



# Winspector 1.0

Welcome to the Winspector help. Check out the various help sections for information on how Winspector is used and what it can do for you.

Winspector  
Version 1.0  
8/02/93

Noriel Chang Reissig  
Av. República do Líbano, n. 434  
PELOTAS-RS CEP 96055-710  
BRAZIL

## Monitor Help Sections

### Overview

### Winspector Limitations

### Using Winspector

### Using the Winspector API functions

### Commands

Winspector is based in a program called Monitor by Brandon Booth (vnet: ocanuc at msnvm1), wich is IBM internal use only. Monitor is a nice program, when I was in IBM it were helpfull like a debugging tool. So, here is a shareware version.

Please feel free to upload the shareware version of Winspector 1.0 to any electronic bulletin board or to give copies to other users (be sure to upload the complete version with all files, see them in overview). But please, do not give it away altered or as part of another system.

The author makes no warranties of any kind, express or implied, including without limitation, any warranties of merchantability and/or fitness for a particular purpose. Furthermore, the author shall not be liable for any damage to data or property which may be directly or indirectly caused by the use of Winspector 1.0. In no event will the author be liable for any damages, including lost profits, or for any claim by any other party.

The registration fee for Winspector is US\$40.00 + US\$4.00 Shipping & Handling. You are entitled to evaluate Winspector 1.0 for a period of up to 30 days. If after that time you intend to

continue to use the program you must register it. If you do not intend to continue to use the program you must delete it from your system.

You can access an order form which you can use to register Winspector in the file ORDFORM.TXT which you can use to contact the author. Alternatively a handwritten note will do just fine. Use check to pay, please make all checks payable to **Noriel Chang Reissig** in U.S. dollars. **U.S.A. Bank checks only!** Unfortunately, we cannot cash foreign private checks here.

The shareware copy of Winspector that you have is not restricted in any way. However there are significant benefits to registering your copy.

Upon registration you will receive a copy of the most up-to-date version of Winspector, without the start-up screens and a DLL version of the Winspector's API, so you could use Winspector with any language that support DLLs.

And apart from all that, the knowledge that you are helping to support the Shareware concept, a distribution system that brings you quality software at realistic prices.

## Overview

---

Winspector is a useful alternative to using `MessageBox()` as a debugging technique because the flow of control of your application is not interrupted (as it would be to respond to the `MessageBox`) and no overlay of your application's window takes place (which can cause unwanted messages to be placed on your message queue). And because the debug information is displayed in a listbox you can see the history of the debugging session. Winspector were mainly designed for C/C++ Windows programmers.

### Winspector Package Contents

#### Tools

WINSPECT.EXE	the executable module
WINSPECT.HLP	Windows help file
WINSPECT.H	C header file
WINSPECS.LIB	small model library
WINSPECM.LIB	medium model library (registered only)
WINSPECL.LIB	large model library (registered only)
WINSPECD.DLL	dynamic link library (registered only)

#### Library Examples

WSPTEXAM.C	winspector use example
WSPTEXAM.DEF	example definition file
WSPTEXAM.MAK	example make file

#### DLL Examples (registered only)

WSPTDLL.C	winspector DLL use example
WSPTDLL.DEF	example definition file
WSPTDLL.MAK	example make file

#### Additional files

ORDFORM.TXT	order form
-------------	------------

Winspector is a Microsoft Windows 3.x debugging program which serves as an alternative to using the Windows `MessageBox()` function in many situations.

To use Winspector you link the Winspector library with your program, and then make calls to one of the Winspector API functions:

- `StartWinspector()`
- `CloseWinspector()`
- `TellWinspector()`
- `TellWinspectorFmt()`
- `WinspectorAutoSave()`

These functions find the Winspector program and send your debugging data to it. Winspector displays the data sent to it by Windows programs in a listbox which allows you to scroll through the displayed messages. The listbox is sensitive to the size of the Winspector window so it can be made any size you like.

