

Ribo User's Guide

Ribo

Version 2.0

Document ID: 37542-01-0200-01

Last revised: October 1999

Copyright © 1989-1999 by Sybase, Inc. All rights reserved.

This publication pertains to Sybase database management 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, the Sybase logo, ADA Workbench, Adaptable Windowing Environment, Adaptive Component Architecture, Adaptive Server, Adaptive Server Anywhere, Adaptive Server Enterprise, Adaptive Server Enterprise Monitor, Adaptive Server Enterprise Replication, Adaptive Server Everywhere, Adaptive Server IQ, Adaptive Warehouse, AnswerBase, Anywhere Studio, Application Manager, AppModeler, APT Workbench, APT-Build, APT-Edit, APT-Execute, APT-FORMS, APT-Translator, APT-Library, Backup Server, ClearConnect, Client-Library, Client Services, Data Pipeline, Data Workbench, DataArchitect, Database Analyzer, DataExpress, DataServer, DataWindow, DB-Library, dbQueue, Developers Workbench, Direct Connect Anywhere, DirectConnect, Distribution Director, E-Anywhere, E-Whatever, Embedded SQL, EMS, Enterprise Application Server, Enterprise Application Studio, Enterprise Client/Server, Enterprise Connect, Enterprise Data Studio, Enterprise Manager, Enterprise SQL Server Manager, Enterprise Work Architecture, Enterprise Work Designer, Enterprise Work Modeler, EWA, Gateway Manager, ImpactNow, InfoMaker, Information Anywhere, Information Everywhere, InformationConnect, InternetBuilder, iScript, Jaguar CTS, jConnect for JDBC, KnowledgeBase, MainframeConnect, Maintenance Express, MAP, MDI Access Server, MDI Database Gateway, media.splash, MetaWorks, MySupport, Net-Gateway, Net-Library, NetImpact, ObjectConnect, ObjectCycle, OmniConnect, OmniSQL Access Module, OmniSQL Toolkit, Open Client, Open ClientConnect, Open Client/Server, Open Client/Server Interfaces, Open Gateway, Open Server, Open ServerConnect, Open Solutions, Optima++, PB-Gen, PC APT Execute, PC DB-Net, PC Net Library, Power++, power.stop, PowerAMC, PowerBuilder, PowerBuilder Foundation Class Library, PowerDesigner, PowerDimensions, PowerDynamo, PowerJ, PowerScript, PowerSite, PowerSocket, Powersoft, PowerStage, PowerStudio, PowerTips, Powersoft Portfolio, Powersoft Professional, PowerWare Desktop, PowerWare Enterprise, ProcessAnalyst, Report Workbench, Report-Execute, Replication Agent, Replication Driver, Replication Server, Replication Server Manager, Replication Toolkit, Resource Manager, RW-DisplayLib, RW-Library, S Designor, S-Designor, SDF, Secure SQL Server, Secure SQL Toolset, Security Guardian, SKILS, smart.partners, smart.parts, smart.script, SQL Advantage, SQL Anywhere, SQL Anywhere Studio, SQL Code Checker, SQL Debug, SQL Edit, SQL Edit/TPU, SQL Everywhere, SQL Modeler, SQL Remote, SQL Server, SQL Server Manager, SQL SMART, SQL Toolset, SQL Server/CFT, SQL Server/DBM, SQL Server SNMP SubAgent, SQL Station, SQLJ, STEP, SupportNow, Sybase Central, Sybase Client/Server Interfaces, Sybase Financial Server, Sybase Gateways, Sybase MPP, Sybase SQL Desktop, Sybase SQL Lifecycle, Sybase SQL Workgroup, Sybase User Workbench, SybaseWare, Syber Financial, SyberAssist, SyBooks, System 10, System 11, System XI (logo), SystemTools, Tabular Data Stream, Transact-SQL, Translation Toolkit, UNIBOM, Unilib, Uninull, Unisep, Unistring, URK Runtime Kit for UniCode, Viewer, Visual Components, VisualSpeller, VisualWriter, VQL, WarehouseArchitect, Warehouse Control Center, Warehouse Studio, Warehouse WORKS, Watcom, Watcom SQL, Watcom SQL Server, Web Deployment Kit, Web.PB, Web.SQL, WebSights, WebViewer, WorkGroup SQL Server, XA-Library, XA-Server and XP Server are trademarks of Sybase, Inc. 9/99

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 their respective companies.

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., 6475 Christie Avenue, Emeryville, CA 94608.

Contents

CHAPTER 1	Using Ribo	1
	Starting Ribo	2
	Capturing data.....	3
	Syntax and parameters	3
	Examples.....	4
	Translating data	6
	Examples.....	6
	Using the Ribo GUI	7
	Using filters	10
	Known problems	13

Using *Ribo*

This document describes how to use the **Ribo** diagnostic utility.

