

# ODBC Spy Contents

[Overview](#)

[ODBC Spy Menus](#)

[Spying with ODBC Spy](#)

[Spying on an Application-Driver Connection](#)

[Spying on an Application \(Emulating a Driver\)](#)

[Spying on a Driver \(Emulating an Application\)](#)

[Using ODBC Spy with the CodeView Debugger](#)

## Overview

ODBC Spy is a utility included with the ODBC SDK with which you can debug your drivers and applications. With ODBC Spy you can:

- Intercept and copy ODBC commands being sent from an application to an ODBC driver (spy on the application-driver connection). This is useful in determining how an ODBC connection fails.
- Emulate the actions of an ODBC driver receiving requests from an application (spy on an application). This is useful for debugging an application.
- Emulate an application and make requests to an ODBC driver (spy on a driver). This is useful for debugging a driver.

ODBC Spy is a low-level debugging tool. While ODBC Test enables you to test the functionality of your driver, ODBC Spy enables you to determine the specific point at which your driver may be failing.

Both 16- and 32-bit versions of ODBC Spy are available. If you are running the Windows NT operating system, the ODBC SDK Setup program copied both versions to your system. If you are running Windows 3.1, only the 16-bit version was installed.

**Note** The 16-and 32-bit versions of ODBC Spy may be run simultaneously under Windows NT. However, only one instance of either one may be run at a time.

## ODBC Spy Menus

The ODBC Spy menus are as follows:

<b>Menu</b>	<b>Description</b>
<u>F</u> ile	Loads and displays a log file.
<u>E</u> dit	Copies log file information to the Clipboard.
<u>S</u> py	Captures ODBC function calls or emulates drivers and applications.
<u>D</u> ebug	Adds and removes breakpoints, runs the log file, or switches to a debugger.
<u>H</u> elp	Provides Help and version information.

## File Menu Commands

The File menu contains commands to open, save, and print the log file, and to quit ODBC Spy.

Command	Description
Clear	Clears the ODBC Spy workspace if no operation is currently active; otherwise, a message box appears. Choose the OK button to clear, or choose the Cancel button to preserve the workspace.
Log File	Saves tracing of ODBC functions in a log file.  <b>Note</b> Because you can start logging to a file at any time with this menu item, the log file you create might not contain enough calls to ODBC functions to successfully emulate an application or driver.
Log on Screen	Prints tracing output to the screen.
Save to File	Saves the contents of the ODBC Spy workspace to a file. If a log file has already been specified, this command writes to this file without prompting.
Exit	Disconnects from any data sources and closes ODBC Spy.

## **Edit Menu Commands**

The Edit menu enables you to copy text from the main window to the Clipboard.

<b>Comm and</b>	<b>Description</b>
Copy	Copies selected text onto the Clipboard. This command is disabled when no text is selected.

## Spy Menu Commands

The Spy menu commands toggle between on and off modes.

Command	Description
Capture/ Capture Off	Traces ODBC function calls between an application and a driver. In Off mode, turns off tracing and saves the current log file.
Emulate Driver/ Emulate Driver Off	Plays back the saved ODBC function calls to a requesting application. In Off mode, turns off emulate driver mode and closes the current log file.
Emulate App/ Emulate App Off	Plays back the saved ODBC function calls to a driver. In Off mode, turns off emulate application mode and closes the current log file.

## Debug Menu Commands

Command	Description
Add Break	Adds a breakpoint to the selected line.
Remove Break	Removes the breakpoint from the selected line.
Remove All Breaks	Removes all breakpoints from the current file.
Step	Executes the current line (same as pressing the F10 key).
Go	Continues to the next breakpoint, or to the end of the log file if no more breakpoints are set (same as pressing the F5 key).
Enter Debugger	Switches from ODBC Spy to a debugging utility, such as Microsoft CodeView. (Disabled if ODBC Spy is not running under a debugger.)

## Help Menu

Comm and	Description
Help	Displays the ODBC Spy Help file.
About	Displays version information.

## **Spying with ODBC Spy**

In order to spy on an application or a driver, you must first spy on an application-driver connection and save the ODBC calls made by the application to a log file. You then use the information in the log file to emulate either a driver or an application.

