

SYBASE®

New Features

EDI Products 5.2.2

Document ID: DC39126-01-0522-01
Last revised: April, 2011

Copyright © 2011 by Sybase, Inc. All rights reserved.

This publication pertains to Sybase software and to any subsequent release until otherwise indicated in new editions or technical notes. Information in this document is subject to change without notice. The software described herein is furnished under a license agreement, and it may be used or copied only in accordance with the terms of that agreement.

To order additional documents, U.S. and Canadian customers should call Customer Fulfillment at (800) 685-8225, fax (617) 229-9845.

Customers in other countries with a U.S. license agreement may contact Customer Fulfillment via the above fax number. All other international customers should contact their Sybase subsidiary or local distributor. Upgrades are provided only at regularly scheduled software release dates. No part of this publication may be reproduced, transmitted, or translated in any form or by any means, electronic, mechanical, manual, optical, or otherwise, without the prior written permission of Sybase, Inc.

Sybase trademarks can be viewed at the Sybase trademarks page at <http://www.sybase.com/detail?id=1011207>. Sybase and the marks listed are trademarks of Sybase, Inc. A ® indicates registration in the United States of America.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world.

Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

Unicode and the Unicode Logo are registered trademarks of Unicode, Inc.

All other company and product names used herein may be trademarks or registered trademarks of the respective companies with which they are associated.

Use, duplication, or disclosure by the government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of DFARS 52.227-7013 for the DOD and as set forth in FAR 52.227-19(a)-(d) for civilian agencies.

Sybase, Inc., One Sybase Drive, Dublin, CA 94568

Contents

1. About this Document.....	3
2. Runtime Switches.....	4
Runtime Switch to Specify extended ASCII: -ah.....	4
Runtime Switch to Specify BASIC ASCII: -aa.....	4
Runtime Switch to Specify Lower ASCII: -al.....	4
Runtime Switch to Specify Compliance NO EDI Text Blocking: -nn.....	5
Runtime Switch to Log each ISA and GS: -isa.....	6
3. EC Map.....	7
New Rule Command Options.....	7
New Run Map Options.....	9
4. ECGateway.....	11
New Run Map Options.....	11
5. Logging.....	13
6. Appendix A: TRLog table definition.....	14
7. Appendix B: ASCII Tables.....	22

1. About this Document

New Features in Sybase EDI Products 5.2.2

This section describes new features and changed functionality for Sybase EDI Products, version 5.2.2.

Previous versions of the EDI products have allowed users to use certain functions when running compliance maps. The 5.2.2 version of the EDI Products has expanded those functions to meet HIPAA Addenda and CMS requirements.

These changes include the following major changes:

- Improved usage for ASCII compliance
- Expanded Functional Acknowledgement Capability
- Logging of each ISA GS

Further information on these changes as well as changes and improvements made relating to these can be found in this document.

2. Runtime Switches

Runtime Switches for ASCII when running compliance maps:

Per the ASC X12 AN Implementation Guide standards for the HIPAA Addenda, Appendix B, sections 1.1.2.2 and 1.1.2.3 there are two sets of data defined – the basic character set and the extended character set.

EXTENDED ASCII is generally defined as the character set below where the numbers are decimal representations of the character value (BASIC ASCII) plus the addition of an Extended Character Set. All of the characters that are valid in BASIC ASCII are also valid in Extended ASCII.

According to the IG Appendix B section 1.1.2.3, the Extended Character Set includes the following characters:

a-z % ~ @ [] _ { } \ | < > # \$

The Basic Character Set includes A-Z, 0-9, and a series of other non-alphanumeric characters.

The default option in the EDI product is all characters ≥ 32 and ≤ 255 as characters which can be used in text fields.

Note: See Appendix B in this document for ASCII Table and Extended ASCII Codes.

Specify extended ASCII: -ah

The difference between the Lower ASCII in previous versions of the EDI products and the new Extended ASCII is the EXTENDED ASCII does not allow the carat symbol (decimal value 94) and the grave accent character (decimal value 96) as valid characters. The Extended Character set will also not allow the character signal DEL (value 127) as a valid character. The command line switch for Extended ASCII is -ah.

Specify BASIC ASCII: -aa

BASIC ASCII is defined as the character set below where the numbers are decimal representations of the character value. The command switch for Basis ASCII is -aa.

Specify lower ASCII: -al

In previous versions of the EDI products, Lower ASCII option would not write errors for characters < 32 . Characters < 32 were automatically skipped unless the no noise option (-nn command line switch) was set. When the no noise option was set Lower ASCII would not flag characters < 32 as errors. With the -al command switch, those errors are now written.

Compliance NO EDI Text Blocking: -nn

In previous versions of EDI when "Compliance No EDI Text Wrapping" was turned on, Lower ASCII would not flag characters < 32 as errors. This has been renamed to "Compliance NO EDI Text Blocking" to allow handling of a segment delimiter of CRLF as well as a segment delimiter followed by CRLF.

EDI files no longer need to be one continuous line when the "no noise" switch (-nn) is used. EDI text that is blocked at a fixed length cannot be processed when "no noise" is used. Setting just Extended ASCII or Basic ASCII will cause error messages to be written for invalid characters >= 32.

Setting Extended ASCII or Basic ASCII and also the no noise option, "Compliance NO EDI Text Blocking", will cause error messages to be written for invalid characters < 32 in addition to invalid characters > 32.

Valid Extended ASCII character set:

32-93
95
97-126

BASIC ASCII is defined as the character set below where the numbers are decimal representations of the character value.

32-34
38-59
61
63
65-90

Runtime Switch to log each ISA and GS: -isa

When the new Compliance option to "Log Each ISA GS", -isa command line switch is present, the ECRTP will write one new "NEW ENVELOPE" message to the TRLOG before the first "START TRANS" is written for each ISA to IEA envelope.. The System Variables for ISA, GS, and ST segments will have been loaded at the time the "NEW ENVELOPE" message is written and the System Variables will be written to the TRLOG as a part of the "NEW ENVELOPE" message.

