

White Paper

Binary (BIN) Segment Support

Case # S973599

GENTRAN:Server[®] for AS/400 Product Development

Last revised: 9-07-01

Sterling Commerce
An IBM Company

The information contained in this document represents the current view of Sterling Commerce on the issue discussed as of the date of publication. Because Sterling Commerce must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Sterling Commerce, and Sterling Commerce cannot guarantee the accuracy of any information presented after the date of publication.

This document is for informational purposes only. STERLING COMMERCE MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS DOCUMENT.

© 2001 Sterling Commerce. All rights reserved.



Table of Contents

<i>Table of Contents</i>	2
<i>Introduction</i>	3
<i>General Description</i>	4
<i>Outbound Processing</i>	5
<i>Inbound Processing</i>	9
<i>Programs changed</i>	12



Introduction

The BIN and BDS segments are ASC X12 segments that are used to transfer binary data in a single data segment. Our current processes will only support elements up to 264 bytes in length, and almost all binary documents will be larger than this length. In order to accommodate these two new segments, we have modified our programs to handle the binary elements differently than all others. The Editor programs and Mapper programs now have the ability to handle binary elements of up to 799,997,000 bytes in length.

The major benefit of this enhancement is to provide HIPAA compliance for all ASC X12 documents.

Major enhancement areas:

- New configuration parameters
- PRCTRNOU and PRCTRNI flow modifications
- Editor and Mapper changes



General Description

The BIN segment is an ASC X12 segment that is used to transfer binary data in a single data segment. This segment contains two elements. The first element is numeric and contains the length of the Binary Data, with a minimum of 1 and maximum of 15. The second element contains the Binary data stream. There is no identification of the internal structure of the binary data. This binary data stream can have up to 799,997,000 bytes. For standards version 004030, the BIN segment can be found in the following transactions: 105, 176, 186, 187, 197, 240, 252, 255, 267, 268, 275, 280, 841, 842.

The BDS segment is an ASC X12 segment that is used in the Associated Data Transaction Set (102). This transaction set may be used to convey associated data. Associated data is defined as an object, a set of functionally-related information not using the usual transaction set structure in segments. Objects may include graphic, text, parametric, tabular, image, spectral, audio, etc. data. This transaction set may be included in any functional group within an interchange.

The BDS segment contains three elements. The first contains a Filter ID Code. This is a three byte coded field that specifies the encoding algorithm, if any. The second contains the length of the Binary Data, with a min of 1 and max of 15. The third element contains the binary data stream and can have up to 799,997,000 bytes.

For both the BIN and BDS segments, it is crucial to have an accurate length in the BIN01/BDS02 element. When dealing with these segments, many times there are characters embedded in the binary data that are the same as the segment terminator for the document. In order to process successfully, we need to use the length in the BIN01/BDS02 elements to determine where the end of the binary data falls. If the length is not accurate, our programs will try and process what is recoverable, but many times will have to abort.

Caution

When transmitting and receiving data that contains binary data, it must remain in EBCDIC format. It is not possible to translate the data back and forth between ASCII and EBCDIC, because the binary data cannot be translated.



Outbound Processing

Binary data can be provided to GENTRAN:Server for AS/400 in two different ways. If the binary length is 264 or less, then the binary data can be located in the same file as the rest of the application data and processed as a regular application field. If the length is longer than 264 then the binary data MUST be located in a separate file.

Note

The binary data is not required to be greater than 264 bytes in order to store it in an external file. You can use an external file for any length of data.

Mapping binary data from a regular application field:

In this scenario, normal mapping will take place. The user will have mapped the binary application field to the BIN02/BDS03. If the length is greater than 80 characters, then the BIN02/BDS03 will need to be sub-fielded and the binary application field will need to be broken down accordingly. The user will then map each of these sub-fields.

The length mapped in the BIN01/BDS02 is extremely crucial. It needs to represent the actual number of characters found in the binary data. If this length is inaccurate, you will get undesirable results.

Note

The outbound processing flow in this situation is not altered. The typical flow of data from the mapper into the editor will take place.