**Important** While in use, ODBC Spy temporarily modifies the ODBC.INI file. In the event of a system failure, the changed information will be retained. You may want to first create a backup copy of your ODBC.INI file before running ODBC Spy.

## Spying on an Application-Driver Connection

As stated in the previous section, before you can emulate a driver or an application, you must first record a set of ODBC function calls between an application and its driver, and save it to a log file. You do this with the Capture command. The general procedure is as follows:

► To capture a set of ODBC function calls

1. From the Spy menu, choose Capture. Select a data source compatible with your driver. If you choose the Log To File box, ODBC Spy prompts you for a log file.

**Note** If you will use the information captured here in a driver or application emulation, you must save it to a log file.

2. Start the application that will send commands to the driver.
3. Use the application to access information through your selected data source. This can be done using automated test scripts or user-driven events (by hand). ODBC Spy displays and records all requests to the driver.

**Note** If you plan to use ODBC Spy to emulate a driver, and you are not using automated test scripts, write down the sequence of operations you have the application perform. You will need this exact sequence when you use ODBC Spy to emulate a driver.

4. When you have finished recording the set of function calls between the application and its driver, save the log file by choosing Capture Off from the Spy menu. The ODBC Spy workspace clears, and the open log file is closed.
5. Repeat steps 1 through 4 for all the sets of function calls you want to capture. Specify a different log file each time so you can use them later for your emulations.

Once you have finished recording function calls, you can use ODBC Spy to emulate either the driver or the application involved in one of the original recordings.

## Understanding Capture Results

The general structure of the information written to the ODBC Spy log file and workspace is:

```
ODBC Function
  argument value
.
.
.
  RETCODE
```

You can use ODBC Spy to identify the actual values being passed between the functions called by the application and driver. The left-justified text is the name of the application's function call, and the indented text is the arguments of the call and the driver's response. If you look up the syntax for each captured function in the *Microsoft ODBC SDK Programmer's Reference*, you will find that for each function, the number of indented lines in the captured information matches the total number of arguments (including the return code) specified for the function.

For example, the following results are written to a log file:

```

SQLAllocEnv
    0x00010000
    SQL_SUCCESS
SQLAllocConnect
    0x00010000
    0x00010000
    SQL_SUCCESS

```

In the first statement, the environment handle (*phenv*, 0x00010000) and the return value (SQL\_SUCCESS), were generated by the driver. The same process is true for the second statement: the **SQLAllocConnect** statement produced the environment handle (*henv*), the connection handle (*phdbc*), and the return code (RETCODE), in that order.

**Note** Because ODBC Spy resides between the Driver Manager and a driver, it captures calls to ODBC functions made by the application and calls made solely by the Driver Manager. For example, after the Driver Manager loads the driver, it calls **SQLGetInfo** to determine which version of ODBC the driver supports.

### A Capture Example

The following procedure leads you through the capture of the function calls that occur between ODBC Test and an existing driver. The log file produced here will be used as the basis for the emulation examples, later in this chapter.

- To capture function calls between ODBC Test and a driver
- 1. From the Spy menu, choose Capture. The Capture dialog box appears.  
Ordinarily, you would choose a data source that uses the driver you would be testing. Because this example uses ODBC Test, you can choose any available data source. (If there is no compatible data source for your application, you have to create one. For information on creating a data source, see the *Microsoft ODBC SDK Guide*.)

**Note** You can use only one data source at a time.

- 2. Select the Log To File check box. Then choose the OK button. The Log File dialog box appears.
- 3. Select a directory and type a filename for the log file to which the captured information is to be written. (The data recorded to this file will be the basis for your emulations.) Then choose the OK button. If the filename you select already exists in the directory you propose, ODBC Spy will prompt you to confirm overwriting the old file. Choose the Yes or No button as appropriate. The Log File dialog box disappears.
- 4. Start ODBC Test. Position it and ODBC Spy for convenient simultaneous viewing in the Windows operating system workspace.
- 5. From the ODBC Test Connect menu, choose Full Connect. The SQL Data Sources dialog box appears.
- 6. Select the same data source you selected in step 1. Then choose the OK button.