The "NEW ENVELOPE" message is written only once, and it is written only if the ISA is good and there is at least one good GS and a good ST in the ISA to IEA envelope. For each new ISA, GS, a log record enters information into the TRLOG fields MSG_TEXT, MSG_NO, TYP, and STAT, in addition to Log Values which are loaded from new ISA, GS and ST.

3. ECMap

New Rule Command Options

Include Context (CTX) Variables

Six new TRLOG Fields and a corresponding six new System Variables have been added to ECMAP/ECRTP. These TRLOG fields and SysVars are: CTX_FIELDVAL), CTX_LINENO , CTX_SEGNAME, CTX_ELEMENT, CTX_SUBELEMENT , and CTX_REPEATNO. They are used at the time of Write Log to enable values to be reported in the 999 functional acknowledgement.

The CTX error codes that are used in the 999 transaction are:

Code	Output Value	Element	Description
2302	2	IK403	Condition Required Element Missing
2310	10	IK403	Exclusion Condition
2340	16	IK304	Implementation Dependent Required Segment Missing
2343	19	IK304	Implementation Not Used Segment Present
2347	113	IK403	Implementation Dependent Not Used Data Present
4568	10	IK403	Exclusion Condition
6053	110	IK403	Implementation Not Used Data Present

To use the variables in the Write Log, the user must check the new “Include Context (CTX) Variables” check box.

Include Log_Value

The contents of the System Variable, LOG_VALUE will be written when a write log command has the Include Log_Value check box checked. If the MAP is an EDI->EDI map and the WRITE LOG command takes place during the outbound portion of an EDI->EDI MAP, then the contents of the SysVar LOG_VALUE_EDIOUT is written to the TRLOG Field FIELDVAL. Both SysVars LOG_VALUE and LOG_VALUE_EDIOUT have been increased from 30 to 99 characters. And the TRLOG Field FIELDVAL has been increased to 99 characters.

The Write Log command GUI shows the “Include Context (CTX) Variables” and the “Include Log_Value” check boxes.

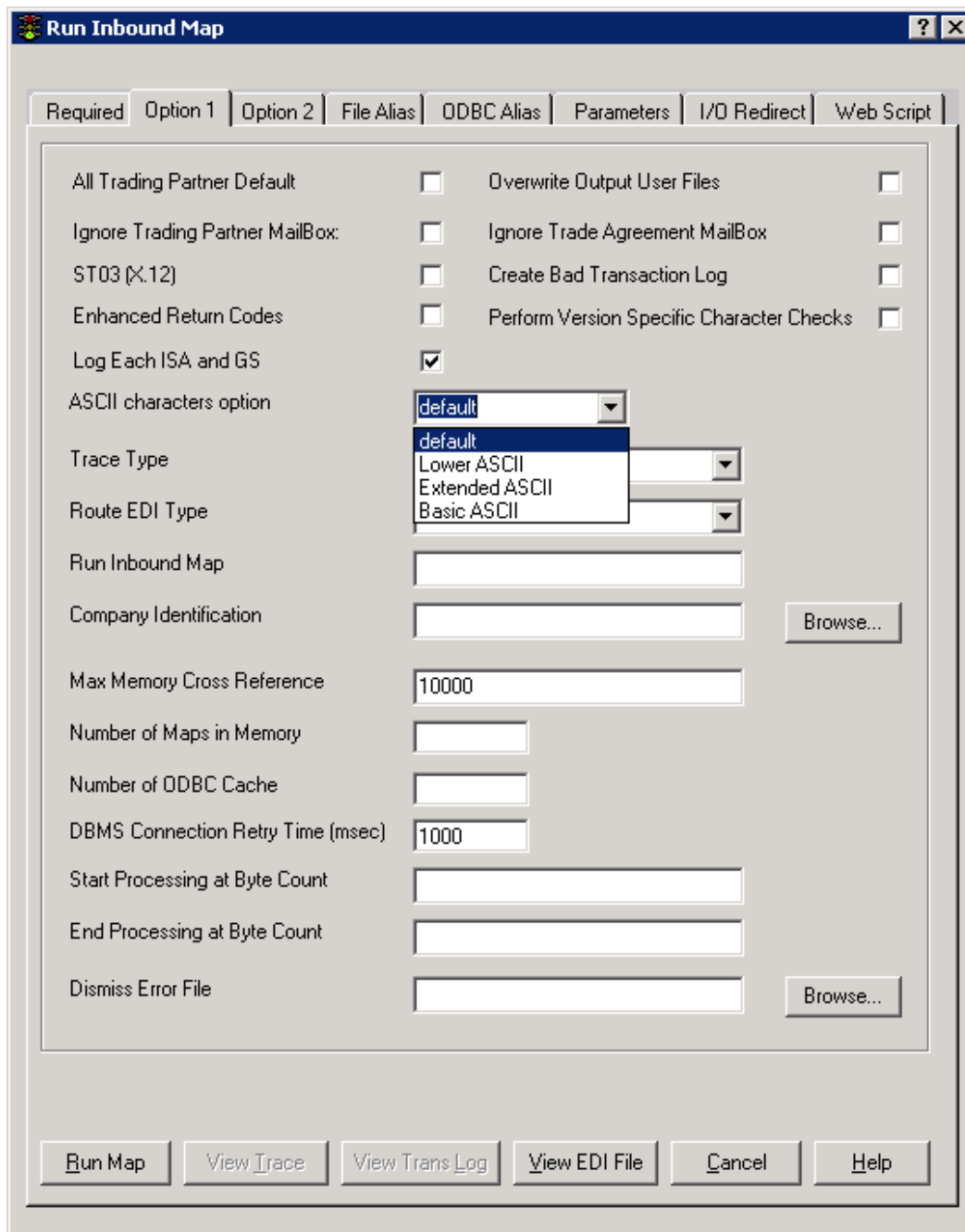
The screenshot shows a window titled "Write Log File" with the following fields and controls:

- Msg No : [Dropdown] [Text]
- Sev Code: [Dropdown] [Text]
- Message : [Dropdown] [Text]
- Record Seq No : [Dropdown] [Text]
- Segment Name : [Text]
- Element No : [Dropdown] [Text]
- Subelement No : [Dropdown] [Text]
- Segment Count : [Dropdown] [Text]
- Record Name : [Text] [Browse...]
- Field Name : [Text]
- Include Context (CTX) Variables :
- Include Log_Value :
- Buttons: OK, Cancel, Help

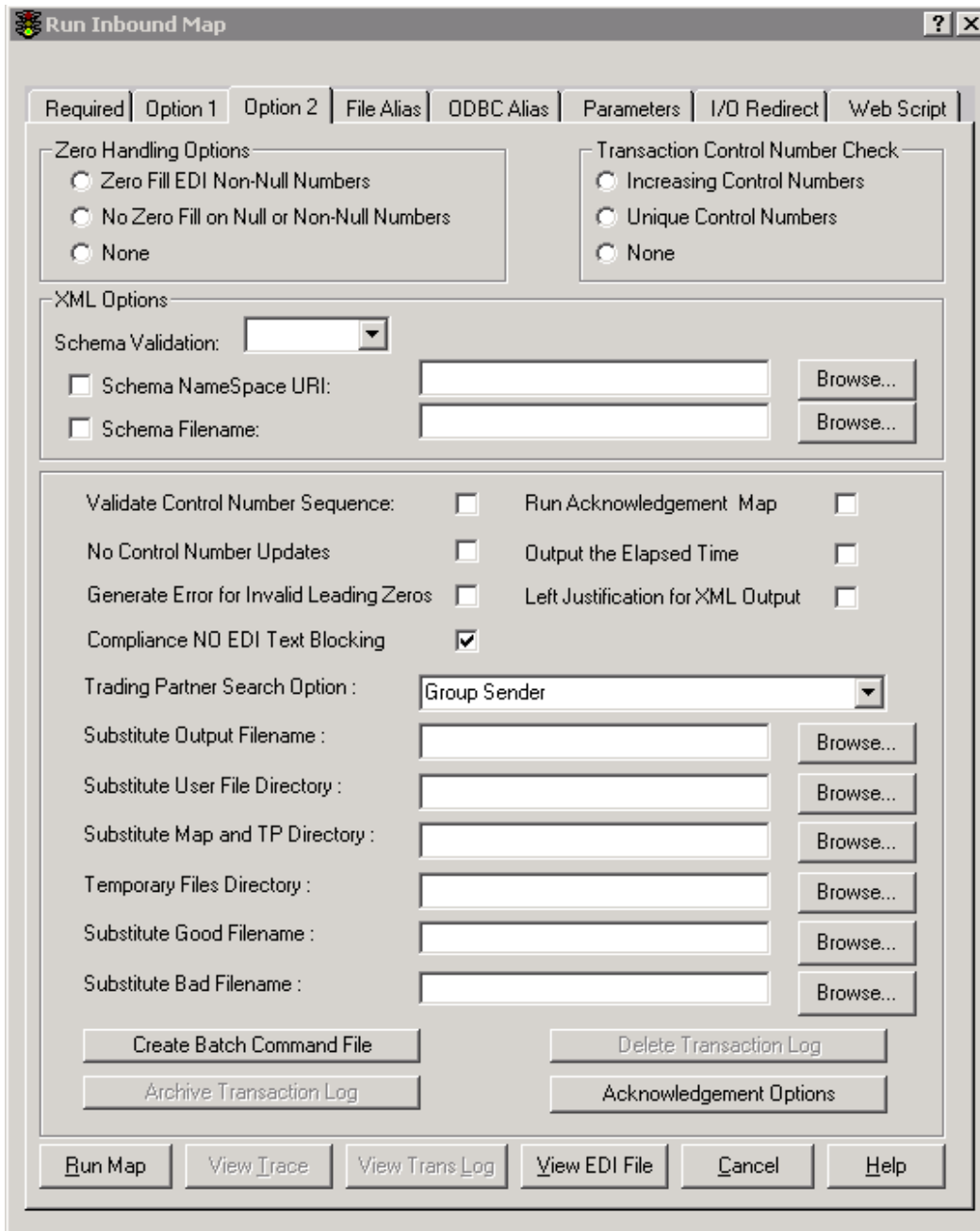
Run Map Options

GUI Runtime options

In addition to the command line runtime switches, the GUI can be used to perform the switch functions. The Run Inbound Map Option 1 tab shows the drop down menu for the “ASCII characters option” and the check box for “Log Each ISA and GS”.



The Run Inbound Map Option 2 tab shows the “Compliance NO EDI Text Blocking” check box.

