



## Contents

# SQL Sniffer for Windows



The following Help Topics are available:

[Introduction](#)

[Disclaimer](#)

[About Blue Lagoon Software](#)

[SQL Sniffer menu commands](#)

[SQL Sniffer options dialog box](#)

[SQL Sniffer's sniffed data](#)

## Introduction

Every application under Windows 3.1 written to access SQL Server, whether in Microsoft Visual Basic, Power Builder, C, or Microsoft Excel does so through some library specific to the tool being used. These libraries in turn interface with Microsoft \_dbLib (W3DBLIB.DLL).

The version of **SQL Sniffer for Windows** you have purchased is to be used under Windows 3.1 and against SQL Server 4.2. The Sniffer allows you to *sniff* most \_dbLib API calls simultaneously. These \_dbLib APIs used in this application were carefully chosen because they are the most commonly used and provide the most useful information when debugging.

To make the most out of **SQL Sniffer for Windows**, you should have a copy of *The Microsoft SQL Server Programmer's Reference, V4.2* handy. This valuable reference lists all of the \_dbLib API along with the parameters passed and return values. This information will be needed when examining SQL Sniffer's *sniffed* data.

## Disclaimer

**SQL Sniffer for Windows**, Copyright © 1992-1994 by Blue Lagoon Software, is provided as-is without warranty of any kind. Blue Lagoon Software shall not be liable for any loss of profits, loss of use, interruption of business, nor indirect, special, incidental or consequential damages of any kind from the use of this product.

## About Blue Lagoon Software

**SQL Sniffer for Windows** was designed and developed by Blue Lagoon Software. Blue Lagoon Software is a Windows consulting and product development company with expertise in the following areas:

- Client/Server application development;  
Microsoft and Sybase SQL Server  
Visual Basic  
C and C++  
Excel macro language  
ODBC
- Multimedia Training systems  
Ultimedia
- Windows SDK and DDK development  
Experts in user interface design.  
Experts in GDI implementation.  
Experts in Windows 3.1 architecture.

**Blue Lagoon Software**  
**6659 Hesperia Avenue**  
**Reseda, Ca. 91335**

**Voice: (818) 807-2785**

**Fax: (818) 345-8905**

**BBS: (818) 343-8433**

**CServe: 70152,1601**

## SQL Sniffer menu commands



SQL Sniffer for Windows has four menus; Sniffer, Windows, Options, and Help.

### Sniffer menu commands

#### Sniffer On/Off

Toggles *sniff* mode. When SQL Sniffer is *sniffing*, the caption bar of the application will be set to "<< SQL Sniffer >>". When SQL Sniffer is not sniffing, the caption bar of the application will be set to "SQL Sniffer".

#### Exit

Exits SQL Sniffer.

### Window menu commands

#### Clear Window

Clears the SQL Sniffer window. Any output in the SQL Sniffer window will be cleared.

#### Options! menu command

Brings up the SQL Sniffer options dialog box. It is within this dialog box that *sniffing* options are setup.

