Release Bulletin
Mainframe Connect™ Server Option for IMS and MVS 15.0

Document ID: DC75214-01-1500-01

Last revised: August 2007

1. Accessing current release bulletin information

A more recent version of this release bulletin may be available on the Web. To check for critical product or document information added after the product release, use the Sybase® Product Manuals Web site.

❖ **Accessing release bulletins at the Sybase Product Manuals Web site**

1. Go to Product Manuals at http://www.sybase.com/support/manuals/.
2. Select a product and language and click Go.
3. Select a product version from the Document Set list.
4. Select the Release Bulletins link.
5. From the list of individual documents, select the link to the release bulletin for your platform. You can either download the PDF version or browse the document online.

2. Product summary

Enclosed is Mainframe Connect™ Server Option for IMS and MVS 15.0, which is a programming environment that enables you to develop mainframe applications that Open Client™ applications can execute.

The Server Option for IMS and MVS runs on an IBM z/Series or plug-compatible mainframe computer. It uses the TCP/IP communications protocol and an IMS TM or native MVS host transaction processor.

2.1 Hardware and software requirements

The following hardware and software are compatible with the Server Option for IMS and MVS 15.0:

- Hardware:
    - IBM mainframe: z/Series or plug-compatible
- Software:
    - IBM z/OS version 1.7 or later
    - IMS TM version 8.1 or later
    - IMS Connect version 2.1 or later
    - IBM TCP/IP

For planning, installation, and configuration information, see the Mainframe Connect Server Option for IMS and MVS *Installation and Administration Guide*.

2.2 Product media

Table 1 lists the Server Option for IMS and MVS 15.0 distribution media.

Table 1: Server Option for IMS and MVS 15.0 media

Media title	Media ID
Mainframe Connect Server Option for IMS and MVS 15.0	CD68187-55-1500-01
Mainframe Connect 15.0 Getting Started CD	CD00222-55-1500-01
Mainframe Connect 15.0 SyBooks™ CD	CD30045-55-1500-01

Note For directory and file information, see the *MEMLIST* member of the *JCL* data set for your Server Option 15.0 installation.

2.3 Product documentation

Table 2 lists all documentation for version 15.0 of the Server Option for IMS and MVS. Although not all documents are shipped as paper copy, all documents are available on the Web and on the Getting Started CD or the SyBooks CD.

Table 2: Server Option for IMS and MVS 15.0 documentation

Document title	Document ID
Mainframe Connect Server Option for IMS and MVS *Installation and Administration Guide*	DC34368-01-1500-01
Mainframe Connect *New Features*	DC00182-01-1500-01
Mainframe Connect *Overview Guide*	DC37572-01-1500-01
Mainframe Connect Server Option *Programmers Reference for COBOL*	DC36520-01-1500-01
Mainframe Connect Server Option *Programmers Reference for PL/1*	DC36560-01-1500-01
Mainframe Connect Server Option *Programmers Reference for Remote Stored Procedures*	DC35605-01-1500-01
Mainframe Connect Client Option and Server Option *Messages and Codes*	DC36450-01-1500-01
Mainframe Connect Server Option for IMS and MVS *Release Bulletin* (this document)	DC75214-01-1500-01

3. Special installation instructions

This section describes special installation instructions for the Server Option for IMS and MVS.

3.1 Registering the product license

This product requires a permanent authorization key. However, a temporary key that is valid for 30 days has been included within the order at installation time. To avoid interruption of operations, within those 30 days, call Customer Service at 1-800-8Sybase, (1-800-879-2273) select Option 3, and select Option 3 again to request a permanent key.

Be prepared to provide this information:

- Product name
- Order number
- Your machine type
- CPU Serial number of the machine
- Valid e-mail address.

Within seven business days, Sybase will send a permanent key to you by e-mail. Then, to register the product license, define the license key in the JCL member *IxHOST*, and run the SYGWXCPH configuration.

4. Changed functionality in this version

For information on new features and functionality in version 15.0 of the Server Option for IMS and MVS, see Mainframe Connect *New Features* (DC00182-01-1500-01).

5. Known issues

This section describes known issues in the Server Option for IMS and MVS 15.0.

5.1 License key messages

(CR #391723) If the license key is invalid for a client connection, a message is written to the IMS log, but no message is returned to the client.