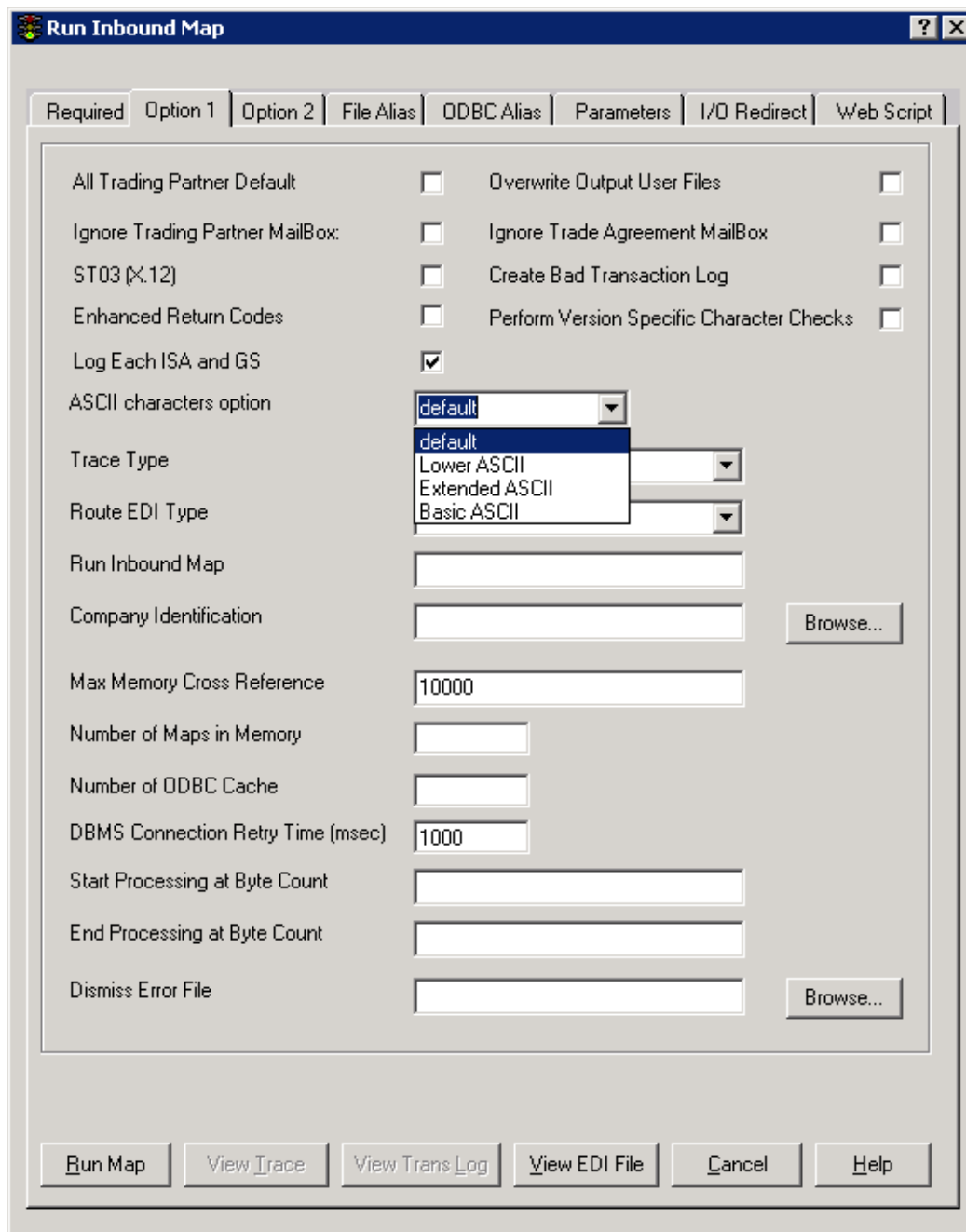


4. ECGateway

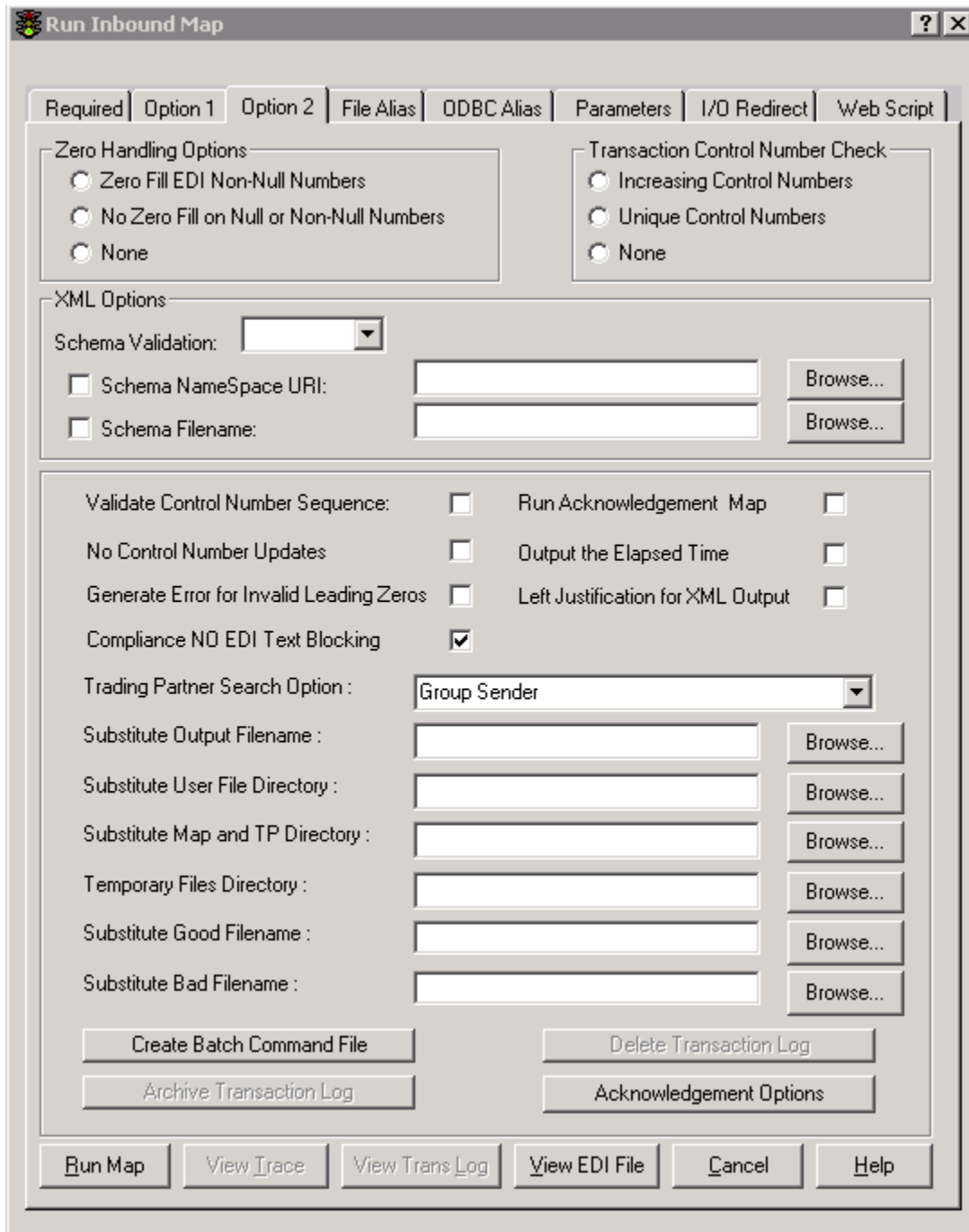
Run Map Options

GUI Runtime options

In addition to the command line runtime switches, the GUI can be used to perform the switch functions. The Run Inbound Map Option 1 tab shows the drop down menu for the “ASCII characters option” and the check box for “Log Each ISA and GS”.



