

# Contents

## Getting Started

- [Notes for Win32 versions](#)
- [Running TracePlus for the First Time](#)
- [Setting Trace Preferences](#)
- [Setting Data View Preferences](#)
- [Setting Filters](#)
- [Specifying the API Location](#)
- [Specifying the API Help File Location](#)
- [Prepending File Name and Line Number in Log File](#)

## Window Descriptions

- [The API View](#)
- [The Data View](#)
- [The Socket View](#)

## Running Traces

- [Starting a Trace](#)
- [Appending a Comment](#)
- [Pausing a Trace](#)
- [Viewing Trace Event Details](#)
- [Continuing a Trace](#)
- [Clearing the Current View](#)
- [Clearing all Views](#)
- [Exporting Trace Log Files](#)
- [Importing Trace Log Files](#)

## General Procedures

Note that these procedures are listed in alphabetical order.

- [Exiting TracePlus](#)
- [Finding a Trace Record](#)
- [Getting API Help](#)
- [Printing the contents of a View](#)
- [Setting up Your Printer](#)
- [Saving the Configuration](#)
- [Saving Trace Information Without a Previously Specified Destination](#)
- [Setting Trigger Conditions](#)
- [Using SSTPrintf\(\) and SSTPrintfEx\(\)](#)
- [Viewing a Remote PC Configuration](#)
- [Viewing API Performance](#)
- [Viewing Winsock Information](#)

## References

- [Keyboard Shortcuts](#)
- [Toolbar](#)
- [Index](#)

## Notes on Win32 versions

32 bit versions of TracePlus use the standard Microsoft defined debugging interface for Win32 applications. The Microsoft interface does not permit detaching from a process after debugging has started. Therefore, if TracePlus is terminated, it will also terminate all processes being traced. Stopping the trace will not detach TracePlus from a process.

Be sure that if you are tracing a server application, leave the TracePlus32 application running after stopping the trace, or terminate the server application, and restart it (be sure that TracePlus32 is not running or the trace is stopped before starting the server application again).

- TracePlus32 can trace a Win32 console application. To do this, launch the console application from TracePlus32. See the topic [Setting Filters](#) for more details on this procedure.
- TracePlus32 does not trace Win32 applications started before TracePlus32, unless they are explicitly selected from the *Tasks* listbox. See the topic [Setting Filters](#) for more details on this procedure.
- TracePlus32 does not trace 16 bit Windows applications.

# Running TracePlus for the First Time

The first time you run TracePlus/Winsock, follow the procedures below:

- 1 In the Help menu, select View README.WRI.... This command provides information that has become available since the manual was printed, and other vital information.
- 2 Select Options... (Trace menu) to set your trace preferences. Alternatively, press **Ctrl + O**, or select the Trace icon in the tool bar.
- 3 Select Filters... (Trace menu) to set trace functions and tasks. Alternatively, press **Ctrl + F**, or select the Filters icon in the tool bar.

To access help from the TracePlus window, press **F1**.