The SQL Data Sources dialog box disappears, and ODBC Test sends the functions needed to connect to the data source. It sends **SQLAllocEnv**, **SQLAllocConnect**, **SQLConnect**, six **SQLGetInfo** statements, and one **SQLAllocStmt** request.

The ODBC Spy workspace displays all ODBC function calls to the data source's driver. Each function is left-justified, with the arguments and return code indented. This information is written to the log file and to the screen simultaneously. (For details on the response information, see the previous section, "Understanding Capture Results.") When the last statement is executed, ODBC Test displays a message in the Output window indicating that it has successfully connected to the data source.

**Note** You can disable the ODBC Spy workspace screen refresh by choosing Log on Screen from the ODBC Spy File menu. When the current functions finish, you can choose Log on Screen from the File menu again, and the next function will be highlighted.

7. From the ODBC Test Connect menu, choose Full Disconnect. The ODBC Spy workspace displays the execution of **SQLDisconnect**, **SQLFreeConnect**, and **SQLFreeEnv**. ODBC Test displays a message indicating that the connection has closed.
8. From the ODBC Spy Spy menu, choose Capture Off. The Spy workspace clears, and the open log file is saved and closed.
9. From the ODBC Test File menu, choose Exit. The ODBC Test utility closes.

In this example, the ODBC Test connection was closed first. It is not necessary to close the connection before you quit ODBC Spy; you may close ODBC Spy while a connection is open. This will not affect the communication link between the application and the driver. However, if you close ODBC Spy first, remember that all logging will stop.

## Spying on an Application (Emulating a Driver)

To emulate a driver, you need to have already captured a set of function calls between the application with which you want ODBC Spy to interact, and the driver you want ODBC Spy to emulate.

**Note** Emulating a driver is particularly useful when the data source connection is expensive (remote), and your main object is to debug an application.

The general procedure for emulating a driver is as follows:

► To emulate a driver

1. Capture a set of function calls between the application with which you want ODBC Spy to interact, and the driver you want ODBC Spy to emulate. This can be done using automated test scripts or user-driven events (by hand). ODBC Spy displays and records all requests to the driver. If you are running the application by hand, write down the sequence of operations you have the application perform. You will need this exact sequence when you want ODBC Spy to emulate the driver.
2. From the ODBC Spy Spy menu, choose Emulate Driver. The Emulate Driver dialog box appears.
3. Choose the data source compatible with the driver you want ODBC Spy to emulate. Then choose the OK button. The Emulate Driver dialog box disappears, and the Log File dialog box appears.
4. Select the log file that contains the set of function calls you want the driver to use. Then choose the OK button. The Log File dialog box disappears. The contents of the log file are copied into the ODBC Spy workspace. The first function (and its arguments and return values) is highlighted.
5. Run the application that was used to create the current log file.
6. Using the sequence of operations you recorded in step 1 as a reference, have the application call the same ODBC functions as it had originally.

**Note** The application must call the same functions as when the log file was first created, and it must call them in the same order for a successful test.

As each function is called from the application, it is highlighted in the ODBC Spy workspace. If the application fails during the emulation, the errant function will be highlighted.

7. From the ODBC Spy Spy menu, choose Emulate Driver Off. The ODBC Spy workspace is cleared and the open log file is closed.

**Note** During driver emulation, ODBC Spy can only handle a single pass of the set of function calls. If you try to use the same log file again without reloading it, ODBC Spy will generate a replay error and close the application (in this case ODBC Test).

If you want to run the emulation sequence again, repeat steps 2 through 7.

## Driver Emulation Example

The following procedure leads you through the emulation by ODBC Spy of the driver used in the capture example earlier in this chapter. It uses the log file produced from the brief session between ODBC Spy and ODBC Test.

► To emulate a driver receiving requests from ODBC Test

1. From the ODBC Spy Spy menu, choose Emulate Driver. The Emulate Driver dialog box appears.
2. Choose the same data source you used during the capture example. Then choose the OK button. The Emulate Driver dialog box disappears, and the Log File dialog box appears.
3. Select the log file that contains the set of ODBC function calls you used for the capture example. Then choose the OK button. The Log File dialog box disappears. The contents of the log file are copied into the ODBC Spy workspace.
4. Run ODBC Test.

