New Features
EDI Products 4.2

Document ID: DC39126-01-0420-05

Last revised: February 14, 2006

This document describes new features for EDI Products 4.2 and subsequent EBF releases for ECMap™, EC Gateway™, and ECRTP™. Each new feature is now tagged with the EBF version it is released in, for ease of reference.

Overview

This document introduces feature enhancements and modifications that are of potential interest to Sybase® customers. Changes include support for new EDI standards, RunMap options, and command line switch parameters.

Before you begin

System requirements

Supported platforms

The following lists include the supported operating systems for the EDI products.

Windows products

- Windows Server 2003 with Service Pack 1 or later
- Windows XP with Service Pack 2 or later
- Windows 2000 with Service Pack 4 or later

UNIX products

- Sun Solaris 2.6, 2.7, 2.8, and 2.9
- IBM AIX 5.1 and 5.2
- HP-UX 11i

Supported databases

The following list includes the supported database systems for the EDI products.

- Microsoft SQL Server 2000
- Sybase ASE 12.5 or later
- Oracle 9i or later
- IBM DB2 UDB 7.1 or later

Note
If you plan to use IBM DB2, you must bind the ODBC driver to the database. Note that MS MDAC does not include a DB2 driver. For more information about binding, see the DataDirect ODBC documentation that will be installed with the ODBC drivers or the IBM ODBC driver documentation.

- Microsoft Access 2000 or later

Note
MS Access is not supported for a production environment.

In addition, for the latest supported platforms and databases for EDI products, please see the Availability and Certification Reports website at http://certification.sybase.com.

ECMap

ECMap includes mapping tools that allow you to develop complex maps for processing EDI transactions. ECMap supports sequential and delimited file formats and standards, which include ANSI X12, UN/EDIFACT, NCPDP, and HL7.

Changes in ECMap 4.2 and subsequent EBF releases include support for new EDI standards, new rules, and new RunMap options.

Read_Uncommitted option [EBF 4.2.5]

The ECMap GUI has been updated with a new Read_Uncommitted option in the drop-down list of File Properties. This new option enables you to specify a lower isolation level that reduces locking contention for reading from the database. It is used by ECRTP to greatly reduce the locking contention for certain fields such as control numbers and Run IDs. You are able to select this option in ECMap.

Support for new versions of standards [EBF 4.2.4]

Table 1 lists the new versions of X12 and NCPDP standards that ECMap 4.2 supports.

Table 1: X12 and NCPDP standards supported by ECMap 4.2

Standard	Version
X12	• 005010 • 005011 • 005012 • 005020 • 005021 • 005022
NCPDP	• 5.1 • 5.2 • 5.3 • 5.4 • 5.5 • 5.6

TA1 acknowledgements

This section describes enhancements to EDI Products that enable real-time processing of TA1 acknowledgements. The following maps have been added: the TA1XLG and LOGUPDATE maps.

TA1XLG map [EBF 4.2.4]

A TA1 map, named TA1XLG, has been added to ECMap. This map is based on the original TA1 maps, TA1ODBCA and TA1ODBCT. TA1XLG reads log messages from an expanded text log instead of from an ODBC database. The other TA1 maps use ODBC. You set up the TA1XLG map as you would any other, through the Run Map screens in either ECMap or EC Gateway.

This map was added to support faster real time transaction processing by enabling the writing of log records to a memory buffer, instead of to disk. Writing to memory requires the records be written in expanded text log format. As a result, users can now run a TA1 map against these records stored in a memory buffer. As a final step, these log records should be written to disk by running the LOGUPDATE log persistence map. See “LOGUPDATE map [EBF 4.2.4]” on page 6 for more information.

LOGUPDATE map [EBF 4.2.4]

A log persistence map, named LOGUPDATE, has been built to take an in-memory ECRTP log, in expanded text log format, and write it to a log on disk, either in an ODBC database or an expanded text log file. See “TA1XLG map [EBF 4.2.4]” on page 5 for information on how to read log messages from the expanded text log.

This map allows the logging to disk to be delayed until the real-time critical processes have been completed. These include running the HIPAA compliance map and optionally the TA1 acknowledgement map, the 997 acknowledgement map, business rules compliance map, translation map and, if a query/reply transaction, other maps associated with the reply process.

Table 2 describes the parameters for the LOGUPDATE map.

Table 2: Parameters for LOGUPDATE map

Parameter	Description
1	If this parameter is set to Y, ECRTP writes the log to an expanded text log disk file. If this parameter is empty, the log will be written to a database. The DSN used for the log is 997. Follow the same instructions associated with running the 997 Acknowledgment map to correctly run this map.
2	If this parameter is set to Y, ECRTP updates the log to indicate a 997 Acknowledgment has been sent.
3	If this parameter is set to Y, ECRTP updates the log to indicate a TA1 Acknowledgment has been sent. If the inbound file had an error in the Interchange Acknowledgement Code (ISA14) or the sender ID (ISA06) is unknown or invalid, the log will be marked with an error indicating the TA1 could not be sent. An E appearing in the TR_ACK_TYP column in the TRLOG table reflects this error.

Rules

Rules are commands that govern the way ECMap processes your maps. Version 4.2 includes new rules that allow you to specify the location of incoming binary data and import rules from other maps.

Binary data placement [EDI 4.2]

A rule has been added that enables you to override the default location for the binary data of an incoming BIN segment by specifying the full path for the data to be stored.

Enhanced rules import [EDI 4.2]

The rule import feature has been enhanced to enable you to import multiple rules from another map and to specify how to handle duplicate rule numbers during the import. To use this feature, in the Rule Definitions window, select Rule | Import to open the Import Multiple Rules window. In the Import Rules Options section, you have options to:

- Overwrite all duplicate rules
- Skip all duplicate rules
- Renumber all incoming duplicate rules
- Overwrite, skip, or renumber rules on a rule-by-rule basis

For more information, see “Creating and using rules” in Chapter 20, “Working with Rules,” in the *ECMap User Guide*.