The Run Inbound Map Option 2 tab shows the “Compliance NO EDI Text Blocking” check box.



5. Logging

New Fields added to TRLOG and Expanded Text Log for 999 Functional Acknowledgement

To meet current 999 transaction requirements, Six new TRLOG Fields and a corresponding six new System Variables have been added to ECMAP/ECRTP. These TRLOG fields and SysVars are: CTX_FIELDVAL , CTX_LINENO, CTX_SEGNAME, CTX_ELEMENT, CTX_SUBELEMENT, and CTX_REPEATNO. The expanded Translog.in and translog.out expanded log text files have also been modified by adding these fields on the end of the translog line.

The new variables are used in a Write Log command, where the user must check the new "Include Context (CTX) Variables" check box.

The CTX error codes that are used in the 999 transaction are:

Code	Output Value	Element	Description
2302	2	IK403	Condition Required Element Missing
2310	10	IK403	Exclusion Condition
2340	16	IK304	Implementation Dependent Required Segment Missing
2343	19	IK304	Implementation Not Used Segment Present
2347	113	IK403	Implementation Dependent Not Used Data Present
4568	10	IK403	Exclusion Condition
6053	110	IK403	Implementation Not Used Data Present

6. Appendix A TRLog table definition

The following table describes the TRLog table, columns, and usage for the data in both the XML and nonXML scenarios. This table is the same as in the 5.2.1 version of the EDI products, with the addition of the six new columns for Context (CTX) usage appended to the end of the table.

Name	Type	Chars	Description	XML Log Field Specs
AFLD	SQL_INTEGER	10	Auto increment field (Auto Number)	same
RUN_ID	SQL_BIGINT	9	Runtime ID - loaded from internal run id number that was passed in as a -id parameter. This run id is also loaded once into the SYS_RUN_ID system variable, but the value of SYS_RUN_ID is not used for TRLOG. The run id always numeric from initial parameter value	same
TYP	SQL_VARCHAR	1	Type flag: <ul style="list-style-type: none"> • H - Header (ST) • T - Trailer (SE) • D -detail messages between ST and SE • U - user write log command 	H record written at root element start, and T record written at end root element
RUN_DATE	SQL_TIMESTAMP	14	Runtime date - loaded from the SYS_DATE and SYS_HHMMSS system variables (SYS_DATE and SYS_HHMMSS are loaded from the system time.)	same
ACKBY_DATE	SQL_TIMESTAMP	14	Date by which an acknowledgement must be made <ul style="list-style-type: none"> • Inbound - Before each log write, if TYP is H, date is loaded from the INT_HEAD_DATE and INT_HEAD_TIME system variables, which are loaded from envelope. If TYP is not H, then null date. • Outbound - Before each log write, if TYP is H and an acknowledgement was requested, date is calculated from today's date plus the value in the DAYS, HOURS, and MINUTES fields in the tradstat database. If TYP is not H, then null date. 	Always NULL Date
TRANS_CODE	SQL_VARCHAR	2	Transaction code <ul style="list-style-type: none"> • Inbound - SYS_TRCODE loaded from envelope. • Outbound - SYS_TRCODE loaded from parameter. 	SYS_TRCODE - not loaded - should be empty
TRANS_NAME	SQL_VARCHAR	6	Transaction name - loaded from the SYS_TRANS system variable, which is loaded from internal transaction code before each log write. Message Type MSH 00009	SYS_TRANS Not loaded - should be empty
TPTNER_ID	SQL_VARCHAR	35	Code used to identify the trading partner Loaded from the SYS_TRADNO system variable. <ul style="list-style-type: none"> • Inbound - TPTNER_ID is looked up in trading partner database based on lookup criteria and data in EDI envelope. • Outbound - TPTNER_ID is loaded from an application file. 	Initially ALL TP loaded Value will change if XML Trade Partner rule executed.

New Features in Sybase EDI Server 5.2.2

VERSION	SQL_VARCHAR		Version of EDI standard used in the map Loaded from the X12_VERSION system variable. <ul style="list-style-type: none"> • Inbound - X12 version loaded from the envelope • Outbound - Loaded from tradstat table • Version/Release/Industry Identifier Code - GS08 • Message Version Number - UNH S009 0052 • Message Release Number - UNH S009 0054 	Not loaded. Should be empty for XML
ISA_TYPE	SQL_VARCHAR		EDI standard used by this trading partner in this transaction <ul style="list-style-type: none"> • Inbound - Loaded from INT_VERSION system variable, which is loaded from the EDI envelope • Outbound - loaded from the ISA_TYPE system variable, which is loaded from tradstat table • Interchange Control Version Number - ISA12 • Syntax Identifier - UNB S001_0001 • Syntax Version Number UNB S001 0002 	Not loaded.
INTERCHANG	SQL_VARCHAR		Interchange code - loaded from the INT_HEAD_NUM system variable <ul style="list-style-type: none"> • Inbound - Value taken from the EDI envelope. • outbound - Loaded from the ISA control number taken from trading partner table and incremented • Interchange Control Number - ISA13 • Interchange Control Count - UNB S004 0020 	Loaded with left justified MAP NAME if there was a - m MAPNAME on command line.
GROUP_NO	SQL_VARCHAR		Group number - loaded from the FUNC_GP_NUM system variable <ul style="list-style-type: none"> • Inbound - Value taken from EDI envelope • Outbound - Loaded from GS control number taken from the tradstat table and incremented • Functional Group Header Control Number - GS06 • Batch Control ID - BHS 00091 	Not loaded
TRANS_NO	SQL_VARCHAR		Transaction number - loaded from the TRANS_CTRL_NUM system variable <ul style="list-style-type: none"> • Inbound - Value taken from the EDI envelope • Outbound - loaded from 1000 * (GS control number taken) + transaction count 	Not loaded
APP_RCV_CD	SQL_VARCHAR		Code must be used to identify the group level receiver on outbound maps and sender on inbound maps. Loaded from the APP_RECV_CODE system variable. <ul style="list-style-type: none"> • Inbound - Loaded from EDI envelope. • Outbound - Loaded from Tradepartner. • Application Receiver Code - GS03 • Interchange Receiver Internal Sub-ID - UNB 0003 0046 • Receiving Facility - MSH 00006 • File Receiving Facility - FHS 00072 • Batch Receiving Facility - BHS 00086 	Not Loaded