5.2 Setting IBMUNICODE in SYGWXCPH

(CR #475273) If you set IBMUNICODE=N in SYGWXCPH, iso_1 code page translation is always used instead of the code page from the login.

6. Product compatibilities

For full functionality with the current release, use these Sybase products, as available at your site:

Table 3: Mainframe Connect Option release compatibility

Component	Release level
Mainframe Connect Client Option	15.0
Mainframe Connect Server Option	15.0
Mainframe Connect DirectConnect for z/OS Option	15.0

7. Documentation updates and clarifications

This section contains updates and clarifications for the Mainframe Connect product manuals.

7.1 Unicode support

The current version of the Server Option for IMS and MVS contains support for Unicode based on the Unicode support provided by IBM z/OS, including the conversion environment and conversion services. With the conversion environment and services installed and set up, the Server Option can convert character streams from one Coded Character Set Identifier (CCSID) to another. This support is provided in addition to the support for language and character sets offered in previous versions.

For details on Unicode, refer to IBM documentation.

7.1.1 Installing and enabling the IBM z/OS conversion environment and services

❖ **Installing Unicode support**

Use the following procedure to establish the necessary IBM z/OS conversion environment.

1. Create an *IMAGE* member using the IBM image generator utility, CUNMIUTL.
2. Copy the created image, member *CUNIMG01*, from *WORK.IMAGE* to *SYS1.PARMLIB*.
3. Load the image, member *CUNIMG01*, into z/OS using the SET UNI=01 command.
4. The DISPLAY UNI, ALL command displays the current active image and the character set conversions defined for that image.

To enable Server Option Unicode support, set the USEIBMUNICODE configuration parameter to Y. The USEIBMUNICODE is specified in the SYGWMCST macro in the SYGWXCPH customization module. The Server Option uses the newly defined unichar, univarchar, and unitext internal datatypes and performs conversions between UTF-8, UTF-16, and other CCSIDs.

For information on installing Unicode support for IBM z/OS, see "Support for Unicode Using Conversion Services" (SA22-7649-01).

7.1.2 SYGWXCPH customization module

The character set translation routines in the Server Option use tables in the SYGWXCPH customization module for the conversion of character sets. Because IBM Unicode support requires the CCSIDs of the character sets involved in conversion, the translation tables in the SYGWXCPH customization module and the SYGWMCXL macro have been modified to contain CCSIDs.

SYGWMCST

The USEIBMUNICODE parameter has been added to the SYGWMCST customization macro. The following are valid values for the USEIBMUNICODE parameter:

- Y – Use IBM support for character set conversions.
- N – Use the original Server Option support.

SYGWMCXL

The SYGWMCXL macro has been modified to include the following parameters, which are used for character conversion:

- CCSID – the CCSID for the character set.

- CHARSETTYPE – the character set type. A indicates ASCII, and E indicates EBCDIC.
- CHARSIZE – the maximum length of a character, between 1 and 4 bytes.
- PAD – the padding character. This parameter value depends on the type of character set. For ASCII, the padding character is 20. For EBCDIC, the padding character is 40.

Example 1

```
SYGWMCXL TYPE=ENTRY,
      CHARSET=cp939,CHARSETBYTES=D,
      CCSID=939,CHARTYPE=E,CHARSIZE=2,PAD=40
```

Example 2

```
SYGWMCXL TYPE=ENTRY,
      CHARSET=Russian,CHARSETBYTES=S,
      CCSID=1025,CHARTYPE=E,CHARSIZE=1,PAD=40
```

7.1.3 Datatypes for Unicode support

Components of Mainframe Connect have two new datatypes using the UTF-16 encoding of the Unicode character. The new unichar and univarchar datatypes are independent of the existing char and varchar datatypes but behave similarly. Like the char datatype, unichar is a fixed-width, non-nullable datatype. Like the varchar datatype, univarchar is a variable-width, nullable datatype. Each unichar or univarchar character requires 2 bytes of storage, so a unichar or univarchar column consists of 16-bit Unicode values.

7.1.4 Unicode support in the Server Option for IMS and MVS

The unichar, univarchar, and unitext datatypes have been added for Unicode support in the Server Option. These three datatypes are mapped to TDS_LONGBINARY with a user type of 34, 35, or 36, as shown in Table 4.