Topics covered are:

Name	Page
Starting Ribo	2
Capturing data	3
Translating data	6
Using the Ribo GUI	7
Using filters	10
Known problems	13

What is *Ribo*?

The Ribo utility captures, translates, and displays the Tabular Data Stream™ (TDS) protocol flowing between a TDS client and TDS server. TDS clients include jConnect™ for JDBC™, isql, jisql, and Open Client™. TDS servers include Adaptive Server® Enterprise, Adaptive Server Anywhere, Adaptive Server IQ, and Open Server™.

For more information about TDS tokens, see the TDS 5.0 Functional Specification at <http://www.sybase.com/products/TDSfunctionalSpecForm.html>.

Ribo has three modes:

- Capture a TDS protocol stream to a file
- Translate a captured file to the TDS protocol's text representation
- Capture a TDS protocol's stream to a file and translate the protocol's text representation to the screen or a GUI window "on-the-fly"

Starting Ribo

Ribo ships with a UNIX shell script and a MS-DOS *.bat* file. Before you use **Ribo**:

- Give yourself execute permissions on the UNIX script. At a command prompt where the **Ribo** files are installed, type:

```
chmod +x Ribo
```

- Set the JAVA_HOME and RIBO_HOME environment variables. For example:

```
set RIBO_HOME=c:\jutils-2_0\ribo  
set JAVA_HOME=c:\jdk1.1.8
```


Capturing data

To use **Ribo** to capture TDS protocol data and save that data to a file, enter the following from a UNIX, Linux, or DOS command window:

```
Ribo <command_line_options>
```

Syntax and parameters

Ribo uses the following syntax and command line options.

```
Ribo [-l <listen_port>] [-s <server_host>] [-p <server_port>]
      [-c <capture_file_prefix>] [-t <trans_file_prefix>] [-x <charset>]
      [-gui] [-d] [-f <filter_file>] [-h]
```

Table 1-1: Ribo command line options

Parameter	Description	Default
-l	The port on which to listen.	5005
-s	The host name of the database server.	localhost
-p	The port number of the database server.	2638
-c	By default, Ribo captures TDS data to a file. This option lets you specify the prefix added to the generated file name of the captured data. This file can be used by technical support personnel to help you troubleshoot problems with your system.	<i>capX.tds</i> where “X” is the generated file name.
-t	Translates captured data to the TDS protocol’s text representation and saves it to a file. This option lets you specify the prefix added to the generated file name of the translated data.	<i>outX.tds</i> where “X” is the generated file name.

Parameter	Description	Default
-x	<p>Lets you specify the default character set conversion to use when dumping TDS files.</p> <ul style="list-style-type: none"> The character set you specify must match a character set in be one supported by Sybase. If Ribo finds character set information in the LOGIN record, that information overrides the charset you specify using -x. If you do not specify a character set using -x, and the TDS file contains no character set information, no character set translation is done. Text is written using the the server's default character set. <hr/> <p>Note For more information about character sets and character-set conversion, see Chapter 2 of the <i>jConnect for JDBC Programmer's Reference</i> or Chapter 7 of the <i>Sybase Adaptive Server Enterprise Version 12.0 Installation Guide</i>.</p>	n/a
-gui	<p>Invokes the Ribo GUI, which you can use to:</p> <ul style="list-style-type: none"> Specify command line parameters Start and stop the capturing process View translated data “on the fly” as it is being analyzed 	n/a
-d	<p>Displays translated data while the data is being captured. If you are using the Ribo GUI, the data displays in a separate window. If you are not using the GUI, the data displays on the screen.</p>	n/a
-f	<p>Lets you specify a user-defined, previously saved filter.</p>	n/a
-h	<p>Displays help on Ribo usage.</p>	n/a

Examples

To have **Ribo** listen on local machine port 2638 and forward the data to another machine—rubicon, port 2525—you would enter:

```
Ribo -l 2638 -s rubicon -p 2525
```

To have **Ribo** listen on a local machine port 4000, display the data in the GUI as it is being captured, and filter the data using a script you have created named `myscript.filter`, you would enter:

```
Ribo -gui -l 4000 -f myscript.filter -d
```

Translating data

To analyze captured TDS protocol data and save the text representation of that data to a file, enter:

```
Ribo <input_capture_file> <output_file>
```

where *<input_capture_file>* is the name of the file from which to analyze previously captured data and *<output_file>* is the file name under which to save the translated data. If you do not specify an output file, the translated data is sent to **stdout**.