RunMap options

RunMap options apply to ECRTP, ECMap, and EC Gateway. Updates to the RunMap dialog box in ECMap and EC Gateway reflect additional runtime parameters and switches provided by ECRTP. For additional information about these new options and parameters, see “RunMap options” on page 18. You can also see Chapter 4, “Working with the Address Book,” and Chapter 17, “Running Maps,” in the *ECMap User Guide*.

Compliance map acknowledgement options [EDI 4.2]

New trading partner options allow you to specify how you want ECMap to issue an acknowledgement if it cannot find a trade agreement match. The three new options are:

- Acknowledge based on trade agreement (default behavior)
- Acknowledge if no trade agreement match is made
- Always acknowledge

See “Adding a new trading partner,” in Chapter 4, “Working with the Address Book,” in the *ECMap User Guide*.

Cross-reference table import utility [EDI 4.2]

A new utility allows you to import cross-reference tables from one map to another. This reduces the amount of work required to reuse cross-reference tables for multiple maps.

See “Importing cross-reference tables,” in Chapter 22, “Working with Cross-Reference Tables in the *ECMap User Guide*.

Record size increase [EBF 4.2.1]

The default record size has been increased from 32,767 bytes to 99,999 bytes.

Applies to	ECMap and ECRTP
ECMap version	4.2.1
Resolves	CR#314959

EC Gateway

EC Gateway is an enterprise-wide message management software, designed to support inter- and intra-company electronic messages. EC Gateway has three main functions: process management, communications, and scheduling. It includes extensive reporting facilities, a trading partner interface, a single ODBC-compliant log database, and archive/restore functions.

Updates to EC Gateway 4.2 include support for reusing e-FTP scripts, added security for files created by pfs/Async, new search functions, wildcard support for e-mail attachments, script support for command line parameters, and new RunMap options.

Reusing e-FTP scripts for multiple systems [EBF 4.2.3]

The e-FTP module can now obtain the script file name, server name, user ID, and password from the communications channel. With this added functionality, you can reuse the same e-FTP scripts for multiple systems by referencing different communication channels. Previously, only the script file name came from the communications channel; you could obtain the other information only from the script file itself.

Applies to	EC Gateway
EC Gateway version	4.2.3
Resolves	CR#382553

Added security for files created by pfs/Async [EBF 4.2.3]

The user ID for the person running pfs/Async is now recorded in the USER_IDENT field of the log file, *trlog*. Also, the files archived by the pfs/Async host will have the RunID of the pfs/Async host appended to the end of the file name. This ensures that the archived files have unique names, even if they are archived into common directories.

Applies to	EC Gateway
EC Gateway version	4.2.3
Resolves	CR#369152

Find and replace functions [EDI 4.2]

New Find and Replace options allow you to search for and replace strings in specific modules. Find and Replace options are available in the Comm Channel, Mailbox, Trade Partner, Company ID, and Process modules.

Searching for strings

To search for an instance of a string in a module:

1. Select Edit | Find.
2. In the Find field, enter the string you want to find.
3. Click Find Next.

Searching and replacing strings

To replace a search string with a replacement string in a module:

1. Select Edit | Replace.
2. In the Find field, enter the string you want to find, then type the replacement string in the Replace With field.

3 Select one of these buttons:

- Replace – replaces the current instance of the search string with the replacement string.
- Replace All – replaces all instances of the search string with the replacement string.
- Find Next – finds the next occurrence of the search string.
- Details – identifies the location of the search string.

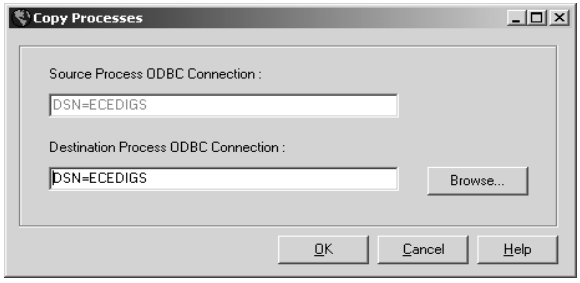
Copying process scripts to other systems [EDI 4.2]

A new Copy Process option is available in the Process module that lets you copy a process (or script) from the current DSN (specified by the current system) to another DSN (may be specified by any system). This lets you run the same process against multiple trading partner databases.

1 Right-click a process name and choose Copy Process.

Alternatively, click a process name, select Utility, then select Copy Processes.

A dialog box prompts you for the DSN of the destination database. The destination DSN box displays the source DSN by default.



2 In Destination Process ODBC Connection enter the current DSN you want to use.

Note In Windows, add the prefix “Copy (n) of...” to the name of any copy you make with the Copy Processes option. To change the name of the process, right-click the name of the copy, choose Properties, and change the name in the Process Name box.

Wildcard e-mail attachments [EDI 4.2]

E-mail parameters on the EC Gateway Process dialog box in Windows include an Other Options field, which allows you to include e-mail attachments. In versions of EC Gateway earlier than 4.2, you had to explicitly identify each file to include as an e-mail attachment with a separate –attach command for each binary file, and a separate –attacht command for each text file. You can now use wildcards to attach multiple files.

To attach multiple binary files enter:

–attach *path/filename*

For example, to attach all binary executables in the *attachments* directory that start with the letter 'a', enter:

```
–attach c:\attachments\a*.exe
```

This example attaches all tar files in the *attachments* directory:

```
–attach c:\attachments\*.tar
```

To attach multiple text files, enter:

–attacht *path/filename*

This example attaches all text files in the *attachments* directory:

```
–attacht c:\attachments\*.txt
```

Note The UNIX version of wsproces can functionally process the –uuencode or –mime options with file names containing asterisks into multiple file names to be attached to an e-mail message.