Table 4: Unicode datatype mappings

SQL datatype	TDS datatype	User type	Comment
unichar	TDS_LONGBINARY	34	Fixed-length UTF-16 data
univarchar	TDS_LONGBINARY	35	Variable-length UTF-16 data
unitext	TDS_LONGBINARY	36	UTF-16 encoded data

The Server Option has the following three datatypes to support unichar, univarchar, and unitext:

- TDSUNICHAR – Internal type 26
- TDSUNIVARCHAR – Internal type 27

- TDSUNITEXT – Internal type 28

Note Currently, there is no special support for TDSUNITEXT.

The following API calls have been changed in the Server Option to accommodate support for Unicode:

- TDPROPS
- TDESCRIB

TDPROPS

The TDPROPS API call maintains character set conversion properties.

Syntax

```
COPY SYGWCOB

01  TDSPROC             PIC S9(9) COMP.
01  RETCODE             PIC S9(9) COMP.
01  OPER                PIC S9(9) COMP.
01  PROPERTY            PIC S9(9) COMP.
01  VALUE               PIC S9(9) COMP.

CALL ‘TDPROPS’ USING TDSPROC RETCODE OPER PROPERTY
VALUE.
```

Arguments

Argument	Description
TDPROC	(I) Handle for this client-server connection.
RETCODE	(O) Variable in which the result of function execution is returned.
OPER	(I) To retrieve the property specified by *PROPERTY*, set to TDS-GET; to change the property specified by *PROPERTY*, set to TDS-SET.

Argument	Description
PROPERTY	(I). TDPROPS supports the following properties: • TDS_CLIENT_CCSID – defines the CCSID to which the Server Option converts server data. The value of this property defaults to the CCSID of the character set negotiated between the client and the server at login. • TDS_SERVER_CCSID – defines the CCSID to which the Server Option converts client data. The value of this property defaults to the CCSID of the character set negotiated between the client and the server at login. A UTF-8 connection is established in the case when the client-requested character set at login is UTF-8, and Unicode support is enabled for the Server Option. • TDS_PROG_CCSID – controls the conversion of data between the character sets of the server and the server application. For example, if a Server Option application sets TDS_PROG_CCSID to 1025 (Russian EBCDIC, CCSID=1025), and data received from the server is in UTF-8 (CCSID=1208), a parameter retrieved to a character variable in a TDRCVPRM call will be implicitly converted from CCSID=1208 to CCSID=1025. • TDS_DATA_CCSID – controls the conversion of metadata. For example, if an OSC application program sets TDS_DATA_CCSID property value to 1025 (Russian EBCDIC, CCSID=1025) and data received from the server is in UTF-8 (CCSID=1208), then the column names will be retrieved for the application program after being implicitly converted from CCSID=1208 to CCSID=1025.
VALUE	(I) The value of the property specified in *PROPERTY*.

Note For a connection established with the UTF-8 character set, the default values for the TDS_CLIENT_CCSID, TDS_SERVER_CCSID, TDS_PROG_CCSID, and TDS_DATA_CCSID parameters are 1208, 1208, 500, and 500, respectively.

TDPROPS may specify values for these properties any time after a connection has been established. The default values for these properties depend on the character set established for the connection at login.

Note The Server Option does not reset any CCSID property values set by an application program. Once an application changes a CCSID property value, the setting remains for all API calls until it is reset by the application.

Example 1

TDS_PROG_CCSID is set to 1208 (UTF-8), and both TDS_SERVER_CCSID and TDS_CLIENT_CCSID default to 1208. The server application program calls TDRCVPRM.

- When the server reads data from the client, the data is in UTF-8.
- When a Server Option application requests data from the server, the data is retrieved in UTF-8.

Example 2

TDS_PROG_CCSID is set to 1025 (Russian EBCDIC, CCSID=1025), and both TDS_SERVER_CCSID and TDS_CLIENT_CCSID default to 1208. The application calls TDRCVPRM:

- When data is read by the server from the client, the data is in UTF-8.
- When data is requested by the Server Option application from the server, the data is retrieved in EBCDIC CCSID=1025.