Processing binary data from an external file:

When mapping binary data from an external file, you will need to sub-field the BIN02/BDS03 into 4 separate fields of 10 bytes each. In the first field, the literal 'BINARY LOC' must be mapped. The next 3 fields will contain the library, file name, and the member name respectively, identifying where the binary data resides. The member name field must be listed and can contain values of *FIRST, *LAST or a specific member name. The following example illustrates the required mapping:



GENTRAN Group
AS/400 Enhancements

EDIX507 Subfield MAZ 08/30/01
CTL01 16:19:33

Transaction ID	BINARYOUT	Segment ID	BIN
Send or Receive	S	Element ID	0785
Segment Sequence	2500	Element Type	B
Element Sequence	20	Minimum Length	1
Mandatory Code	M	Maximum Length	264
Segment Description	BINARY DATA		
Element Description	BINARY DATA		

Type option, press Enter.
4=Delete

Opt	Seq No	Positions From	To	M	Type	Description
—	22	1	10	M	AN	<u>binary label</u>
—	24	11	20	M	AN	<u>bin library</u>
—	26	21	30	M	AN	<u>bin file</u>
—	28	31	40	M	AN	<u>bin member</u>

More...

F1=Help F4=Prompt F5=Refresh F10=Update F12=Cancel

EDIX511 Element Mapping Outbound 08/30/01

Tran ID	BINARYOUT	S/R	S
Seg ID	BIN BINARY DATA		
Seg Seq	2500	Area Code	1

Type option, press Enter.

1=Appl Fields 11=Ext Map 12=Subfield 13=Codes

Opt	Constant/Field	Table ID	Map	No	ID	Cd	P	Description	C
—	<u>F020.BINLEN</u>		Y	10	784	M	NO	<u>LENGTH OF BINARY DATA</u>	R
—				20	785	M	SF	<u>BINARY DATA</u>	
—	<u>'BINARY LOC'</u>		Y	22		O	AN	<u>binary label</u>	
—	<u>F020.BINLIB</u>		Y	24		O	AN	<u>bin library</u>	
—	<u>F020.BINFIL</u>		Y	26		O	AN	<u>bin file</u>	
—	<u>F020.BINMBR</u>		Y	28		O	AN	<u>bin member</u>	

Bottom

F1=Help F4=Prompt F10=Update F12=Cancel F13=Services F24=More Keys

The length mapped in the BIN01/BDS02 is extremely crucial. It needs to represent the actual number of characters found in the binary file. If this length is inaccurate, you will get undesirable results.



The 'BINARY LOC' literal is a keyword that will trigger the mapper to perform special binary processing. The BIN/BDS segment will be generated containing the filter code (if it is the BDS segment), the length, and the binary data element. The binary data element will contain the values of the 4 sub-fields. Following the BIN segment an additional record will be written. This record will be prefixed with ####BIN followed by the library, file name, and member name of where the binary data resides, length of the binary data, interchange partner id/qual, group partner id/qual, application and map name, version, functional group, transaction set, and interchange, group, and transaction control numbers. All of this detailed information will be provided in this record as a tracking mechanism.

The mapper will produce 4 new error messages when certain situations occur. These error messages are listed below:

Error Problem

- 841 The length found in the BIN01/BDS02 is less than the binary data to be mapped.
 RESULT: This is a 4 level error, causing the binary data to be truncated to the length specified in the BIN01/BDS02.

- 842 The length found in the BIN01/BDS02 is greater than the binary data to be mapped.
 RESULT: This is a 4 level error, causing the BIN02/BDS03 segment to be padded with spaces to match the length in the BIN01/BDS02.

- 843 Incomplete binary data file location. Need library, file, and member name (can map *FIRST or *LAST for member name).
 RESULT: This is an 8 level error, causing the BIN/BDS segment to be dropped.

- 846 The length found in the BIN01/BDS02 is greater than 264 and 'BINARY LOC' was not specified to be mapped in the BIN02/BDS03 element.
 RESULT: This is an 8 level error, causing the BIN/BDS segment to be dropped.

Upon completion of the mapper, the outbound flow will perform a new function if a ####BIN record was generated. A new program, EBDI040, will execute and replace any ####BIN records with the binary data that is held in the file specified. It uses the length mapped in the BIN01/BDS02 to move the appropriate bytes of data into the BIN02/BDS03 from the specified file. A new intermediate file, GT42OAP will contain the ####BIN record and perform as the input to EBDI040. The intermediate file, GT88IAP will serve as the output file from EBDI040 and will contain the actual binary data stream that will be used as the input to the Editor.