Passing input to the EC Gateway executable [EDI 4.2]

You can now pass input parameters to the EC Gateway executable on the command line. The switches for these are -p1 through -p20. A new Assign command has been added to simplify passing parameter values from the command line to Message variables, Parameter variables, or Shared Memory variables.

When you use the Assign command, you must select Destination and Source parameters for the assignment.

- The Destination parameter identifies the variable to initialize. You can choose a Message variable, Parameter variable, or Shared Memory variable for the destination.

- The Source parameter identifies the command line parameter to assign to the Destination parameter. To pass a value from the command line, point to CMD_Line_Parameter and then choose the one you want.

For the wsproces executable, the command line syntax is as follows:

> **wsproces** *script file name* [**-t** *directory for the wsproces trace file*] [**-r** *the run id for this wsproces run*] [**-td** *maximum number for the threads*] [**-p1** *command parameter 1*] [**-p2** *command parameter 2*] … [**-p20** *command parameter 20*]

In this example, the MQ manager and the MQ queue names are not explicitly defined in *testscript.pfs*. Instead, the script should assign CMD_Line_Parameter-1 to the parameter with the MQ manager name and CMD_Line_Parameter-2 to the parameter with the MQ queue name.

```
wsproces c:\ecedigs\testscript.pfs -p1 mq_mgr1 -p2
mq_queue1
```

Passing input to and getting output from the EC Gateway DLL shared library [EDI 4.2]

The EC Gateway DLL shared library provides a powerful new capability for passing input into EC Gateway from the calling program and returning output to the calling program. This is accomplished through an array of data structures called DATAIOSTRUCT where each structure contains a tag name, a pointer to the value, the length of the value, and the length of the buffer containing the value. Each structure can be referenced by its tag name or by its index, which is its position in the array structure.

With this capability, EC Gateway can dynamically increase the size of the array that it returns to the calling program. For example, if EC Gateway were to return an output for each item in the data, it would not know in advance, nor would the calling program know, how many items are to be returned. However, the array can be increased in size, one structure for each item found in the data.

The syntax for calling the DLL shared library is:

> __declspec(dllexport) int __stdcall wsprocesDataIO (LPSTR lpszCmdLine, DATAIOSTRUCT ** pDataIO)

where lpszCmdLine is the command line for the wsproces executable defined in section 3.6.

The structure DATAIOSTRUCT is defined as:

```
typedef struct {
char *tag;
char **pdata;
```

```
long *pdatalen;
long *pbuflen;
} DATAIOSTRUCT;
```

This structure has four fields, which are:

- tag – is for a pointer to the name of the data that can be referenced within a script.
- pdata – is for a pointer to the buffer for the data.
- pdatalen – is for a pointer to the length of the data.
- pbuflen – is for a pointer to the length of the storage buffer allocated for the data.

The array is terminated by an extra structure with its tag pointer set to NULL.

With this API, if there is no input data or output data to be passed in memory, the array of data I/O structures is not needed and the calling application should set the second parameter to NULL. The calling application must define a command line, which must include the script file name. For upward compatibility, for example if converting from the executable to the DLL, you can still enter the input parameters using the command line syntax of -p1 through -p20 and you can still assign them using the Assign command.

Two additional commands have been added, AssignInput and AssignOutput, to move data between the array of data structures and EC Gateway Message variables, Parameter variables, or Shared Memory variables.

When using the AssignInput command, you must provide values for the Destination, Source By, and Source Value parameters.

- The Destination parameter identifies the variable to initialize. You can choose a Message variable, a Parameter variable, or a Shared Memory variable for the destination.
- Source By is a string, either Tag or Index.
- Source Value depends on the value for Source By:
  - If the Source By is a Tag, then the value should be a string with the tag name of one of the structures in the array.
  - If the Source By is an Index, then it should be an integer with the index of a structure in the array.

  The first structure in the array corresponds to index zero (0).

When you use the AssignOutput command, you must provide the values for Destination By, Destination Value, Source Data, and Overwrite/Append parameters.

- Destination By and Destination Value define which structure in the array you want to set. Destination By is a string, either Tag or Index. Destination Value depends on the Destination By value.
    - If the Destination By value is a Tag, then the value should be a string. If that value matches a tag name in the original structure, that structure is updated. If that value does not match any tag name in the structure, EC Gateway automatically adds another structure to the array with that tag name.
    - If the Destination By value is an Index, then the value must be an integer in the range of the already existing structures in the array. If the Index is not in range, it does not add another structure to the array.
- The Source Data identifies the variable with the data to be used to update the structure. You can choose a Message variable, a Parameter variable, or a Shared Memory variable for the destination.
- The Overwrite/Append indicates whether the data should overwrite the data already in the buffer or append to it.

RunMap options

RunMap options apply to ECRTP, ECMap, and EC Gateway. Updates to the RunMap dialog box in ECMap and EC Gateway reflect additional runtime parameters and switches provided by ECRTP. For more about these new options and parameters, see “RunMap options” on page 18.

Return command [EDI 4.2]

The Return command allows you to include an optional return code, which is passed to the program that called it. You can use the Return command in the main script of the executable and the DLL shared library versions.

You can use Counter or Constant type variables as return codes. In the case of of the DLL shared library version, if no return code value is assigned, a value of zero (0) is returned. The Return command itself, if executed inside a subscript called by Do, DoWhile, WhileDo, or RunThread, terminates the processing of the subscript. There is no value in specifying a return code in this instance because it is ignored. If you execute the Return command in the main script, the main script terminates.