Example 3

TDS_PROG_CCSID is set to 939 (Japanese EBCDIC, CCSID=939), and both TDS_SERVER_CCSID and TDS_CLIENT_CCSID default to sjis (CCSID=943). The application calls TDRCVPRM:

- When data is read by the server from the client, the data is in sjis.
- When data is requested by the Server Option application from the server, the data is retrieved in EBCDIC CCSID=939.

TDESCRIB

The TDESCRIB API call now allows use of the TDSUNICHAR, TDSUNIVARCHAR, and TDSUNITEXT datatypes.

Table 5 lists new datatype conversions supported.

Table 5: New datatype mappings

Datatype	Datatype
TDSCHAR	TDSUNICHAR
TDSCHAR	TDSUNIVARCHAR
TDSCHAR	TDSUNITEXT
TDSVARCHAR	TDSUNICHAR
TDSVARCHAR	TDSUNIVARCHAR
TDSTEXT	TDSUNITEXT

7.2 Text and image data

Client applications send text and image data to the Server Option in a writetext stream. To process writetext stream data, a Server Option application cannot employ functions normally used to process parameter data. Instead, a Server Option application must use special text and image functions.

A Server Option application can send text or image data to a client application in either of the following ways:

- *data stream* – If the row of returned data contains one column of text or image data, the row may be sent as a data stream. The length of the data is between 0 and 2 gigabytes.
- *describe/send row* – If the row of returned data contains columns in addition to a text or image column, the text or image data may be sent using the describe/send row method. The length of the data cannot exceed 32KB.

The following subsections describe text and image issues for the Server Option:

- CS_IODESC structure
- Retrieving data from a client
- Returning data to a client
- Text and image functions

7.2.1 CS_IODESC structure

The CS_IODESC structure describes text or image data and is used to pass information between a Server Option application and the API functions that process this data.

The general structure for a CS_IODESC, regardless of programming language, is shown in Table 6.

Table 6: CS_IODESC structure

This field	Contains this information
IOTYPE	Indicates the type of input or output to perform. For text and image operations, *IOTYPE* always has the value CS_IODATA.
DATATYPE	The datatype of the data object. The only legal values for *DATATYPE* are TDSTEXT and TDSIMAGE.
LOCALE	Not used in the Server Option. Set this to NULL.
USERTYPE	Not used in the Server Option.
TOTAL_TXTLEN	In bytes, the total length of the text or image value.
OFFSET	Reserved for future use.
LOG_ON_UPDATE	Determines whether the update to this text or image value should be logged. This field is not used by the Server Option.
NAME	The name of the text or image column.
NAMELEN	In bytes, the length of *NAME*.
TIMESTAMP	The text timestamp of the column. A text timestamp marks the time of the last modification to a text or image column.
TIMESTAMPLEN	Not used by the Server Option.
TEXTPTR	A text pointer to a table row ID.
TEXTPTRLEN	In bytes, the length of *TEXTPTR*. This length is currently set at 16.

The CS_IODESC structure is defined in the *SYGWCOB* copy book for COBOL (under the name CS-IODESC) and in the *SYGWPLI* INCLUDE member for PL/1.

When receiving text or image data from a client application, a Server Option application invokes the TDINFTXT function with the ACTION parameter set to TDS_GET. The Server Option application must provide the correct text or image DATATYPE field value *before* TDINFTXT is invoked so that the Server Option can translate incoming text data. Only the value of the TOTAL_TXTLEN field is provided by TDINFTXT here.

When sending text or image data to a client application, the Server Option application also invokes the TDINFTXT function with the ACTION parameter set to TDS_SET. The Server Option application must describe the text or image data to be sent to the client by providing values for the appropriate CS_IODESC fields *before* TDINFTXT is invoked.

7.2.2 Retrieving data from a client

A writetext stream retrieved from a client application is handled as bulk data by the Server Option application.

An application processes incoming text or image data in two steps:

1. The TDINFTXT function retrieves a description of the text or image data and places the description in a CS_IODESC structure. The TDINFTXT function call returns information including the total length of incoming data. This length enables the Server Option application to determine whether the data should be retrieved in one unit or in sections. The Server Option application also determines the size of the buffer that must be allocated to store the incoming data. TDINFTXT is invoked with the ACTION parameter set to TDS_GET. The DATATYPE field of the CS_IODESC structure must be provided by the Server Option application before TDINFTXT is invoked. See “TDINFTXT” for details on this function.
2. The TDGETTXT function retrieves the incoming text and image data from the client application in the specified section size and stores the data in the specified buffer. See “TDGETTXT” for details on the TDGETTXT function.

