102

Global Parameters A-1

Editor processing A-2

error rejection setting 3-86

Global Parameter reference tables A-5

## Index

---

Inbound and Outbound parameter record layouts  
A-13  
Inbound parameter record layouts A-25  
introduction A-2  
Outbound parameter record layouts A-49  
Partner acknowledgments A-3  
Partner databank level A-3  
runtime global parameter processing 2-12  
GS Envelopes  
    Generating GS05 3-65  
GS Time Record 3-36  
GS## G-15

## H

---

HASH-TOTAL-## G-11 to G-12  
header  
    *see* Glossary

## I

---

IBM IN Response Processor program (EDID810)  
    6-7  
ICS## G-14  
ID data type G-19  
inbound  
    reserved word constants G-11  
Inbound Application Databank Archive file  
    (EDIIAAR) 8-6  
Inbound Application Databank Change Audit file  
    (EDIACA) 8-7  
Inbound Application Databank Directory file  
    (EDIIAA) 8-9  
Inbound Application Databank Extract program  
    (EDID405) 5-59  
Inbound Application Databank Maintenance  
    program (EDID401) 5-66  
Inbound Application Databank Message Store  
    file (EDIAS) 8-11  
Inbound Application Databank Pending file  
    (EDIAP) 8-13  
Inbound Application Databank Transaction  
    Queue file (EDIQ095) 8-15  
Inbound Data Mapping program (EBDI041)  
    2-62

Inbound EDI Databank Archive file (EDIIEAR)  
    8-17  
Inbound EDI Databank Change Audit file  
    (EDIIECA) 8-18  
Inbound EDI Databank Directory file (EDIIEA)  
    8-20  
Inbound EDI Databank Link file (EDIIEL) 8-22  
Inbound EDI Databank Maintenance program  
    (EDID301) 5-73  
Inbound EDI Databank Message Store file  
    (EDIIES) 8-24  
Inbound EDI Databank Pending file (EDIIEP)  
    8-27  
Inbound EDI Databank Transaction Queue file  
    (EDIQ097) 8-28  
Inbound Editor program (EBDI001) 2-9  
Inbound Processing 2-1  
    Partner Relationship Usage 2-32  
    pass-thru record processing 2-12  
    system flow 2-3  
    unwrap process 2-7  
Inbound Split program (EBDI015) 2-55  
Inbound Unwrap Processing program (EBDI010,  
    EBDI010A) 2-7  
in-house interface  
    *see* Glossary  
Instream Parameter 3-5  
interface, in-house  
    *see* Glossary  
ISA## G-14

## J

---

J8 data type G-18  
JD data type G-18

## L

---

logical document  
    *see* Glossary



---

**M**

---

mandatory  
*see* Glossary

map, transaction  
*see* Glossary

mapping  
*see* Glossary  
 Version Comparison and Conversion Report  
 program (EBDI088) 4-80