Exit command [EDI 4.2]

The Exit command allows you to use Counter and Constant type variables as a return code. Setting the code is optional, and if not specified, is set to zero (0) for the DLL version and to 260 for the executable version for backwards compatibility. The Exit command terminates execution of the entire script, including any subscripts, in the same thread or not. In general, do not use the Exit command in subscripts; doing so generates a warning during compilation.

StringLength command [EBF 4.2.1]

You can determine the length of a message at runtime. The StringLength command allows you to specify a Message variable as a string type, which returns the length of the message specified in the variable.

Applies to	EC Gateway, wsproces
EC Gateway version	4.2.1
Resolves	CR#372914

ECRTP

ECRTP is both a data transformation engine that analyzes, transforms, and routes messages and a compliance verification engine that checks EDI data for compliance with the corresponding EDI standard. ECRTP includes map files, a trading partner database, and log files.

- Map files contain the business rules and logic that define the relationships between incoming and outgoing data.
- The trading partner database contains information that routes messages between trading partners and that selects the specific map that should be run.
- Log files maintain an audit trail of the transaction processing.

Changes in ECRTP 4.2 include information about previously undocumented features, new RunMap options, and command line switches.

Setting a maximum waiting time for ECRTP to establish a database connection [EBF 4.2.5]

To reduce the probability of an ECRTP failure due to a temporary inability to establish a database connection, you can now set a maximum waiting time for ECRTP which specifies the maximum time period that ECRTP will wait for a successful database connection.

You can specify the time in milliseconds for ECRTP to try and connect to the database. If a database connection is or becomes available during that time period, ECRTP starts using the connection immediately.

If no connection becomes available by the end of the specified time period, ECRTP gives up waiting and returns with an error that says no database connection is available.

See “Setting a maximum waiting time for ECRTP to get a database connection [EBF 4.2.5]” on page 20 for the option used for setting this limit.

Log Bulk Insert feature [EBF 4.2.4]

With version 4.2.4, ECRTP provides a new feature on Windows only, Log Bulk Insert, that improves performance when using an ODBC database for the log, TRLOG. With this feature, ECRTP writes multiple log records at one time, instead of writing them one at a time, as was done prior to this enhancement. This feature will be provided on Unix platforms in an upcoming release.

The feature is controlled via an environment variable, TRLOG_BLKSIZE, which allows users to specify the maximum number of records to be inserted into the TRLOG database at one time. The allowable range is 0 to 500 records. The default value is 100.

ECRTP will automatically detect if the TRLOG database supports bulk inserts. If the TRLOG database does not support bulk inserts or if the TRLOG_BLOCKSIZE environment variable is set to 0 or 1, the program inserts records into TRLOG one at a time, as it does in EDI 4.2.3 and earlier releases.

If ECRTP cannot get sufficient memory for the bulk insert at the specified block size, it retries the insert with a block size of 50. If there is insufficient memory for the bulk inserts at a block size of 50, ECRTP inserts records one at a time.

In addition, with version 4.2.4, a *transNNN.log* will be written to disk if either the single TRLOG insert fails or the bulk insert fails. The *transNNN.log* file is a dump of the LOG records that were buffered in memory but could not be written to the ODBC TRLOG table.

The *transNNN.log* file will be stored in one of the following 2 directories:

- the default map directory, indicated by the -dg switch. This is a required switch for all ECRTP runs.
- the directory specified for the text log file with the -ed switch. This is an optional switch used for text log files.

If the insert to the ODBC TRLOG table fails, and the *transNNN.log* file is written to disk, the following error message will be written to the TRACE file (*TRNNN.DAT*).

(1042) FATAL: Failed To Write TRLOG - dumping log to *full path directory\transNNN.log*

See Chapter 2, "Running ECRTP," of the *ECRTP Reference Guide* for information about other ECRTP environment variables, and for the short and expanded formats for writing log records to disk. The expanded format matches the record layout of the TRLOG table.

JNI wrapper for ECRTP [EBF 4.2.3]

ECRTP 4.2.3 contains a Java Native Interface (JNI) Wrapper for the new ECRTP API, implemented in EDI 4.2. This wrapper enables Java developers to not only call ECRTP, a C/C++ based program, but also take advantage of the new memory I/O features of the API.

The JNI wrapper consists of one package, com.sybase.ecrtp, which contains the following classes:

- RTP
- RTPException
- UtilCountMapResult

For more information, see the *README.txt* file, Javadoc files, and demo files, which are provided as part of the ECRTP installation.

PADEDI environment variable [EDI 4.2]

PADEDI is a previously undocumented environment variable, which pads numeric fields with leading zeros or alphanumeric fields with trailing spaces at runtime to ensure that data meets minimal field length requirements.

ECRTP also includes a -pe switch, which pads EDI data to minimum length in compliance maps. This switch is an optional replacement for the PADEDI environment variable.

Windows

In Windows, in Control Panel | System | Advanced | Environment Variables | System Variables, set the Variable Name to PADEDI. Set the Variable Value to Y.

UNIX

In UNIX, open a shell and run the following commands relevant to your shell:

- ksh:

```
set PADEDI=YES ; export PADEDI
```

- sh:

```
set PADEDI=1 ; export PADEDI
```

- csh:

```
setenv PADEDI 1
```

Resolves

CR#297413

For more complete information about PADEDI, see “Padding EDI data to minimum length [EDI 4.2]” on page 30.

RunMap options

RunMap options apply to ECRTP, ECMap, and EC Gateway. Updates to the RunMap dialog in ECMap and EC Gateway reflect additional runtime parameters and switches provided by ECRTP.

Character checks for compliance maps [EBF 4.2.5]

Two new switches, -al and -vs, have been added to the Run Inbound Map windows to handle character checks for compliance maps. The -al switch disallows the use of high order ASCII characters between 128 — 255 in EDI data, while the -vs switch disallows using either the ^ (circumflex) or the ' (accent grave) characters as data characters if the X12 version is before 4040. It was not until X12 version 4040 that these two characters were permitted as data characters. Note also that prior to version 4040, even though these two characters weren't permitted to be used as data characters, they were permitted to be used as delimiters.