Note A call to TDINFTXT must always precede a call to TDGETTXT. The TDGETTXT routine must be called until all text has been read from the client.

Table 7 illustrates the sequence of API function calls for retrieving text or image bulk data from the client.

Table 7: API function call sequence for data retrieval

Function	Action performed
TDSQLLEN	Determines the length of the incoming writetext string
TDSRCVSQL	Retrieves a writetext string. The TDSRCVSQL function call receives a writetext bulk command, which indicates that text or image bulk data follows. The writetext bulk command occurs in the following format: writetext bulk <*object_name*> <*text_pointer*> timestamp = <*time_stamp*> [with log | without log] The parameters of the writetext bulk command are as follows: • <*object_name*> is the name of the object name to which data is to be sent. • <*text_pointer*> is a text pointer. • <*time_stamp*> indicates the value of the timestamp parameter. • The text *with log* or *without log* is not used by the Server Option. For example: `writetext bulk SYBASE.au_txt.TXT 0xa1a0bbd014a6d005060e016a20400100 timestamp = 0x0000000000000000 with log` The manner in which the writetext bulk parameters are used depends on the Server Option application and on the destination of the incoming text and image data.
TDSNDDON	Notifies the sender that the SQL string has been received. Use the connection option of TDS_ENDREPLY to change the communication state from *send* to *receive*.
TDINFTXT (using TDS_GET)	Returns the length of the entire text or image data stream. The Server Option translates incoming data based on the value of the DATATYPE field in the CS_IODESC structure.
TDGETTXT	Retrieves a section of the text or image data stream. TDGETTXT is invoked in a loop until all incoming data is retrieved.
TDSNDDON	Notifies the sender that all data has been received.

7.2.3 Returning data to a client

A Server Option application sends text or image data to a client application in one of two ways, depending on the number of columns in the data row.

If there is one text or image column in the row to be sent, the Server Option application proceeds as follows:

1. Using the TDESCRIB function, the Server Option application describes the format in which the client receives the text or image column.
2. Optionally you can use the TDSETUDT function to set the user-defined datatype for the text or image column.

3. The Server Option application invokes the TDINFTXT function with the ACTION parameter set to TDS_SET to indicate the total length of the returning data.
4. The Server Option application invokes the TDSNDTXT function to send the data to the client in sections.

Table 8 illustrates the sequence of API function calls for sending text or image bulk data to the client.

Table 8: API function call sequence for sending bulk data only

Function	Action performed
TDESCRIB	Describes the text or image column to be sent to the client.
TDSETUDT (optional)	Sets the user-defined datatype for the column.
TDINFTXT (using TDS_SET)	Describes the text or image column to the Server Option. The Server Option application provides values for the CS_IODESC fields before invoking the TDINFTXT function. The TDINFTXT function is invoked once for every row that is to be sent to the client.
TDSNDTXT	Sends a section of the text or image data stream. The TDSNDTXT function is invoked in a loop until all the data for a given row is sent to the client.
TDSNDDON	Notifies the client that all data has been sent.

If there are other columns in addition to the text and image data in the row to be sent, the Server Option application proceeds as follows:

1. Using the TDESCRIB function, the Server Option application describes the format in which the client receives a column of data. The Server Option application invokes the TDESCRIB function once for each column of data.
2. The Server Option application invokes the TDINFTXT function with the ACTION parameter set to TDS_SET to provide text pointer and timestamp information. The Server Option application invokes the TDINFTXT function once for each text or image column in a row.
3. The Server Option application transfers the data to the client application using the TDSNDROW function, which is invoked once for each row of data. The text or image column size must not exceed 32KB.

Table 9 illustrates the sequence of API function calls for sending rows in which there are other columns in addition to the text or image data columns.

Table 9: API function call sequence for sending row data of varied column datatypes

Function	Action performed
TDESCRIB	Describes a column to be sent to the client. The TDESCRIB function is invoked once for each column of data to be sent to the client.