5. From the ODBC Test Connect menu, choose Full Connect. This is the first command executed during the capture example. If you don't choose this tool first, an error will occur.

As each request is sent from ODBC Test, the current line in ODBC Spy (highlighted in blue) is advanced. ODBC Spy simply matches the ODBC Test request with that of the current line, sets argument values as specified, and returns the last line of the indented information to ODBC Test. When ODBC Spy returns the values for the **SQLAllocStmt** statement, ODBC Test displays a message in the Output window indicating a successful connection.

6. From the ODBC Test Connect menu, choose Full Disconnect. This is the second operation done during the capture example. The current line in the ODBC Spy workspace advances as before, and the sequence finishes.
7. From the ODBC Spy Spy menu, choose Emulate Driver Off. The ODBC Spy workspace is cleared and the open log file is closed.

Once you have reached the end of the log file, you cannot replay the log file a second time. If you want to run the emulation sequence again, repeat the procedure.

## Spying on a Driver (Emulating an Application)

To emulate an application, you need to have already captured a set of ODBC function requests between the driver with which you want ODBC Spy to interact, and the application you want ODBC Spy to emulate. The general procedure is as follows:

► To emulate an application

1. Capture a set of function calls from the application you want ODBC Spy to emulate to the driver with which you want ODBC Spy to interact. (You do not need to write down the operations the application performs, because the application itself will not be involved in the playback.)
2. From the ODBC Spy Spy menu, choose Emulate App. The Emulate Application dialog box appears.
3. Choose the data source that uses the driver used by the application you want ODBC Spy to emulate. Then choose the OK button. The Emulate Application dialog box disappears, and the Log File dialog box appears.
4. Select the log file that contains the set of ODBC function calls you want the driver to use. Then choose the OK button. The Log File dialog box disappears. The contents of the log file are copied into the ODBC Spy workspace. The first function (and its return values) is highlighted.
5. If you want to pause while playing back the captured function calls, set the breakpoints now. (For more information, see "Setting and Removing Breakpoints" later in this section.)
6. Start the playback by choosing Go from the Debug menu (or press F5). Playback will stop at the line with the first breakpoint. Or, you can execute the current line by choosing Step from the Debug menu (or press F10).  
ODBC Spy calls the functions in the log file, and compares the arguments and return code returned by the driver to those saved in the log file. Any deviation from the saved values results in an error.
7. If you have run ODBC Spy from within the CodeView® debugger, you can switch back to the debugger while emulating an application by choosing Enter Debugger from the Debug menu, or by pressing CTRL+ALT+SYS RQ.
8. Once you have reached the end of the log file, you must restart the log file before you can play ODBC function calls back again. To restart the log file, choose the Go command from the Debug menu and choose the OK button when ODBC Spy prompts you to restart. To play back the function calls again, return to step 6.
9. When you are finished, choose Emulate App Off from the Spy menu. The ODBC Spy workspace clears and the log file is closed.

## Setting and Removing Breakpoints

You can set as many breakpoints as you want in the current log file; note that breakpoints are not saved when you close the log file.

► To set a breakpoint

- Double-click the line in the ODBC Spy workspace on which you want to set a breakpoint. Or, move to the line using the PAGE UP, PAGE DOWN, HOME, END, or arrow keys, and then press F9. The line changes from black to red. When the breakpoint line becomes the current line, its color changes to purple.

► To remove a breakpoint

- Double-click the line in the ODBC Spy workspace on which you want to clear a breakpoint. Or, move to the line using the PAGE UP, PAGE DOWN, HOME, END, or arrow keys, and then press F9. The line changes from red to black.

► To remove all existing breakpoints

- From the Debug menu, choose Remove All Breakpoints. All the breakpoints you have set are cleared.

## Application Emulation Example

The following procedure leads you through the emulation by ODBC Spy of the application used in the capture example earlier in this chapter. It uses the log file produced from the brief session between ODBC Spy and ODBC Test.