The output file displays:

- SQL statements sent to the server
- Parameters sent to the server
- Results returned from the server

Examples

To have Ribo listen on local machine port 2638, translate the TDS protocol data on the fly as the data is being captured, you would enter:

```
Ribo -l 2638 -t
```

The data will be written to *outX.tds* where X matches *capX.tds*.

To have Ribo translate captured TDS protocol data, you would enter:

```
Ribo cap0.tds tds0.out
```

Using the *Ribo* GUI

To start the Ribo graphical user interface and listen on local machine port 2638, you would enter:

```
Ribo -gui -l 2638
```

When you include the **-gui** parameter, you see this screen:

Figure 1-1: Ribo GUI

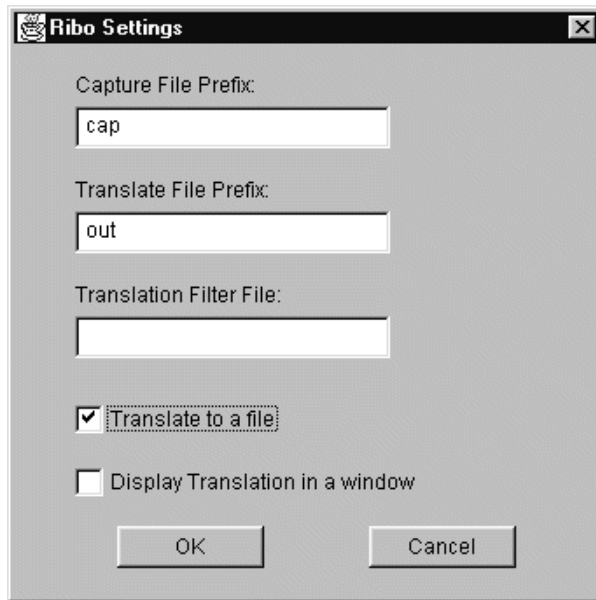


If you specified the listen port, server host, and server port when you started **Ribo**, these parameters are filled in for you on the GUI screen.

❖ **To capture and translate data:**

- 1 Select Preferences from the File menu. You see this dialog box:

Figure 1-2: Ribo Settings dialog box



- 2 Enter or change the values shown in the dialog box, which correspond to various command line parameters.

If you specified a value for these parameters when you started **Ribo** at the command line, those values display in this dialog box.

If you did not specify these parameters, the program displays the default values.

Capture File Prefix – Corresponds to the **-c** (*capture_file_prefix*) parameter. Specify the prefix added to the generated file name of the captured data.

Translate File Prefix – Corresponds to the **-t** (*trans_file_prefix*) parameter). Specify the prefix added to the generated file name of the translated data.

Translation Filter File – Corresponds to the **-f** (*filter_file*) parameter. Displays the file name of the filter you specified when you started **Ribo** at the command line. See Using Filters for more information.

Translate to a file – Also corresponds to the **-t** (*trans_file_prefix*) parameter. Check this box to translate captured data to the TDS protocol's text representation and save it to a file with the prefix you specified.

Display Translation in a window – Corresponds to the **-d** parameter. Check this box to display translated data in the GUI while the data is being captured.

See Table 1-1 on page 3 for more details about the command line parameters.

3 Click **OK**.

4 Click **Start Capture**.

To stop the capture process click **Stop Capture**. You must stop the capture process to change the host, ports, or Preferences.

5 When you finish, click End.

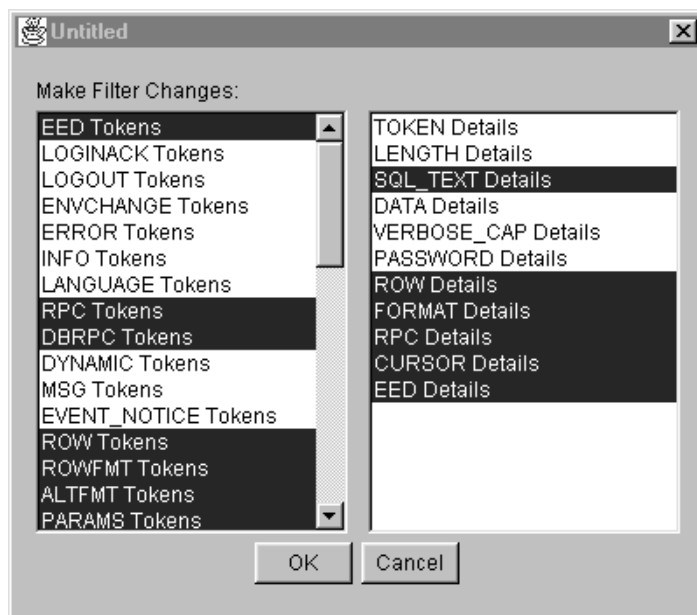
Using filters

You can create a filter that shows some or every detail of the TDS protocol data using the **-f** command parameter. If you do not specify a filter file name, the default filter is used. To see the default filter selections, start the **Ribo** GUI:

```
Ribo -gui -l 4000 -d
```

When the GUI displays, select Edit Filter from the File menu. You see a screen similar to this:

Figure 1-3: Default filter selections



The column on the left displays types of tokens; the column on the right displays details available for some or all tokens. Selected items are highlighted and represent the token details you want to see from the TDS protocol data.