The EBDI040 program can produce 5 new error messages if the following instances occur:



<u>Error</u>	<u>Problem</u>
841	Length of binary element is less than actual binary data. RESULT: The binary data will be truncated to match the binary length stated in the BIN01/BDS02 element.
842	Length of binary element is greater than the actual binary data. RESULT: The BIN02/BDS03 will be padded with spaces to match the binary length.
844	Error opening the binary data file. RESULT: There will be a BIN/BDS segment written with no elements.
847	Binary data file was empty. RESULT: There will be a BIN/BDS segment written with no elements.
848	Object not found for LIB/FILE/MBR specified for binary data file. RESULT: There will be a BIN/BDS segment written with no elements.

In all of the error situations above, a BIN/BDS segment will be produced, but there will be no data in the elements.

The Editor will not perform a length validation based upon the standard. It will use the length recorded in the BIN01/BDS02 element to verify the number of bytes in the BIN02/BDS03. If there is a variance in the size specified and the actual bytes of binary data present, an 845 error message will be issued and the outbound process will terminate with a 16 return code.



Inbound Processing

In order to accurately handle the BIN/BDS segment, program modifications were required. The total number of bytes transferred in BIN02/BDS03 can be extremely large and would be quite cumbersome if it were required to be mapped. In many cases this data needs to be extracted from the transaction and placed in an external file in preparation for some other function. This new process allows the binary data stream to be extracted directly into a physical file.

To process the inbound BIN/BDS segment, there are two set-up procedures that should occur prior to performing any processing. There is a new system configuration flag that controls the placement of the binary data stream if the total number of bytes is 264 or less. To set this configuration flag, select option 6 – Set Binary Mapping Option from the GENSYSCFG menu, or prompt the SETBINMAP command. The SETBINMAP command has one prompt and looks as follows:

```
                Set Binary Mapping Option (SETBINMAP)

Type choices, press Enter.

Binary Mapping Option . . . . . > *YES          *YES, *NO

                                                                    Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

When this option is set to *NO, regular application mapping will occur for the binary data, providing the total number of bytes in the BIN02/BDS03 is 264 or less. If the length of the binary string is greater than 264, the binary data string cannot be mapped to an application file, instead it will be placed in a physical file.

When this option is set to *YES, the binary data string will always be placed in a physical file, regardless of the length. *YES is the default for this system flag.



The other set-up function is to specify a library where the external binary data files will reside. To set this configuration flag, select option 5 – Set Binary Mapping Library from the GENSYSCFG menu, or prompt the SETBINLIB command. The SETBINLIB command has one prompt and looks as follows:

```
                Set Binary Data Library (SETBINLIB)

Type choices, press Enter.

Binary Library Name . . . . . > GENBINDTA      Character value

                                                    Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

Enter the name of an existing library where the binary data files should reside. The library name entered will be stored in the data area GENBINLIB that resides in the GENTRAN program library.

Note

If this step is omitted and the library configuration is left blank, the user’s GENTRAN data library will be used.

Mapping binary data into a regular application field:

In this scenario, normal application mapping will take place providing the following 2 requirements are met:

1. The binary data stream is 264 bytes or less.
2. The system configuration flag for the binary mapping option is set to *NO.

The user will map the binary data stream into an application field from the BIN02/BDS03. If the length is greater than 80 characters, then the BIN02/BDS03 will need to be sub-fielded and the binary application field will need to be broken down accordingly. The user will then map each of these sub-fields.



Placing the binary data stream into an external file:

In this scenario, the binary data stream will be placed in an external file providing one of the following two requirements are met:

1. The binary data stream is greater than 264 bytes and the binary mapping option is set to *NO.

or

2. The binary mapping option is set to *YES. (The binary data length has no bearing on the placement of this data.)

Upon encountering a BIN/BDS segment in the inbound Editor, a new program is called that will write the data to an external file. The library to store the binary data (value stored in GENBINLIB) is retrieved and a unique file is created in this library at 512 bytes in length. The file name will be GT9999999, where 9999999 represents a sequential number. The sequential number will be stored in a data area called GENBINSEQ and will reside in the GENTRAN data library.