Inbound map runs

The -al switch disallows the use of the extended ASCII characters 128 — 255. These characters are generally used for Western European languages only.

Applies to: Inbound maps.

ECRTP version: EBF 4.2.5

Resolves: CR#409591 and 409592

Command line syntax: The syntax for this switch is:

-al

❖ **Allowing only ASCII characters for inbound maps in Windows**

1 Open the Run Inbound Map window.

2 Click the Option 1 tab.

3 Select Allow only ASCII characters.

Inbound map runs

The -vs switch disallows the use of the ^ (circumflex) or ' (accent grave) as data characters in X12 versions prior to version 4040. An error will be generated if one of these is used in this way. Note this -vs switch allows the use of these characters as delimiters, even before version 4040.

Applies to: Inbound maps.

ECRTP version: EBF 4.2.5

Resolves: CR#409591 and 409592

Command line syntax: The syntax for this switch is:

-vs

❖ **Performing version specific character checks for inbound maps in Windows**

1 Open the Run Inbound Map window.

2 Click the Option 1 tab.

3 Select Perform Version Specific Character Checks.

Setting a maximum waiting time for ECRTP to get a database connection [EBF 4.2.5]

A new -tc *NN* switch, where *NN* is a number in milliseconds, allows you to specify a maximum waiting time for ECRTP to establish a database connection. If a connection is or becomes available during that time period, ECRTP immediately stops waiting and starts using the connection. If no connection becomes available during that time period, it gives up waiting and returns with an error that says no database connection is available.

See “Setting a maximum waiting time for ECRTP to establish a database connection [EBF 4.2.5]” on page 16.

Inbound map runs

The option for the -tc *NN* switch is called “DBMS Connection Retry Time (msec)” and it appears on the Option 1 tab for inbound processing.

Applies to	Inbound maps.
ECRTP version	EBF 4.2.5
Resolves	CR#295784, 407159 and 407160
Command line syntax	The syntax for this switch is: **-tc *NN***

❖ **Setting a maximum waiting time for inbound map runs in Windows**

1 Open the Run Inbound Map Acknowledgment window, or the Run Inbound Map window.

2 Click the Option 1 tab.

3 Enter a value in DBMS Connection Retry Time (msec).

Note The default *NN* value for this switch is 1000 msec, where msec refers to milliseconds. It can take a range of values between 0 and 21600000 milliseconds (6 hours).

Outbound map runs

The option for the -tc *NN* switch is called “DBMS Connection Retry Time (msec)” and it appears on the Option 1 tab for outbound processing.

Applies to	Outbound maps.
ECRTP version	EBF 4.2.5
Resolves	CR#295784, 407159 and 407160
Command line syntax	The syntax for this switch is: **-tc *NN***

❖ **Setting a maximum waiting time for outbound maps in Windows**

1 Open the Run Outbound Map window.

2 Click the Option 1 tab.

3 Enter a value in DBMS Connection Retry Time (msec).

Note The default *NN* value for this switch is 1000 msec, where msec refers to milliseconds. It can take a range of values between 0 and 21600000 milliseconds (6 hours).

Controlling updates and lookups for control numbers [EBF 4.2.4]

A new -cn switch allows you to control updates and lookups for control numbers when running maps. This switch works differently for inbound and outbound map runs. Using this option will improve performance for those customers who do not care about control numbers.

Inbound map runs

The -cn switch disables control number lookup and updates in the trading partner database during inbound map runs.

Applies to	Inbound maps.
ECRTP version	EBF 4.2.4
Resolves	CR#388842
Command line syntax	The syntax for this switch is: **-cn**

❖ **Disabling control number updates for inbound maps in Windows**

1 Open the Run Inbound Map window.

2 Click the Option 2 tab.

3 Select No Control Number Updates.

Note The Validate Control Number Sequence option in the Option 2 tab of the ECMap Run Inbound Map window is disabled when you select the No Control Number Updates option, as no validation can occur if no control numbers are being updated, and vice versa.

The same holds true for the corresponding command line switches, -sc and -cn. The switches cannot be used concurrently. See "Validating the Interchange (ISA) and Group (GS) control number sequence [EDI 4.2]" on page 23.

If you do not use -cn and -sc on the command line or if you do not select No Control Number Updates and Validate Control Number Sequence in Windows, the control numbers will be updated to the database but will not be validated.

Outbound map runs

The -cn *control number* switch disables control number lookups and updates in the trading partner database during outbound map runs. The *control number* option is provided for the switch to accept a value for *control number* as input from the calling program or from the user. This will override the default value from ECRTP. The value for *control number* is used for the outbound ISA.

Applies to	Outbound maps.
ECRTP version	EBF 4.2.4
Resolves	CR#388842
Command line syntax	The syntax for this switch is: **-cn *control number***

❖ **Disabling control number updates and specifying a control number in Windows**

1 Open the Run Outbound Map window, or open the Run Outbound Map Acknowledgment window.

2 Click the Option 1 tab.

3 Select Enter Control Number Override. A text box will appear to the right of the check box. You may enter a value for the control number in the text box.

Note the following for both Windows and the command line:

- The control number must be a valid integer and can be up to 9 digits.
- If a value for control number with less than 9 digits is entered, ECRTP will fill up the field with zeros as it has a fixed length of 9 digits.
- If the option is checked in Windows and the switch is used in the command line but no entry is made in the text box, the default value of 000000001 (1) will be passed as the control number.
- If multiple interchanges occur, the interchange control number will be incremented by 1 from the user supplied value.
- The group control number will start at 1 for each interchange encountered and increment accordingly.