## Winspector Limitations

---

- The main listbox is limited by Windows to 64Kbytes, this limitations is imposed by Windows regardless of the data type in listboxes, the limit is one segment (64 Kbytes).
- In the DLL version of Winspector's API the function [TellWinspectorFmt\(\)](#) is not available due to the characteristics of the [pascal calling conventions](#) in DLL functions.

Please fell free to let me know any bug or improvement that you could find useful.

### Known Problems:

The shareware version of Winspector may loose some messages during the initial message box asking for loading the help. Besides this, I don't know any bug.

Of course, after registering Winspector that message box will disappear.

## Using Winspector

---

Four steps are needed to use Winspector.

1. Include the header file WINSPECT.H, which contain the functions prototypes for the Winspector API functions you want to use.
2. Include in your project one of the Winspector's libraries, conforming the model you are using:

WINSPECS.LIB	small model library
WINSPECM.LIB	medium model library
WINSPECL.LIB	large model library
3. Include calls to the Winspector's API functions in your application code.

Example:

```
:
case WM_SIZE:
    TellWinspector( "Change in window size");
    break;
:
or
:
case WM_SIZE:
    TellWinspectorFmt( "Change in window size (x,y): %d, %d",
                      LOWORD( IParam), HIWORD( IParam));
    break;
:
```

The TellWinspectorFmt() API function takes arguments which are identical to those of the Windows wsprintf() function. That is a format string and arguments.

A return code of TRUE indicates that API function succeeded in posting the data to the Winspector program. A return code of FALSE indicates failure.

5. Start the Winspector program. Now when your program runs the Winspector API functions will pass your debugging information to Winspector, which will display them in its listbox.

Alternatively, you can call the Winspector API function StartWinspector() to start the Winspector program. To use this function include a function prototype: `BOOL StartWinspector(void)`. Then before you call one of the Winspector API functions, you can call StartWinspector() to make sure Winspector is running. The return value from StartWinspector indicates whether or not Winspector was started. Winspector is assumed to be in the current directory, or in a directory referenced by the PATH environment variable. StartWinspector will not start the

Winspector program if it is already started.

For more explanation in how to use Winspector's API see the examples programs.

## Using the Winspector API functions

---

StartWinspector( void)

CloseWinspector( void)

TellWinspector( LPSTR str)

TellWinspectorFmt( LPSTR str, ...)

WinspectorAutoSave( BOOL OnOff)



## **StartWinspector( void)**

---

Starts the Winspector program. Be sure that winspect.exe is in your PATH, or a message box will be displayed informing you the error.

Returns 1 if winspect.exe was found and is running, otherwise it returns 0.

## **CloseWinspector( void)**

---

Closes the Winspector window if it is in memory.

Returns 1 if Winspector's window is closed or 0 if Winspector was not in memory, also it displays a message box informing the error.

## **TellWinspector( LPSTR str)**

---

This function sends the string pointed by *str* to the Winpector's listbox.

Returns 1 if the command is received by Winspector, otherwise it returns 0.

## **TellWinspectorFmt( LPSTR str, ...)**

---

This function works like printf(), the only difference is that it sends the formatted string to the Winspector's listbox.

This function is not available in the DLL version of Winspector's API.

Returns 1 if the command is received by Winspector, otherwise it returns 0.

## **WinspectorAutoSave( BOOL OnOff)**

---

This function sends to Winspector a command that tell Winspector to switch the autosave feature *on* or *off*.

This is the same as command File | sasve (F2) in Winpector window. This function takes one parameter that tells Winspector whether or not to save the messages as they are received.

Returns 1 if the command is received by Winspector, otherwise it returns 0.

## Commands

---

- Clear** Removes all messages in the listbox.
- Delete** Deletes the selected message in the listbox.
- Save** Writes received strings immediately to the save file *winspect.log*. Good for programs that hang Windows. If the file already exists Winspector displays a message box asking if you want to append the strings to the end of the file.
- Search** Brings up a dialog which allows you to search for the initial characters of an message received in the Winspector listbox.
- Help** Brings up the help file.
- About** Brings up a dialog box with author and version information on it.