Function	Action performed
TDINFTXT (using TDS_SET)	Describes a text or image column to the Server Option. The Server Option application provides values for the CS_IODESC fields before invoking the TDINFTXT function. The TDINFTXT function is invoked in two nested loops, once for every text or image column in a row to be sent to the client, and once for every row to be sent to the client.
TDSNDROW	Sends a row of data to the client. The TDSNDROW function is invoked in a loop for every row of data to be sent to the client and preceded by a number of TDINFTXT calls describing the text and image columns in a row.
TDSNDDON	Notifies the client that all data has been sent.

7.2.4 Text and image functions

The Server Option provides three new functions: TDINFTXT, TDGETTXT, and TDSNDTXT. These functions can be invoked from within a Server Option application written in COBOL or PL/1. The TDINFTXT, TDGETTXT, and TDSNDTXT functions are described in the following sections using COBOL syntax.

TDINFTXT

Function

Sets or gets a description of text or image data.

Syntax

```
01 TDPROC          PIC S9(9) USAGE COMP SYNC.
01 RETCODE         PIC S9(9) USAGE COMP SYNC.
01 ACTION          PIC S9(9) USAGE COMP SYNC.
01 ITEM-NUMBER     PIC S9(9) USAGE COMP SYNC.
01 CS-IODESC FROM SYGWCOB


CALL 'TDINFTXT' USING TDPROC, RETCODE, ACTION, ITEM-
NUMBER, CS-IODESC.
```

Arguments

Argument	Description
TDPROC	(I) Handle for the client/server connection. The value here must be the same value specified in the associated TDACCEPT function call. The *TDPROC* handle corresponds to the connection and command handles in Open Client™ Client-Library.
RETCODE	(O) Variable to which the result of function execution is returned. The value of this variable is one of the codes listed below under “Returns.”

Argument	Description
ACTION	(I) Action to be taken by this call. *ACTION* is an integer variable that indicates the purpose of this call. Assign *ACTION* one of the following symbolic values: • TDS_GET (1) – the Server Option updates the CS_IODESC structure with the total length of the text or image data to be read from the client. Typically, this is followed by a call to the TDGETTXT function. The Server Option application must set the DATATYPE field in the CS-IODESC structure to TDSTEXT or TDSIMAGE before invoking TDGETTXT. • TDS_SET (2) – the Server Option sets internal Server-Library structures to describe a text or image data object. The TDINFTXT call updates a text or image column with the information contained in CS-IODESC. The Server Option application must describe the column using TDESCRIB before TDINFTXT is invoked.
ITEM-NUMBER	(I) The column number of the column being described. The first column in a row is column 1. This parameter is ignored when *ACTION* is TDS_GET.
CS-IODESC	(I) A pointer to the CS-IODESC for the application.

Comments

- TDINFTXT is used to describe text or image columns for sending a result row or retrieving a parameter.
- If *ACTION* is TDS_GET, TDINFTXT must be called prior to the first or only call to TDGETTXT for a row.
- If *ACTION* is TDS_SET, TDINFTXT must be called for each text or image datatype column in a row before TDSNDTXT or TDSNDROW is called.
- Text and image data is transferred to the client using either TDSNDTXT or TDSNDROW.

Returns

The *RETCODE* argument can contain any of the following values:

- TDS_OK (0)
- TDS_INVALID_PARAMETER (-4)
- TDS_INVALID_DATA_TYPE (-171)
- TDS_ILLEGAL_REQUEST (-5)
- TDS_INVALID_LENGTH (-173)
- TDS_RESULTS_COMPLETE (500)
- TDS_WRONG_STATE (-6)
- TDS_CONNECTION_FAILED (-4998)
- TDS_CONNECTION_TERMINATED (-4997)

See also

Related functions:

- TDSNDTXT
- TDGETTXT

TDSNDTXT

Function

Sends a subsequent part of the text or image data stream to the client.

Syntax

```
01 TDPROC                   PIC S9(9) USAGE COMP SYNC.
01 RETCODE                  PIC S9(9) USAGE COMP SYNC.
01 HOST-VARIABLE-NAME       PIC X(n).
01 BUFLEN                   PIC S9(9) USAGE COMP SYNC.


CALL ‘TDSNDTXT’ USING TDPROC, RETCODE, HOST-VARIABLE-
NAME, BUFLEN.
```

Arguments

Argument	Description
TDPROC	(I) Handle for the client/server connection. The value here must be the same value specified in the associated TDACCEPT function call. The *TDPROC* handle corresponds to the connection and command handles in Open Client Client-Library.
RETCODE	(O) Variable to which the result of function execution is returned. The value of this variable is one of the codes listed below under “Returns.”
HOST-VARIABLE-NAME	(I) Application program variable that contains data for this column.
BUFLEN	(I) The size in bytes of the buffer containing the data.