Mapping Cross-Reference report (EBDI091)  
 7-32

Mapping Parameter Record Layout #2 3-32

Mapping Parameter Record Layout #3 3-32

Mapping Parameter Record Layout #4 3-33

Mapping Parameters file (SYS001) 3-8

Mapping report (Application Sequence)  
 (EBDI053) 7-42

Mapping report (Standard Sequence) (EBDI052)  
 7-50

Mapping Validation report (EBDI082) 7-66

message  
*see* Glossary

message status level  
*see* Glossary

message store, databank  
*see* Glossary

MHD## G-15

MM data type G-18

---

**N**

---

N# data type G-17, G-19

Name and Address records 9-109

NAMEAD1 record 9-109

NAMEAD2 record 9-111

NAMEAD3 record 9-113

NAMEAD4 record 9-115

NAMEAD5 record 9-117

ND data type G-17

network  
*see* Glossary

Network Reconciliation file (EDINRC) 8-30

Network Reconciliation Maintenance program  
 (EDID860) 6-11

Network Reconciliation program (EDID850) 6-9

Network Response file (EDINR) 8-31

network utilities  
 6-1

numeric data types G-16 to G-18, G-19

---

**O**

---

object envelope processing  
 processing methods 3-95

Online Log File Utility program (EDIELMT)  
 4-102

Online Selective Standards Download program  
 (EBDI030) 4-134

outbound  
 reserved word constants G-11 to G-13

Outbound Application Databank Archive file  
 (EDIOAAR) 8-32

Outbound Application Databank Change Audit  
 file (EDIOACA) 8-33

Outbound Application Databank Directory file  
 (EDIOAA) 8-35

Outbound Application Databank Link file  
 (EDIOAL) 8-37

Outbound Application Databank Maintenance  
 program (EDID101) 5-80

Outbound Application Databank Message Store  
 file (EDIOAS) 8-39

Outbound Application Databank Pending file  
 (EDIOAP) 8-41

Outbound Application Databank Transaction  
 Queue file (EDIQ093) 8-42

Outbound Data Mapping program (EBDI042)  
 3-5

Outbound EDI Databank Archive file  
 (EDIOEAR) 8-44

Outbound EDI Databank Change Audit file  
 (EDIOECA) 8-45

Outbound EDI Databank Directory file  
 (EDIOEA) 8-47

Outbound EDI Databank Extract program  
 (EDID205) 5-87

Outbound EDI Databank Maintenance program  
 (EDID201) 5-99

## Index

---

Outbound EDI Databank Message Store file (EDIOES) 8-50  
Outbound EDI Databank Pending file (EDIOEP) 8-53  
Outbound EDI Databank Transaction Queue file (EDIQ091) 8-54  
Outbound Editor program (EBDI002) 3-77  
Outbound Envelope Generation program (EBDI011M, EBDI011L, EBDI011A) 3-63  
Outbound Processing 3-1  
    Outbound Split program (EBDI016) 3-119  
    sequence of processing 3-79  
    splitting process 3-122  
Outbound Split program (EBDI016) 3-119  
    process illustration 3-122

## P

---

P# data type G-17  
Package Structure Processing 2-35  
partner  
    Partner Relationship Usage 2-32  
Partner File Print report (EBDI006) 7-82  
partner profile 3-86  
    partner profile file  
        batch update parameters 9-12  
    *see* Glossary  
Partner Profile file  
    *see also* Batch Partner Profile file (EBDI005)  
Partner Relationship Usage 2-32  
partner/qualifier mode 2-37  
    databank processing 3-99  
    parameter setting 5-94, 5-105, 5-110  
    *see* Glossary  
PART-TEST-FLAG G-14  
PD data type G-18  
PJ data type G-18  
PLUS Record 3-35  
PM data type G-18  
processing  
    inbound unwrap process 2-7  
    sequence of outbound processing 3-79  
processing option  
    Gentran:Viewpoint 3-23

Truncated Sender's Reference (TRADACOMS)-  
Gentran:Plus only 3-23  
processing, batch  
    *see* Glossary  
PROD-TEST-FLAG G-14  
PY data type G-18

## R

---

R data type G-17, G-19  
R# data type G-19  
RAIL Documents 3-65  
Received Acknowledgment report (EBDI017,  
    EBDI018) 7-2  
RECEIVER-ID G-13  
RECEIVER-ID-Q G-14  
record  
    *see* Glossary  
reference ID, databank envelope  
    *see* Glossary  
reference, user document  
    *see* Glossary  
relationship mode 2-37  
    databank processing 3-99  
    parameter setting 2-81, 3-45, 5-94, 5-95,  
    5-110  
    *see* Glossary  
release character  
    *see* Glossary  
release indicator  
    *see* Glossary  
RELPART Record 9-129  
report descriptions and layouts 7-1  
reporting programs  
    *see* reports  
reports  
    Batch Mapping Report Summary (EBDI061)  
    7-59  
    EDI Data Print Report (EBDI019) 7-19  
    EDI Standard Report (EBDI036) program 7-116  
    Mapping Cross-Reference Report program  
    (EBDI091)  
    Mapping Report (Application Sequence) program  
    (EBDI053)  
    Mapping report (Standard Sequence) (EBDI052)  
    7-50  
    Mapping Validation report (EBDI082) 7-66  
    Partner File Print report (EBDI006) 7-82