New Features in Sybase EDI Server 5.2.2

APP_SND_CD	SQL_VARCHAR	35	Code must be used to identify the group level sender on outbound maps and receiver on inbound maps. Loaded from the APP_SEND_CODE system variable. <ul style="list-style-type: none"> • Inbound - Loaded from EDI envelope. • Outbound - Loaded from Tradepartner. • Application Sender Code - GS02 • Interchange Sender Internal Sub-ID - UNB S002 0042 • Sending Facility - MSH 00004 • File Sending Facility - FHS 00070 • Batch Sending Facility - BHS 00084 	Not Loaded
RECV_CODE	SQL_VARCHAR	35	Code must be used to identify the interchange level receiver on outbound maps and sender on inbound maps. Loaded from the RECV_CODE system variable. <ul style="list-style-type: none"> • Inbound - Loaded from EDI envelope. • Outbound - Loaded from Tradepartner. • Interchange Receiver ID Code - ISA08 • Interchange Receiver ID - UNB S003 0010 • Receiving Application - MSH 00005 • File Receiving Application - FHS 00071 • Batch Receiving Application - BHS 00085 	Not Loaded
SEND_CODE	SQL_VARCHAR	35	Code must be used to identify the interchange level sender on outbound maps and receiver on inbound maps. Loaded from the SEND_CODE system variable. <ul style="list-style-type: none"> • Inbound - Loaded from EDI envelope. • Outbound - Loaded from Tradepartner. • Interchange Sender ID Code - ISA06 • Interchange Sender ID - UNB S002 0004 • Sending Application - MSH 00003 • File Sending Application - FHS 00069 • Batch Sending Application - BHS 00083 	Not Loaded
RECV_QUAL	SQL_VARCHAR	4	Qualifier that specifies the type of code used to identify the interchange level receiver on outbound maps and sender on inbound maps. Loaded from the RECV_QUAL system variable. <ul style="list-style-type: none"> • Inbound - Loaded from EDI envelope. • Outbound - Loaded from Tradepartner. • Interchange Receiver ID Qualifier - ISA07 • Interchange Receiver ID Code Qualifier - UNB S003 0007 	Not Loaded
SEND_QUAL	SQL_VARCHAR	4	Qualifier that specifies the type of code used to identify the interchange level sender on outbound maps and receiver on inbound maps. Loaded from the SEND_QUAL system variable. <ul style="list-style-type: none"> • Inbound - Loaded from EDI envelope. • Outbound - Loaded from Tradepartner. • Interchange Sender ID Qualifier - ISA05 • Interchange Sender Code Qualifier - UNB S002 0007 	Not Loaded
ERRORS	SQL_BIGNT	10	Total Errors - loaded from internal system count of errors between ST and SE. The LOG_ERRS and TOT_ERRS system variables are loaded at same time. LOG_ERRS is the number of errors between ST and SE. TOT_ERRS is the total number of errors for the run.	total errors and parser warnings from processing XML Data file. Note a parser error will result in a return code of 3. A parser Fatal Error will be return code of 5. A parser warning will be return code of 1.

New Features in Sybase EDI Server 5.2.2

STAT	SQL_VARCHAR	1	<p>Status - loaded from internal count of total # errors. The LOG_STATUS system variable is also loaded at time of write log.</p> <ul style="list-style-type: none"> • W - SEG_ST, SEG_SE, User Write • T - Bad Tradepartner • S - Bad ISA, GS, or ST • U - Stop Run • A - Abort Trans • F - Fatal Error • E - Other Error Message 	<p>Status - loaded from internal count of total # errors. The LOG_STATUS system variable is also loaded at time of write log.</p> <ul style="list-style-type: none"> • W - SEG_Root Element, End Root Element, User Write • T - Bad Tradepartner • S - Parser Error • U - Stop Run • A - Abort Trans • F - Fatal Error • E - Other Error Message
BYTE_COUNT	SQL_BIGINT	10	<p>Count of the number of bytes between ST and SE - will be zero for every ST and increment until SE is written. The LOG_SIZE system variable is loaded with the byte count at time of write log.</p>	
DIR	SQL_VARCHAR	3	<p>Direction of map</p> <ul style="list-style-type: none"> • Outbound - OUT • Inbound - IN, CMP, PRT <p>There is no system variable</p>	<p>IN for map translation, CMP for compliance</p>
FLOW_LEVEL	SQL_VARCHAR	5	<p>Level of segment in flow - the LOG_LEVEL system variable is loaded at time of write log.</p>	<p>Compliance element depth translation map flow level</p>
RECORD_NAM	SQL_VARCHAR	10	<p>Record name - the LOG_RECNAME system variable is loaded at time of write log. Can be assigned by user in write log if the following conditions are present:</p> <ul style="list-style-type: none"> • Mapping is in progress and the field being mapped to or from is a record field, then this record name value is used. • Error occurs during a rule, then code attempts to identify the record name involved. • The field involved is a memory variable or a string variable, then the memory variable or string variable is written to the RECORD_NAM log field. • No record is associated with the error, then this field is blank. <p>This field cannot be written to by the user assigning a value to the system variable.</p>	<p>Same for map translation. Empty for compliance.</p>