Validating the Interchange (ISA) and Group (GS) control number sequence [EDI 4.2]

The -sc switch validates the sequence of the Interchange (ISA) and Group (GS) control numbers.

If you use this switch, ECRTP checks the current control number against the trading partner database to validate the entry. If the entry in the file is not the next sequential entry, ECRTP reports an error.

Applies to	Inbound compliance maps.
ECRTP Version	4.2
Resolves	CR#322012
Command line syntax	The syntax for this switch is: **-sc**

❖ **Validating ISA and GS control sequence numbers in Windows**

1 Open the Run Inbound Map window.

2 Click the Option 2 tab.

3 Select Validate Control Number Sequence.

Note The Validate Control Number Sequence option in the Option 2 tab of the ECMap Run Inbound Map window is disabled when you select the No Control Number Updates option, as no validation can occur if no control numbers are being updated, and vice versa.

The same holds true for the corresponding command line switches, -sc and -cn. The -sc switch cannot be used concurrently with the -cn switch. See "Controlling updates and lookups for control numbers [EBF 4.2.4]" on page 21.

If you do not use -cn and -sc on the command line or if you do not select No Control Number Updates and Validate Control Number Sequence in Windows, the control numbers will be updated to the database but will not be validated.

Override for compliance map results [EDI 4.2]

New RunMap options allow you to override the default location for good and bad compliance map data. You can now specify the full file paths for storing compliance map results. These options correspond to the new -ab and -ag ECRTP command line parameters. See "RunMap options" on page 18 for more information. See also Chapter 4, "Working with the Address Book," and Chapter 17, "Running Maps," in the *ECMap User Guide*.

Specifying a file to use in place of a BAD EDI file [EDI 4.2]

A new -ab switch allows you to specify which file to use in place of the BAD EDI file. You must specify a full path and file name.

- If you specify the -ab switch from the command line, ECRTP ignores the original BAD EDI destination, and writes the BAD EDI file to the path and file name you specify as parameters on the command line.
- If the file specified by the -ab switch is passed in the MemIOStruct, ECRTP writes to the memory buffer instead of disk.
- If you specify a file name without a path, ECRTP writes the BAD file to the current working directory for disk I/O, which may or may not be the map directory.

Applies to Inbound maps.

ECRTP version 4.2

Command line syntax

The syntax for this switch is:

-ab *path/filename*

❖ **Substituting a BAD file name in Windows**

1 Open the Run Inbound Map window.

2 Click the Option 2 tab.

3 Click Browse next to the Substitute Bad Filename field and choose an appropriate destination directory and file name.

Specifying a file to use in place of a GOOD EDI file [EDI 4.2]

In ECRTP version 4.2, for inbound maps only, a new -ag switch allows you to specify which file to use in place of the GOOD EDI file. You must specify a full path and file name.

- If you pass the -ag switch from the command line, ECRTP ignores the original GOOD EDI destination, and writes the GOOD EDI file to the path and file name you specify as parameters on the command line.
- If the user specifies a file name without a path, then ECRTP writes the GOOD file to the current working directory for disk I/O, which may or may not be the map directory.

Applies to

Inbound maps.

ECRTP Version

4.2

Command line syntax

The syntax for this switch is:

-ag *path/filename*

❖ **Substituting a GOOD filename in Windows**

1 Open the Run Inbound Map window.

2 Click the Option 2 tab.

3 Click Browse next to the Substitute Good Filename field and choose an appropriate destination directory and file name.

Specifying map name extensions [EDI 4.2]

In ECRTP version 4.2, for inbound and outbound maps, a new -mn switch passes the map name extension as part of a command line argument. A Map Name Extension option has been added to the RunMap and Trade Agreement windows.

The Map Name Extension works with other map lookup fields to find a correct map by allowing you to specify an additional identifier to distinguish multiple maps for the same transaction within a single trade agreement. This supports the need for multiple 287 and 837 HIPAA transactions.

For map functions with multiple parameters, such as LOADMAP, the correct map name should be found by using current map lookup fields with the newly added Map Name Extension before calling the functions.

Applies to: Inbound and outbound maps.

ECRTP Version: 4.2

Command line syntax: The syntax for this switch is:

-mn *map name extension*

❖ **Specifying a map name extension in Windows**

1 Open the Run Inbound Map or Run Outbound Map window.

2 Click the Required tab.

3 In the Map Name Extension box, type the appropriate map name extension.

-mns switch removed for ECRTP 4.2 [EDI 4.2]

The -mns switch has been removed as of version 4.2.

Including new return codes [EDI 4.2]

In ECRTP 4.2, for inbound compliance maps only, the -nret switch adds new return codes that provide additional information to the calling program. Based on the return value, you can determine the next step in the process. The return codes reflect the following information:

- At least one interchange is in error, implying a TA1 map should be run.
- At least one group or transaction is in error, implying a 997 map should be run.
- At least one good transaction is present, implying a translation map should be run.

ECRTP returns a new code based on these conditions, and may trigger other processes. A given return code may, for example, indicate that TA1, 997, and transformation maps should be run.

Applies to — Inbound compliance maps.

ECRTP version — 4.2

Command line syntax — The syntax for this switch is:

-nret

❖ **Adding the new return codes in Windows**

1 Open the Run Inbound Map window.

2 Click the Option 1 tab.

3 Select Enhanced Return Codes.

Examples — This list specifies the new return codes and identifies the codes that should trigger the TA1, 997 or translation maps:

- Return Code 16 **–** ECRTP aborted processing.
- Return Code 8 – a map abort occurred.
- Return Codes 4, 5, 6, or 7 – indicates a TA1 acknowledgment map should be run because there was at least one interchange level error.
- Return Codes 2, 3, 6, or 7 – indicates a 997 acknowledgment map should be run because there was at least one group or transaction level error.
- Return Codes 1, 3, 5, or 7 – indicates a translation map should be run because there was at least one good transaction.