### Help menu commands

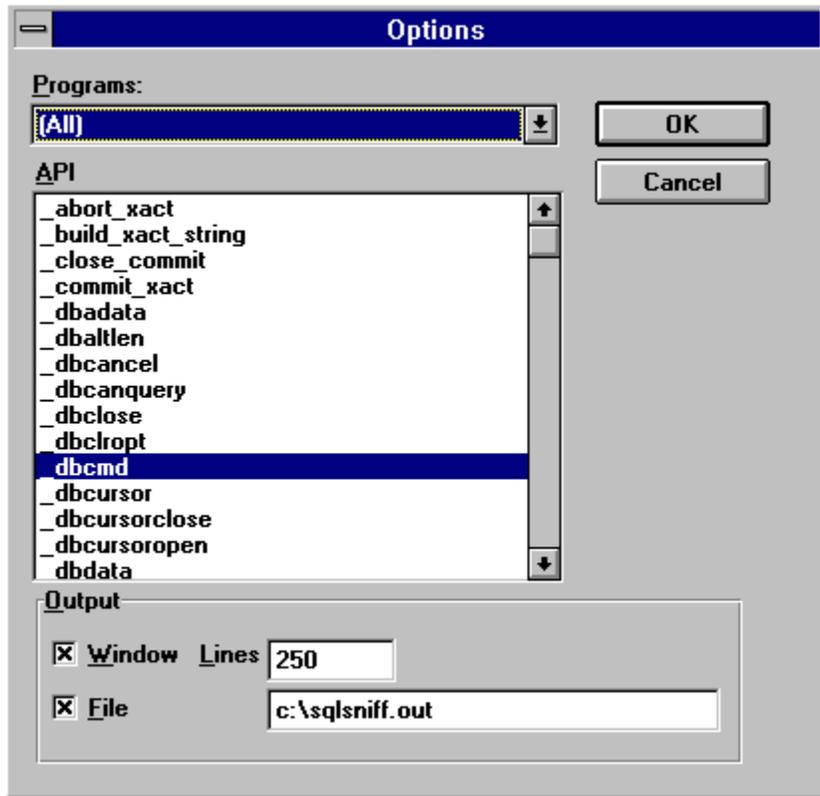
#### Contents

Brings up the contents portion of the help file.

#### About

Displays application about box.

## SQL Sniffer options dialog box



### Programs combo box

The Programs combo box contains a list of all programs currently running under Windows. To *sniff* a specific applications SQL accesses, first make sure that the application is currently running, then select it from this combo box. SQL Sniffer will only display `_dbLib` calls made by this application. The "(All)" option is designed to display `_dbLib` calls made by ALL applications.

### API list box

The API list box is a multiselection list box. Individual APIs are selected to indicate that they should be displayed by the Sniffer when they are called. This allows you to filter out the APIs you're not interested in seeing.

### Window check box

This check box should be selected to indicate that you want SQL Sniffer to display what it *sniffs* in the application area of the SQL Sniffer's application window.

### Lines entry field

This entry field must contain a number between 10 and 250. The number indicates the number of lines of *sniffed* data that the SQL Sniffer's application window will display at any given time.

### File check box

This check box should be selected to indicate that you want SQL Sniffer to create a log file which will contain all *sniffed* data. As SQL Sniffer *sniffs*, it will append its data in ASCII format to a log file whose pathname is specified in the adjacent entry field.

**OK pushbutton**

Accepts the changes made.

**Cancel pushbutton**

Cancels the changes made.

## SQL Sniffer's sniffed data

It is possible to have SQL Sniffer *sniff* data into both SQL Sniffer's application window and/or a log file in ASCII format. The application window is subject to a maximum of 250 lines of output while the log file has no imposed limits.

When a `_dbLib` API call is made, the execution time of the call in *seconds:milliseconds*, the application that made the call, the name of the API along with the parameters and return value are displayed in the application window and/or saved to the log file in the following format:

```
000:016 SQLADMIN: _dbcmd(9332, "select * from foo"), RETVAL=1
```

*Note: the execution time of the call is the amount time in seconds:milliseconds it took \_dbLib to execute the call. So, for example, it took 16 milliseconds to execute the above \_dbcmd call. This performance information is especially useful for commands such as \_dbsqlsend, \_dbsqlok, and \_dbsqlexec because they send SQL commands to SQL Server for execution. By measuring their performance, it is possible to determine how long queries and stored procedures take to execute.*

According to the *Microsoft SQL Server Programmer's Reference V4.2*, `_dbcmd` has two parameters, a handle to the database connection, and a long pointer to the SQL command to execute. The return value is an integer. As we can see, the `_dbcmd` call was made by SQLADMIN, the database handle is 9332, and the command being executed is "select \* from foo". The return value is 1 and according to the *SQL Server* documentation, this means that the API call was successful.