Comments

- TDSNDTXT is used when sending a single column of text or image data to the client.
- The Server Option application must always call TDINFTXT prior to the first call to TDSNDTXT for the data stream, in order to set the total length of the data to be sent. The application then calls TDSNDTXT to send a part of the data. TDSNDTXT is called as many times as there are sections of data in the data stream.
- The item being sent to the client must have previously been described using TDESCRIB.

- A Server Option application can also write text and image data to a client using TDSNDROW. TDSNDTXT allows the application to send the data in sections, whereas the standard TDSNDROW method requires that all the data in the column be sent in one piece.
- A column sent with TDSNDTXT must be of type text or image.
- The Server Option treats text and image data streams identically except for character set conversion, which is only performed on text data.

Returns

The *RETCODE* argument can contain any of the following values:

- TDS_OK (0)
- TDS_ILLEGAL_REQUEST (-5)
- TDS_INVALID_VAR_ADDRESS (-175)
- TDS_CANCEL_RECEIVED (-12)
- TDS_WRONG_STATE (-6)
- TDS_INVALID_LENGTH (-173)
- TDS_CONNECTION_TERMINATED (-4997)

See also

Related functions:

- TDGETTXT
- TDINFTXT

TDGETTXT

Function

Reads a subsequent part of a text or image datastream from the client.

Syntax

```
01 TDPROC                    PIC S9(9) USAGE COMP SYNC.
01 RETCODE                   PIC S9(9) USAGE COMP SYNC.
01 HOST-VARIABLE-NAME        PIC X(n).
01 BUFLEN                    PIC S9(9) USAGE COMP SYNC.
01 OUTLEN                    PIC S9(9) USAGE COMP SYNC.


CALL ’TDSNDTXT’ USING TDPROC, RETCODE, HOST-VARIABLE
NAME, BUFLEN.
```

Arguments

Argument	Description
TDPROC	(I) Handle for the client/server connection. The value here must be the same value specified in the associated TDACCEPT function call. The *TDPROC* handle corresponds to the connection and command handles in Open Client™ Client-Library.
RETCODE	(O) Variable to which the result of function execution is returned. The value of this variable is one of the codes listed below under "Returns."
HOST-VARIABLE-NAME	(I) Application program variable to receive a subsequent part of the incoming text or image client data.
BUFLEN	(I) The size in bytes of the buffer containing the data.
OUTLEN	(I) The length in bytes of the data received.

Comments

- TDGETTXT is used to read bulk data from the client. The bulk data can be of type text or image.
- TDGETTXT must be called until all of the bulk data has been read from a client. The Server Option application must keep track of the data that remains to be read.
- A column read with TDGETTXT must be of type text or image.
- A Server Option application must call TDINFTXT prior to the first call to TDGETTXT for the data stream. The application then calls TDGETTXT to retrieve a section of data. TDGETTXT is called as many times as are necessary to read in the whole stream.
- The Server Option application must set the CS_IODESC DATATYPE field to TDSTEXT or TDSIMAGE before invoking the TDINFTXT and TDGETTXT functions. In the case in which DATATYPE is set to TDSTEXT, the Server Option translates the character set for the client data before sending the data to the Server Option application.

Returns

The *RETCODE* argument can contain any of the following values:

- TDS_OK (0)
- TDS_INVALID_VAR_ADDRESS (-175)
- TDS_INVALID_LENGTH (-173)
- TDS_ILLEGAL_REQUEST (-5)
- TDS_CONNECTION_FAILED (-4998)
- TDS_CONNECTION_TERMINATED (-4997)

See also

Related functions:

- TDSNDTXT
- TDINFTXT

7.3 API return code

The following information applies to Mainframe Connect Client Options and Mainframe Connect Server Options *Programmers Reference* guides.

Mainframe Connect 12.6 introduced the following API return code:

```
-275 (TDS_LICENSE_CHECK_FAIL
```

Description: This return code is issued when the product license information is found to be incorrect:

Action: Update the SYGWLKEY macro in the SYGWXCPH table with a valid license key.