- To emulate ODBC Test making requests to a driver
1. From the ODBC Spy menu, choose Emulate App. The Emulate Application dialog box appears.
  2. Choose the same data source you used during the capture example. Then choose the OK button. The Emulate Application dialog box disappears, and the Log File dialog box appears.
  3. Select the log file that contains the set of ODBC function calls you used for the capture example. Then choose the OK button. The Log File dialog box disappears. The contents of the log file are copied into the ODBC Spy workspace.
  4. Set a breakpoint on the first **SQLGetInfo** function by double-clicking it with the mouse. ODBC Spy highlights the function in red, marking it as a breakpoint. (If you set the breakpoint on the wrong statement by mistake, just double-click it again with the mouse, and the breakpoint will clear.)
  5. From the Debug menu, choose Step, or press F10. ODBC Spy calls the first function in the log file and compares the values returned by the driver to those saved in the log file. Any deviation from the saved values results in an error.  
The current line is advanced to the next command, which becomes highlighted in blue. You can step through as many commands as you want. You do not need to have any breakpoints set to do single-stepping.
  6. From the Debug menu, choose Go, or press F5. ODBC Spy executes all the functions from (and including) the current function, to (but not including) the function on which you have set the breakpoint. The breakpoint becomes the current line, and becomes highlighted in purple, indicating that a breakpoint line is now the current line.
  7. From the Debug menu, choose Go, or press F5. ODBC Spy calls all remaining functions in the log file to the driver, and compares the returned values to those saved in the log file. Any inconsistencies will cause an error.
  8. From the Spy menu, choose Emulate App Off. The ODBC Spy workspace clears, and the open log file is closed.

## Using ODBC Spy with the CodeView Debugger

You can use ODBC Spy in conjunction with the Microsoft CodeView debugger when debugging your driver DLL.

**Note** The ODBC Spy Debug menu Enter Debugger command will work with any debugger that is activated by the **DebugBreak()** function. For more information on the **DebugBreak()** function, see your Windows operating system Software Development Kit documentation.

The general procedure for using ODBC Spy with the CodeView debugger is as follows:

- ▶ To use ODBC Spy with CodeView
- 1. In ODBC Spy, capture a set of ODBC function calls between an application and your driver (or a driver that will produce a log file that a working version of your driver will be able to use).
- 2. Quit ODBC Spy and start Microsoft CodeView on ODBCSPY.EXE.
- 3. Load your driver DLL with the /L switch. You can set CodeView breakpoints at this time.
- 4. From within CodeView, switch to ODBCSPY.EXE by typing **Go** in the CodeView command window, or by pressing F5. The system switches to ODBC Spy.
- 5. From the ODBC Spy Spy menu, choose Emulate App. The Emulate Application dialog box appears.
- 6. Choose the same data source you used during the capture in step 1. Then choose the OK button. The Emulate Application dialog box disappears, and the Log File dialog box appears.
- 7. Select the log file that contains the set of ODBC function calls you captured in step 1. Then choose the OK button. The Log File dialog box disappears. The contents of the log file are copied into the ODBC Spy workspace.
- 8. Set an ODBC Spy breakpoint on or near the point of the log file where you want to invoke the CodeView debugger. ODBC Spy highlights the function in red, marking it as a breakpoint.
- 9. From the ODBC Spy Debug menu, choose Go. ODBC Spy executes the requests against your driver, and stops at the breakpoint you set in step 8.
- 10. Now you are ready to enter CodeView. From the ODBC Spy Debug menu, choose Enter Debugger or press CTRL+ALT+SYS RQ. The system switches to the CodeView debugger.
- 11. Within CodeView, perform debugging operations as needed to debug your driver DLL.
- 12. When you are ready to switch back to ODBC Spy, type **Go** in the CodeView command window, or press F5. The system switches to ODBC Spy.
- 13. Repeat steps 8 through 12 as needed to complete the log file, or from the Spy menu, choose Emulate App Off to close the open log file.

**Note** For more information on using the Microsoft CodeView debugging utility, see your CodeView documentation.