NCPDP file processing [EDI 4.2]

Changes to NCPDP processing include new switches that identify NCPDP batch and telecommunications files.

NCPDP batch files [EDI 4.2]

A new -ncb switch indicates that the EDI file you want to process is an NCPDP batch file. Required for all inbound NCPDP files unless -nct is active.

Applies to — Inbound maps.

ECRTP version — 4.2

Resolves — CR#322012

Command line syntax — The syntax for this switch is:

-ncb

❖ **Processing NCPDP batch files in Windows**

1 Open the Run Inbound Map window.

2 Click the Required tab.

3 Select NCPDP Batch.

NCPDP telecommunications files [EDI 4.2]

A new -nct switch indicates that the EDI file you want to process is an NCPDP telecommunications file. Required for all inbound NCPDP files unless -ncb is active.

Applies to	Inbound maps.
ECRTP Version	4.2
Resolves	CR#322012
Command line syntax	The syntax for this switch is: **-nct**

❖ **Processing NCPDP telecommunication files in Windows**

1 Open the Run Inbound Map window.

2 Click the Required tab.

3 Select NCPDP Telecommunications.

Literal numeric data mapping [EDI 4.2]

The -nz switch maps numeric data (including blank fields) exactly as it appears on the map. Earlier versions of ECRTP padded numeric values with leading zeros are based on field length.

Applies to	Inbound maps.
ECRTP version	4.2
Command line syntax	The syntax for this switch is: **-nz**

❖ **Adding literal mapping options in Windows**

1 Open the Run Inbound Map window.

2 Click the Option 2 tab.

3 In the Zero Handling Options group, select No Zero Fill on Null or Non-Null Numbers.

Examples

Different switch options provide different results. These examples assume a numeric field size of 5, and the following input values: '123', '0', ' ' (empty).

- Default (no switch) option, the output is: `'123'`, `'0'` and `'0'`.
- -z switch option, the output is: `'00123'`, `'00000'` and `' '` (empty).
- -nz switch option, the output is: `'123'`, `'0'` and `' '` (empty).

Leading zeros in HIPAA compliance maps [EDI 4.2]

The -clz switch flags leading zeros in numeric X12 fields as an error on HIPAA compliance maps.

ECRTP flags an error if leading zeros are not used to meet the minimum length requirement of that particular element. If the leading zeros are necessary to meet the minimum length of the element, no error is generated. This switch does not check leading zeros on noncompliance maps.

Applies to

Inbound maps.

ECRTP version

4.2

Command line syntax

The syntax for this switch is:

-clz

❖ **Flagging leading zeros in Windows**

1 Open the Run Inbound Map window.

2 Click the Option 2 tab.

3 Select Generate Error for Invalid Leading Zeros.

ST control number validation

This section describes new control number validation options for inbound compliance maps.

Check for ST control number sequence [EDI 4.2]

The -as switch ensures that each ST Transaction Control Number within a GS to GE group is greater than the previous ST Transaction Control Number.

Applies to

Inbound maps.

ECRTP version 4.2

Command line syntax The syntax for this switch is:

-as

❖ **Validating ST control numbers in Windows**

1 Open the Run Inbound Map window.

2 Click the Option 2 tab.

3 In the Transaction Control Number Check group, select Increasing Control Numbers.

Check for unique ST control numbers [EDI 4.2]

The -cu switch checks for unique control numbers within a transaction. ST Transaction Control Numbers can now occur in any sequence, as long as they are unique.

If duplicate control numbers are found, a 6054 error is written to the log.

Applies to Inbound maps.

ECRTP version 4.2

Command line syntax The syntax for this switch is:

-cu

❖ **Validating control numbers in Windows**

1 Open the Run Inbound Map window.

2 Click the Option 2 tab.

3 In the Transaction Control Number Check group, select Unique Control Numbers.

Padding EDI data to minimum length [EDI 4.2]

The -pe switch pads EDI data to minimum length in compliance maps, which ensures that data meets minimal field length requirements. This switch is an optional replacement for the PADEDI environment variable.

Using this switch pads the data without changing the value of the data. The -pe switch pads alphanumeric fields with trailing spaces if those spaces are required to meet the minimum length of the element. Numeric fields are padded with leading 0s (zeros) if those 0s are required to meet the minimum length of the element.

Applies to	Outbound maps.
ECRTP version	4.2
Command line syntax	The syntax for this switch is: **-pe**

❖ **Padding EDI data to minimum length in Windows**

1 Open the Run Outbound Map window.

2 Click the Option 1 tab.

3 Select Pad EDI Data to Minimum Length.

Outbound trading partner lookup option [EDI 4.2]

The -ol switch triggers a series of lookups against the trading partner database when the ECMap/EC Gateway Log is used as input. If a trading partner match is found, outbound processing uses the entry to populate the EDI envelope.

Applies to	Outbound maps.
ECRTP version	4.2
Command line syntax	The syntax for this switch is: **-ol**

❖ **Adding the outbound trading partner lookup option in Windows**

1 Open the Run Outbound Map window.

2 Click the Option 1 tab.

3 Select Lookup Trading Partner from ECRTP Log.

Specifying the maximum number of cached ODBC connections [EDI 4.2]

A new -sdb switch allows you to specify the maximum number of cached ODBC connections. The default value ‘0’ indicates no ODBC connection caching.