8. Technical support

Each Sybase installation that has purchased a support contract has one or more designated people who are authorized to contact Sybase Technical Support. If you have any questions about this installation or if you need assistance during the installation process, ask the designated person to contact Sybase Technical Support or the Sybase subsidiary in your area.

9. Other sources of information

Use the Sybase Getting Started CD, the SyBooks CD, and the Sybase Product Manuals Web site to learn more about your product:

- The Getting Started CD contains release bulletins and installation guides in PDF format, and may also contain other documents or updated information not included on the SyBooks CD. It is included with your software. To read or print documents on the Getting Started CD, you need Adobe Acrobat Reader, which you can download at no charge from the Adobe Web site using a link provided on the CD.
- The SyBooks CD contains product manuals and is included with your software. The Eclipse-based SyBooks browser allows you to access the manuals in an easy-to-use, HTML-based format.

Some documentation may be provided in PDF format, which you can access through the PDF directory on the SyBooks CD. To read or print the PDF files, you need Adobe Acrobat Reader.

Refer to the *SyBooks Installation Guide* on the Getting Started CD, or the *README.txt* file on the SyBooks CD for instructions on installing and starting SyBooks.

- The Sybase Product Manuals Web site is an online version of the SyBooks CD that you can access using a standard Web browser. In addition to product manuals, you will find links to EBFs/Maintenance, Technical Documents, Case Management, Solved Cases, newsgroups, and the Sybase Developer Network.

  To access the Sybase Product Manuals Web site, go to Product Manuals at http://www.sybase.com/support/manuals/.

9.1 Sybase certifications on the Web

Technical documentation at the Sybase Web site is updated frequently.

❖ **Finding the latest information on product certifications**

1. Point your Web browser to Technical Documents at http://www.sybase.com/support/techdocs/.
2. Click Certification Report.
3. In the Certification Report filter select a product, platform, and timeframe and then click Go.
4. Click a Certification Report title to display the report.

❖ **Finding the latest information on component certifications**

1. Point your Web browser to Availability and Certification Reports at http://certification.sybase.com/.
2. Either select the product family and product under Search by Base Product; or select the platform and product under Search by Platform.
3. Select Search to display the availability and certification report for the selection.

❖ **Creating a personalized view of the Sybase Web site (including support pages)**

Set up a MySybase profile. MySybase is a free service that allows you to create a personalized view of Sybase Web pages.

1. Point your Web browser to Technical Documents at http://www.sybase.com/support/techdocs/.
2. Click MySybase and create a MySybase profile.

9.2 Sybase EBFs and software maintenance

❖ **Finding the latest information on EBFs and software maintenance**

1. Point your Web browser to the Sybase Support Page at http://www.sybase.com/support.
2. Select EBFs/Maintenance. If prompted, enter your MySybase user name and password.
3. Select a product.
4. Specify a time frame and click Go. A list of EBF/Maintenance releases is displayed.

   Padlock icons indicate that you do not have download authorization for certain EBF/Maintenance releases because you are not registered as a Technical Support Contact. If you have not registered, but have valid information provided by your Sybase representative or through your support contract, click Edit Roles to add the “Technical Support Contact” role to your MySybase profile.
5. Click the Info icon to display the EBF/Maintenance report, or click the product description to download the software.

10. Accessibility features

This document is available in an HTML version that is specialized for accessibility. You can navigate the HTML with an adaptive technology such as a screen reader, or view it with a screen enlarger.

Mainframe Connect Server Option for IMS and MVS and the HTML documentation have been tested for compliance with U.S. government Section 508 Accessibility requirements. Documents that comply with Section 508 generally also meet non-U.S. accessibility guidelines, such as the World Wide Web Consortium (W3C) guidelines for Web sites.

Note You might need to configure your accessibility tool for optimal use. Some screen readers pronounce text based on its case; for example, they pronounce ALL UPPERCASE TEXT as initials, and MixedCase Text as words. You might find it helpful to configure your tool to announce syntax conventions. Consult the documentation for your tool.

For information about how Sybase supports accessibility, see Sybase Accessibility at http://www.sybase.com/accessibility. The Sybase Accessibility site includes links to information on Section 508 and W3C standards.