New Features in Sybase EDI Server 5.2.2

RECORD_NO	SQL_VARCHAR	6	Record number - the LOG_READ_CNT system variable can be assigned by user in write log command. Otherwise the system variable and log are loaded and written at the same time. <ul style="list-style-type: none"> • Inbound - This is the line number being processed in the incoming EDI file. • Outbound - This field will contain the count that this record type has been read if the error message involves a record. Otherwise it will be zero. 	XML Source doc Line Count. XML horizontal line offset. SEG_COUNT count of XML elements, and RECORD_NO is line number in XML source document. TRANS_CNT is used for horizontal offset on each XML doc line
FIELD_NAME	SQL_VARCHAR	15	Field name - the LOG_FIELDNAME system variable can be assigned by user in write log command. Otherwise LOG_FIELDNAME and FIELD_NAME log field are loaded at the same time of write log. If mapping is in progress and the field being mapped to or from is a record field, then this field name value is used. If error occurs during a rule, then code attempts to identify the field name involved. This field cannot be written to by the user directly assigning a value to the system variable.	compliance empty. Same for map translation
SEGMENT	SQL_VARCHAR	3	Segment - can be assigned by user in write log command. <ul style="list-style-type: none"> • Inbound - Current segment name LOG_SEG • Outbound - Will have a value on write logs for SEG_ST and SEG_SE types and for errors that occur during direct mapping to EDI file. 	empty for XML
SEG_COUNT	SQL_INTEGER	10	Segment count - can be assigned by user in write log command. Count of segments between ST and SE inclusive. The SEGMENT_COUNT system variable loaded every time segment is written/read from EDI file.	count of elements in XML data file, where root element is segment_count 1. SEGMENT_COUNT system variable used.
ELEMENT	SQL_VARCHAR	2	Element - can be assigned by user in write log command. The LOG_ELEM system variable used.	
SUBELEM	SQL_VARCHAR	2	Sub-element - can be assigned by user in write log command. The LOG_SUBELEM system variable used.	
SEV_CODE	SQL_VARCHAR	2	Severity code - can be assigned by user in write log command. For all system error messages this code is a 1. For other system non-error messages, this code is 0.	
MSG_NO	SQL_VARCHAR	5	Message number - can be assigned by user in write log command. The LOG_MSG_NO system variable used.	MSG_NO same. 2083 warning, 2084 Error 2085 Fatal. 2090 Root Element 0701 End Root Element 0702.; Otherwise same.
MSG_TEXT	SQL_VARCHAR	100	Message text - can be assigned by user in write log command. The LOG_MSG_TXT is used	same

New Features in Sybase EDI Server 5.2.2

FILENAME	SQL_VARCHAR		<p>For ST segments:</p> <ul style="list-style-type: none"> • Outbound - FILENAME contains the current EDI outbound file name (can change based on tradstat and tp mailbox entries). Filename is not available as the system variable. • Inbound - The inbound EDI file name (always the same). Filename is not available as the system variable. For SE segments and inbound transactions: FILENAME. • Contains the current file name of any EDI OUT file names (can change based on tradstat, tp mailboxes and tradstat EDI_OUT field). • Consists of both complete path and file name. • Is available as the EDI_OUT_FILENAME system variable. 	<p>INBOUND XML always the XML Document name on root element on initial log "H" "ST" Start Document write. On "T" stop document write - loaded same as EDI. "SE Segment" write.</p> <p>For messages about attribute values, FILENAME will contain the XML attribute name.</p>
FIELDVAL	SQL_VARCHAR		<p>Field value - the LOG_VALUE system variable. Can be assigned by user in write log command. Otherwise the system variable and log field value are loaded at time of write to the log. If this is an error message, the value of the field code attempts to load the value of the record field, memory variable, or string constant in error.</p>	same
USER_IDENT	SQL_VARCHAR		<p>User-defined field - the SYS_USER_FIELD system variable. Log field is loaded from system variable.</p>	same
ACK_EXPECT	SQL_VARCHAR		<p>Flag that specifies whether a TA1 interchange-level acknowledgement is expected, only on outbound maps:</p> <ul style="list-style-type: none"> • 1 = TA1 acknowledgement requested • 0 = TA1 Acknowledgement not requested <p>The ACK_REQSTD system variable. Inbound - Loaded from EDI envelope ISA14.</p>	empty for XML
TR_ACK_TYP	SQL_VARCHAR		<p>Flag that specifies whether a group-level functional acknowledgement is expected, only on outbound maps.</p> <ul style="list-style-type: none"> • 1 = functional acknowledgment requested • 0 = functional acknowledgement not requested. <p>The corresponding system variable is TR_ACK.</p>	empty for XML
T_P_IND	SQL_VARCHAR		<p>Test/Production Indicator</p> <ul style="list-style-type: none"> • T - Test • P - Production • I - Information • D - Debug • 1-9 • Null <p>The TEST_IND system variable</p> <p>Inbound - loaded from the EDI envelope;</p> <p>Outbound - loaded from Tradstat</p>	loaded from system variable TEST_IND.