Applies to	Inbound and outbound maps.
ECRTP version	4.2
Command line syntax	The syntax for this switch is: **-sdb** *number of cached connections*

❖ **Specifying connection caches in Windows**

1 Open the Run Inbound Map or Run Outbound Map window.

2 Click the Option 1 tab.

3 In the Number of ODBC Cache box, type the number of cache connections.

Command line switches [EDI 4.2]

Table 3 describes the available switches for ODBC and non-ODBC trading partner files. These switches correspond to various RunMap options in ECRTP, EC Gateway, and ECMap, and apply to both Windows and UNIX platforms.

Table 3: Switches for ODBC and non-ODBC trading partners

Parameter	Description	Inbound/ outbound	Required/ optional	NFG Version
-ab	Specifies a new full path and file name to be used in place of a BAD EDI file.	Inbound	Optional	4.2
-ag	Specifies a new full path and file name to be used in place of a GOOD EDI file.	Inbound	Optional	4.2
-al	Handles character checks for compliance maps. Only valid ASCII characters with values 0 — 127 are allowed, and not the extended ASCII characters 128 — 255.	Inbound	Optional	4.2
-as	Checks that each ST Transaction Control Number in a GS to GE is greater than the previous ST Transaction Control Number. Validation assumes that control numbers appear in ascending order.	Inbound	Optional	4.2
-clz	Flags leading zeros in numeric X12 fields as an error on HIPAA compliance maps. An error is flagged if leading zeros are not used to meet the minimum length requirement of that particular element. If the leading zeros are necessary to meet the minimum length of the element, no error is generated. This option does not check leading zeros on noncompliance maps.	Inbound	Optional	4.2

Parameter	Description	Inbound/ outbound	Required/ optional	NFG Version
Inbound: -cn Outbound: -cn *control number*	Disables control number lookups and updates when running inbound and outbound maps. For inbound processing, -cn will prevent the code from doing any control number updates. Use either -cn or -sc. For outbound processing, -cn *control number* will disable control number lookups and updates when running an outbound map and the value provided for *control number* will be used instead as the control number for the outbound ISA.	Inbound and outbound	Optional	4.2.4
-cu	Checks for unique control numbers within a transaction. Control numbers can now occur in any sequence, as long as they are unique. If duplicate control numbers are found, ECRTP logs a 6054 error. Use only one of -as or -cu for any map run. If both switches are present, ECRTP defaults to the last switch encountered.	Inbound	Optional	4.2
-mn	Passes the map name extension as part of a command line argument. The Map Name Extension works with other map lookup fields to find a correct map.	Inbound and outbound	Optional	4.2
-ncb	Indicates that the EDI file to be processed is an NCPDP batch file. Required for all inbound NCPDP files unless -nct is active.	Inbound	Required for inbound NCPDP batch files	4.2
-nct	Indicates that the EDI file to be processed is an NCPDP telecommunications file. Required for all inbound NCPDP files unless -ncb is active.	Inbound	Required for inbound NCPDP telecommunications files	4.2
-nret	Adds new return codes that provide additional information to the calling program. Based on the return value, the user can determine the next step in the process.	Inbound	Optional for compliance maps only	4.2

Parameter	Description	Inbound/ outbound	Required/ optional	NFG Version
-nz	Maps numeric data exactly (including blank fields) as it appears on the map.	Inbound	Optional	4.2
-ol	Triggers a series of lookups against the trading partner database when the ECMap/EC Gateway Log is used as input. If a trading partner match is found, ECRTP uses the entry to populate the EDI envelope.	Outbound	Optional	4.2
-pe	Pads EDI data to minimum length in compliance maps. This option ensures that data meets minimal field length requirements. This switch is an optional replacement for the PADEDI environment variable.	Outbound	Optional	4.2
-sc	Validates the sequence of the Interchange (ISA) and Group (GS) control numbers. If you use this switch, RTP checks the current control number against the trading partner database to validate the entry. If the entry in the file is not the next sequential entry, an error is reported.	Inbound	Optional	4.2
-sdb	Specifies the maximum number of cached ODBC connections. The default value '0' indicates no ODBC connection caching.	Inbound and outbound	Optional	4.2
-tc *NN*	Specifies a maximum waiting time, in milliseconds, for ECRTP to establish a database connection. The default *NN* value for this switch is 1000 milliseconds.	Inbound and outbound	Optional	4.2
-vs	Checks the validity of ASCII characters from 0 — 127 against the version of X12. If the character is ^ (circumflex) or ' (accent grave) and the X12 version is prior to 4040, the character is invalid and an error will be generated.	Inbound	Optional	4.2

ODBC switches [EDI 4.2]

Table 4 identifies switches that are specific to ODBC trading partners, which correspond to the new NCPDP trading partner lookup values. You can select only one ODBC trading partner switch for a given map.

Table 4: Switches for ODBC trading partners only

Parameter	Description	Inbound/ outbound	Required/ optional	NFG Version
-e14	Performs the trading partner lookup based on Batch Sender. Validates interchange and group data against the trading partner database during inbound processing, and, if any information does not match, ECRTP writes an error to the log and returns an error in the TA1 acknowledgment.	Inbound	Optional	4.2
-e15	Performs the trading partner lookup based on Batch Receiver.	Inbound	Optional	4.2
-e16	Performs the trading partner lookup based on Batch Sender and Receiver.	Inbound	Optional	4.2
-e17	Performs the trading partner lookup based on Bin Number.	Inbound	Optional	4.2
-e18	Performs the trading partner lookup based on Processor Control Number.	Inbound	Optional	4.2
-e19	Performs the trading partner lookup based on Batch Sender and Bin Number.	Inbound	Optional	4.2
-e20	Performs the trading partner lookup based on Batch Sender and Processor Control Number.	Inbound	Optional	4.2
-e21	Performs the trading partner lookup based on Batch Receiver and Bin Number.	Inbound	Optional	4.2
-e22	Performs the trading partner lookup based on Batch Receiver and Processor Control Number.	Inbound	Optional	4.2