Note Every token that occurs legally in TDS version 5.0 or later is recognized by **Ribo**. For more information about TDS tokens, see the TDS 5.0 Functional Specification at <http://www.sybase.com/products/TDSfunctionalSpecForm.html>.

The following table describes the items in the right column.

DETAIL	DESCRIPTION
TOKEN_DETAILS	Refers to information about the (single-byte) token itself, like its hex value and whether it is fixed or variable length. If you do not select (highlight) this detail, only the token name is dumped. Pertains to all tokens.
LENGTH_DETAILS	Refers to information about the various length fields found within a token, including the overall token length. Pertains to all tokens.
DATA_DETAILS	Refers to all details of the token beyond what is specified by TOKEN_DETAILS. This is what the TDS 5.0 Specification calls the “data stream” that follows the token itself. Pertains to all tokens.
VERBOSE_CAP_DETAILS	Specifies that the flags in a capability token are to be dumped in “verbose” format, which indicates the name of each flag and its value. If you do not select (highlight) this detail, the flags are dumped as hex. Pertains only to CAPABILITY tokens.
PASSWORD_DETAILS	Specifies that the password contained in the login record is to be dumped. If you do not select (highlight) this detail, the password is dumped. Pertains only to Login Record.
<i>Selecting the following details when you have not selected DATA_DETAILS will cause the details to be dumped anyway for certain tokens. That is, these details override DATA_DETAILS.</i>	
SQL_TEXT_DETAILS	Refers to the text of a SQL query. When you select (highlight) this detail, it overrides DATA_DETAILS. Pertains only to LANGUAGE tokens.
ROW_DETAILS	Refers to the row data of a row, parameter, Alt-row or key token. If you do not select (highlight) this detail, the data is not dumped. Specifying this detail overrides DATA_DETAILS. Pertains only to ALTROW, KEY, PARAMS, RPC, RETURN_VALUE and ROW tokens.

DETAIL	DESCRIPTION
FORMAT_DETAILS	Refers to the format data of a parameter format, row format, or alt-row format token. Specifying this detail overrides DATA_DETAILS. Pertains only to ALTFMT, PARAMFMT, ROWFMT, RPC and RETURN_VALUE tokens.
RPC_DETAILS	Overrides DATA_DETAILS for DBRPC tokens. Pertains only to DBRPC tokens.
CURSOR_DETAILS	Overrides DATA_DETAILS for all cursor tokens. Pertains only to CURCLOSE, CURDECLARE, CURDELETE, CURFETCH, CURINFO and CURUPDATE tokens.
EED_DETAILS	Overrides DATA_DETAILS for EED tokens. Pertains to EED tokens only.

❖ **To create a custom filter:**

- 1 Click on an item to highlight and select it; click again to deselect it.
- 2 Type the file name under which to save your custom filter. The name can be anything you want and doesn't require an extension; for example TDS_1.filter or just TDS1.
- 3 Click Save.
- 4 Click OK to save your changes. A dialog box displays where you can save the new filter.

To use a custom filter you have created, enter something similar to this at the command line:

```
ribo -l 4000 -gui -f TDS_1.filter -d
```

Known problems

This section describes problems that are scheduled to be fixed in a future release of Ribo.

- #197508 - DATETIMN conversion throws an exception.

When dumping a file containing a row token with a DATETIMN column, an exception is thrown. Ribo continues to function, but you never see the row token for that row. This is specific to JDK 1.1.8.

Index

C

Capturing data 3

D

Data 6

description 1

E

Examples 4, 6

F

Filters

 Ribo 10

filters 10

G

Graphical User Interface (GUI)

 Graphical User Interface 7

GUI, using the Ribo 7

J

JAVA_HOME, setting 2

K

known problems

 Known problems 13

P

Parameters 3

R

Ribo 1, 7, 13

 capturing data 3

 parameters and syntax 3

 translating data 6

 using the GUI 7

RIBO_HOME, setting 2

S

Setting

 JAVA_HOME 2

 RIBO_HOME 2

Syntax 3

T

translating

 Translating data 6

Translating data 6

U

Using the Ribo GUI 7