New Features in Sybase EDI Server 5.2.2

TRANS_CNT	SQL_INTEGER	10	Transaction count - no system variable. Inbound/Outbound - This log value is loaded from an internal count of the number of ST-SE teansactions read or written between SEG_GS and SEG_GE.	
FILEOFFSET	SQL_BIGINT	10	Number of Bytes File Offset - no system variable. Written from internal count of number of bytes read (inbound) or number of bytes written to the EDI file.	XML data offset in bytes.
RCOUNT	SQL_SMALLINT	1	Field for record manipulation - always set equal to 1 before log write	
SNDR_ROUTE	SQL_VARCHAR	14	Internal code used to identify the interchange level sender on outbound maps and receiver on inbound maps. Interchange Sender Internal ID - UNB S002 0008	same
SNDR_SUBID	SQL_VARCHAR	35	Internal sub-code used to identify the interchange level sender on outbound maps and receiver on inbound maps (EDIFACT Syntax 4 only) Interchange Sender Internal Sub-ID - UNB S002 0042	empty for XML
RCVR_ROUTE	SQL_VARCHAR	14	Internal code used to identify the interchange level receiver on outbound maps and sender on inbound maps. Interchange Receiver Internal ID - UNB S003 0014	empty for XML
RCVR_SUBID	SQL_VARCHAR	35	Internal sub-code used to identify the interchange level receiver on outbound maps and sender on inbound maps (EDIFACT Syntax 4 only) Interchange Sender Internal Sub-ID - UNB S003 0046	empty for XML
APPL_REF	SQL_VARCHAR	14	Name of the application messages contained in the EDIFACT UNB envelope. APPLICATION Reference - UNB S005 0026.	empty for XML
PROC_PRIOR	SQL_VARCHAR	1	Processing Priority Code UNB S005 0029	empty for XML
COMM_AGM	SQL_VARCHAR	35	Interchange Agreement Identifier UNB S005 0032	empty for XML
APP_SND_QL	SQL_VARCHAR	4	Qualifier that specifies the type of code used to identify the trading partner at the group level - as the sender on outbound maps and as the receiver on inbound maps. Application Sender ID/ID Code Qualifier - UNG S006 0007	empty for XML
APP_RCV_QL	SQL_VARCHAR	4	Qualifier that specifies the type of code used to identify the trading partner at the group level - as the receiver on outbound maps and as the sender on inbound maps. Application Sender ID/ID Code Qualifier - UNG S007 0007	empty for XML
ASSOC_CODE	SQL_VARCHAR	6	Association Assigned Code - • UNG S008 0057 • UNH S009 0057	empty for XML
APP_PSWD	SQL_VARCHAR	14	Application Password - UNG S008 0058	empty for XML

New Features in Sybase EDI Server 5.2.2

CLIST_VER	SQL_VARCHAR		(EDIFACT Syntax 4 only) Code List Directory Version 6 Number - UNH S009 0110	empty for XML
MSG_TYPE	SQL_VARCHAR		(EDIFACT Syntax 4 only) Message Type Sub-Function 6 Identifier - UNH S009 0113	empty for XML
RPT_NO	SQL_VARCHAR		(X12 version 4020 and later) Repeat number - this 2 field holds the number of a repeating element.	empty for XML
ST03	SQL_VARCHAR		Implementation Convention Reference - ST03 Value 35 to be used as the third element in the transaction on outbound X12 processing.	empty for XML
XML_TAG	SQL_VARCHAR	255	XML ELEMENT NAME	ONLY FOR NEW INBOUND XML
XML_URI	SQL_VARCHAR	80	URI for XML element name	ONLY FOR NEW INBOUND XML
XML_SCHEMA	SQL_VARCHAR	175	SCHEMA URI and File Name	ONLY FOR NEW INBOUND XML
CTX_FIELDVAL	SQL_VARCHAR		Loaded at time of write log. User must check the 99 new "Include Context (CTX) Variables" check box. IK404 element, system variable LOG_VALUE. TRLog field FIELDVAL	
CTX_LINENO	SQL_VARCHAR		Loaded at time of write log. User must check the 10 new "Include Context (CTX) Variables" check box. CTX03	
CTX_SEGNAME	SQL_VARCHAR		Loaded at time of write log. User must check the 3 new "Include Context (CTX) Variables" check box. CTX02	
CTX_ELEMENT	SQL_VARCHAR		Loaded at time of write log. User must check the 2 new "Include Context (CTX) Variables" check box. CTX0501	
CTX_SUBELEMENT	SQL_VARCHAR		Loaded at time of write log. User must check the 2 new "Include Context (CTX) Variables" check box. CTX0502	
CTX_REPEATNO	SQL_VARCHAR		Loaded at time of write log. User must check the 4 new "Include Context (CTX) Variables" check box. CTX0503	

6. Appendix B ASCII tables

ASCII Table and Description

ASCII stands for American Standard Code for Information Interchange. Computers can only understand numbers, so an ASCII code is the numerical representation of a character such as 'a' or '@' or an action of some sort. ASCII was developed a long time ago and now the non-printing characters are rarely used for their original purpose. Below is the ASCII character table and this includes descriptions of the first 32 non-printing characters. ASCII was actually designed for use with teletypes and so the descriptions are somewhat obscure. If someone says they want your CV however in ASCII format, all this means is they want 'plain' text with no formatting such as tabs, bold or underscoring - the raw format that any computer can understand. This is usually so they can easily import the file into their own applications without issues. Notepad.exe creates ASCII text, or in MS Word you can save a file as 'text only'

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	((72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051))	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[[123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135]]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL

Source: www.LookupTables.com

Extended ASCII Codes

128	Ç	144	É	160	á	176	☐	192	Ł	208	⌚	224	α	240	≡
129	ù	145	æ	161	í	177	☐	193	ł	209	⌞	225	β	241	±
130	é	146	Æ	162	ó	178	☐	194	ŧ	210	⌘	226	Γ	242	≥
131	â	147	ô	163	ú	179		195	†	211	⌚	227	π	243	≤
132	à	148	ö	164	ñ	180	†	196	—	212	⌞	228	Σ	244	∫
133	à	149	ò	165	Ñ	181	†	197	+	213	⌞	229	σ	245	∫
134	â	150	û	166	ª	182		198	‡	214	⌞	230	μ	246	+
135	ç	151	ù	167	º	183		199	‡	215	‡	231	τ	247	≈
136	ê	152	ÿ	168	¿	184	¶	200	⌚	216	‡	232	Φ	248	°
137	ë	153	Û	169	ƒ	185		201	⌞	217	∫	233	⊖	249	.
138	è	154	Ü	170	ƒ	186		202	⌚	218	∫	234	⊖	250	.
139	ï	155	•	171	½	187	¶	203	ŧ	219	■	235	δ	251	√
140	î	156	£	172	¾	188	⌚	204	†	220	■	236	∞	252	∞
141	ï	157	¥	173	∫	189	⌚	205	=	221	■	237	φ	253	²
142	Ä	158	£	174	«	190	∫	206	‡	222	■	238	e	254	■
143	Å	159	f	175	»	191	∫	207	±	223	■	239	∩	255	

Source: www.LookupTables.com