Received Acknowledgment Report program  
(EBDI017,EBDI018)  
Reprocess Work file (EDIRW) 8-56  
reserved word constants G-11  
return code  
    *see* Glossary  
rror 2-22  
run number, databank  
    *see* Glossary  
Runtime parameters 3-6

---

## S

S# data type G-18  
SCH## G-15  
segment  
    *see* Glossary  
segment separator  
    *see* Glossary  
segment terminator  
    *see* Glossary  
selection criteria  
    *see* Glossary  
Selective Standards Download program  
    (EBDI030) 4-131  
SENDER-ID G-13  
SENDER-ID-Q G-13  
Separator Driver program (EBDI930) 4-27  
separator, data element  
    *see* Glossary  
separator, segment  
    *see* Glossary  
separator, subelement  
    *see* Glossary  
simple data element  
    *see* Glossary  
SNRF Record 3-36  
Sort Generated Acknowledgments program  
    (EBDI110) 4-122  
splitting  
    Outbound Split program (EBDI016) 3-119  
    outbound splitting process 3-122  
ST## G-15  
standard Enveloping  
    processing method 3-90

standards  
    copy utility 4-126  
    determining DASD standards space requirement  
    B-1  
    Online Selective Standards Download program  
    (EBDI030) 4-134  
    Selective Standards Download program  
    (EBDI030) 4-131  
    Standards Update program (EBDI038) 4-135  
    Version Comparison and Conversion Report  
    program (EBDI088) 4-80  
Standards Copy Utility (EBDI032) 4-126  
Standards Update program (EBDI038) 4-135  
status level, message  
    *see* Glossary  
status, document  
    *see* Glossary  
STX## G-15  
subelement separator  
    *see* Glossary  
summary area  
    *see* Glossary

---

## T

T6 data type G-18, G-19  
T8 data type G-18, G-19  
TCR Record 3-35  
terminator, segment  
    *see* Glossary  
TM data type G-18, G-19  
TOTAL-ITEMS G-13  
TOTAL-SEGMENTS G-13  
trading mode  
    parameter setting 5-94  
trading partner  
    PARTHDR record 9-16  
    *see* Glossary  
trading partner processing mode  
    databank processing 3-99  
    *see* Glossary  
TRANIN record 9-92  
TRANMISC record 9-88  
TRANOUT record 9-96  
transaction envelope processing  
    processing methods 3-95

## Index

---

transaction map  
    *see* Glossary  
Transaction Queue Initialize program (EDID520)  
    5-107  
transaction set  
    *see* Glossary  
Transaction Totals report 9-131  
turnaround time  
    *see* Glossary

## U

---

U# data type G-18  
UCS NUMBER Record 3-33  
UNA## G-15  
UNB envelopes  
    generating the UNB07 3-65  
UNB## G-15  
UNG## G-15  
UNH## G-15  
Universal Response Processor program  
    (EDID825) 6-14  
unwrap  
    *see* Glossary  
user  
    *see* Glossary  
User Defined Data records 9-119  
user document reference  
    *see* Glossary  
user exits  
    call block layout (Inbound Mapper) 2-77  
    field on Parameters file 2-75  
    Inbound Mapper 2-67  
USER record 9-119  
user reference  
    *see* Glossary  
utilities 4-1

## W

---

wrap  
    *see* Glossary

## Y

---

YY data type G-18

## Z

---

ZD data type G-18  
ZJ data type G-18  
ZM data type G-18  
ZY data type G-18