Once the binary file has been written, the editor will replace the binary data stream with the literal 'BINARY' followed by 10 bytes for the library name, 10 bytes for the file name, and 10 bytes for the member name of where the binary data stream was written. Also a record will be written to a new file - EDI bin master control file (EDIBIN). This file will contain the Partner/qual, interchange, group, and transaction control numbers, message center sequence number, communications profile where the original data resides, batch sequence number, segment terminator, library/filename where the binary data was written, and the line number and position where the binary data starts in the inbound data file (INBDTA) file. It also contains a date/time stamp as to when it was written. This file will contain the necessary information for the customer to locate where their binary data was stored.



Programs changed

- GENBINSEQ** New data area created in the GENTRAN data library.
Data area to hold the incremental sequence number used in the inbound binary file name.
- GENBINLIB** New data area created in the GENTRAN program library.
Data area to hold the library name of where the inbound binary files are stored.
- GT42OAP** New multi-membered file stored in the GENTRAN data library.
Intermediate file used in the outbound process when the ###BIN pass thru records are used.
- EDIBIN** New physical file stored in the GENTRAN data library.
New file used to store the inbound binary file information.
- SYSCFG** System Configuration File
Modified the column heading to use the extra SCFLG03 field to store the inbound mapping flag.
- GT00EMP** Batch Error Message File
Added new batch errors to existing records for record number 840 through 848.
- RMVMBRP** RMVGENMBR work file
Added an entry for the new intermediate file GT42OAP.
- SETBINLIB** New Command stored in the GENTRAN program library.
Command to set the library that stores the inbound binary files.
- SETBINLIB1** New command stored in the GENTRAN program library.



Command used from the GENSYSCFG to set the library that stores the inbound binary files.

SETBINLIB New CL program stored in the GENTRAN program library.
CL used to process the SETBINLIB command.

SETBINLIBV New CL program stored in the GENTRAN program library.
Validity CL used in the SETBINLIB command.

SETBINLIB1 New CL program stored in the GENTRAN program library.
CL used to process the SETBINLIB1 command.

SETBINMAP New Command stored in the GENTRAN program library.
Command to set the inbound binary mapping option.

SETBINMAP1 New command stored in the GENTRAN program library.
Command used from the GENSYSCFG to set the inbound binary mapping option.

SETBINMAP New CL program stored in the GENTRAN program library.
CL used to process the SETBINMAP command.

SETBINMAP1 New CL program stored in the GENTRAN program library.
CL used to process the SETBINMAP1 command

EDCLBIN New CL program stored in the GENTRAN program library.
CL used to write the inbound binary information to a physical file.

EBDI040 New COBOL program stored in the GENTRAN program library.
Program used to remove the ####BIN record during the outbound flow with the actual binary data.

EBDIBIN New COBOL program stored in the GENTRAN program library.



Program used to write the inbound binary information to a physical file.

GENBINSEQ New CL program stored in the GENTRAN program library.
Program the increments the GENBINSEQ data area for the inbound binary file name.

GENSYSCFG System Configuration Menu
Menu program was modified to add the SETBINLIB and SETBINMAP options.

The following programs have been changed to enable this new feature to function successfully:

EDCL007	Process Outbound Job Stream
EDCL032	Outbound Mapping Sequence
CHKOBJCL	Validate various objects
EBDI001	Inbound Editor
EBDI002	Outbound Editor
EBDI041	Inbound Mapper
EBDI042	Outbound Mapper
EBDI078	EDI Data Print *SEG Level;REQUEUE;FNDSTR
EDIX451	Inbound and Outbound Data Queue Inquiry for INT
EDIX452	Inbound and Outbound Data Queue Inquiry for GRP
EDIX453	Inbound and Outbound Data Queue Inquiry for TRN
EDIX454	Inbound and Outbound Segment Inquiry
EDIX637	Auto Notification User Exit Control

The following programs were recompiled as a result of implementing this new feature.

EBDI045 Outbound Mapping - Load Transaction Subroutine

The following panel groups were added/modified for on-line assistance:

RMVGENMBRH
SETBINLIBH
SETBINMAPH