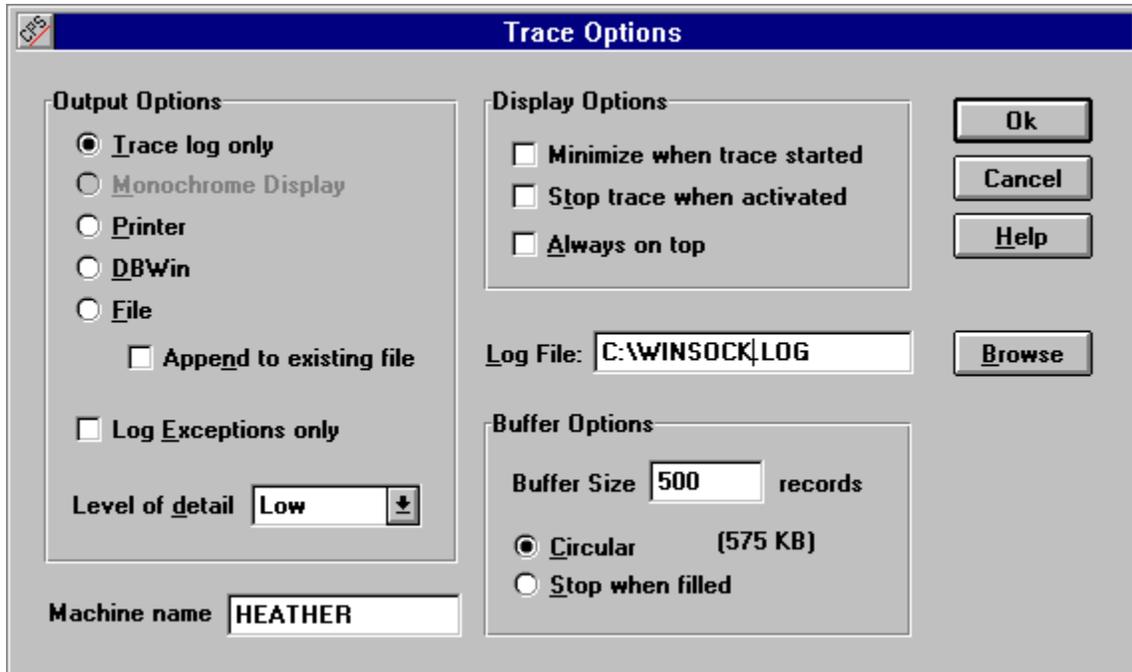
# Setting Trace Preferences

To set your trace preferences, select Trace Options... in the Trace menu, or press **Ctrl + O**.



*Shortcut:* Select the Trace icon.

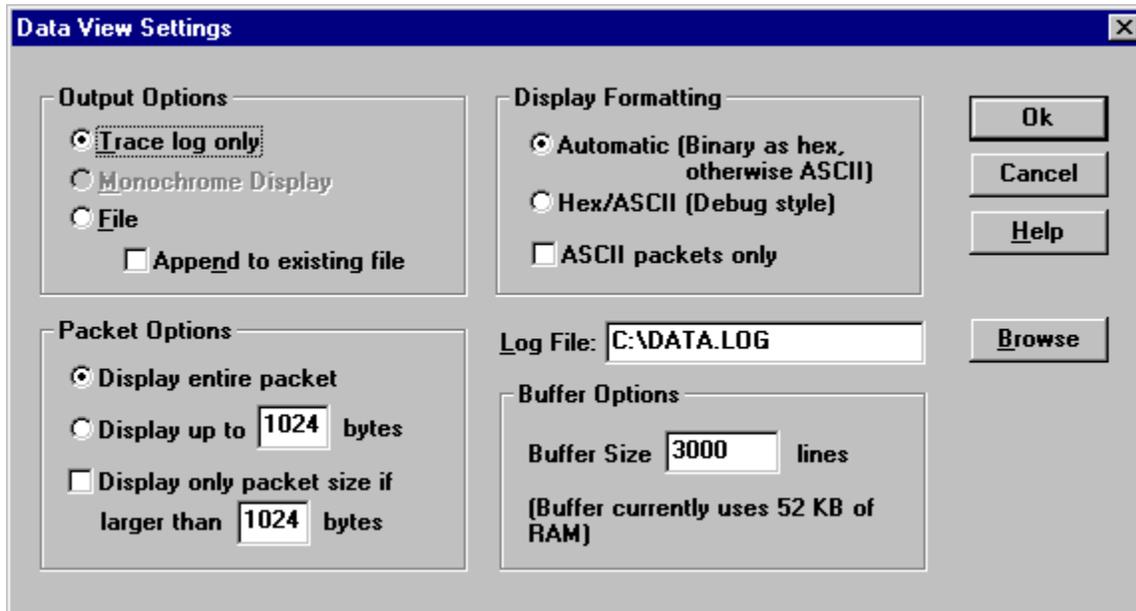
The Trace Options dialog box is displayed. Select any field or button for more information on using this dialog box.

The Trace Options dialog box is shown with a blue title bar. It contains several sections: 'Output Options' with radio buttons for 'Trace log only' (selected), 'Monochrome Display', 'Printer', 'DBWin', and 'File', and a checkbox for 'Append to existing file'; 'Display Options' with checkboxes for 'Minimize when trace started', 'Stop trace when activated', and 'Always on top'; 'Log File' with a text box containing 'C:\WINSOCK\LOG' and a 'Browse' button; 'Buffer Options' with a 'Buffer Size' of '500 records' and radio buttons for 'Circular (575 KB)' (selected) and 'Stop when filled'; and a 'Machine name' text box containing 'HEATHER'. On the right side, there are buttons for 'Ok', 'Cancel', 'Help', and 'Browse'.

Select the Ok button, or press **Enter** to set your preferences and exit the dialog box.

# Setting Data View Preferences

To set your data view preferences, select Data View Options... in the Trace menu, or press **Ctrl + D**.



Select the Ok button, or press **Enter** to set your preferences and exit the dialog box.

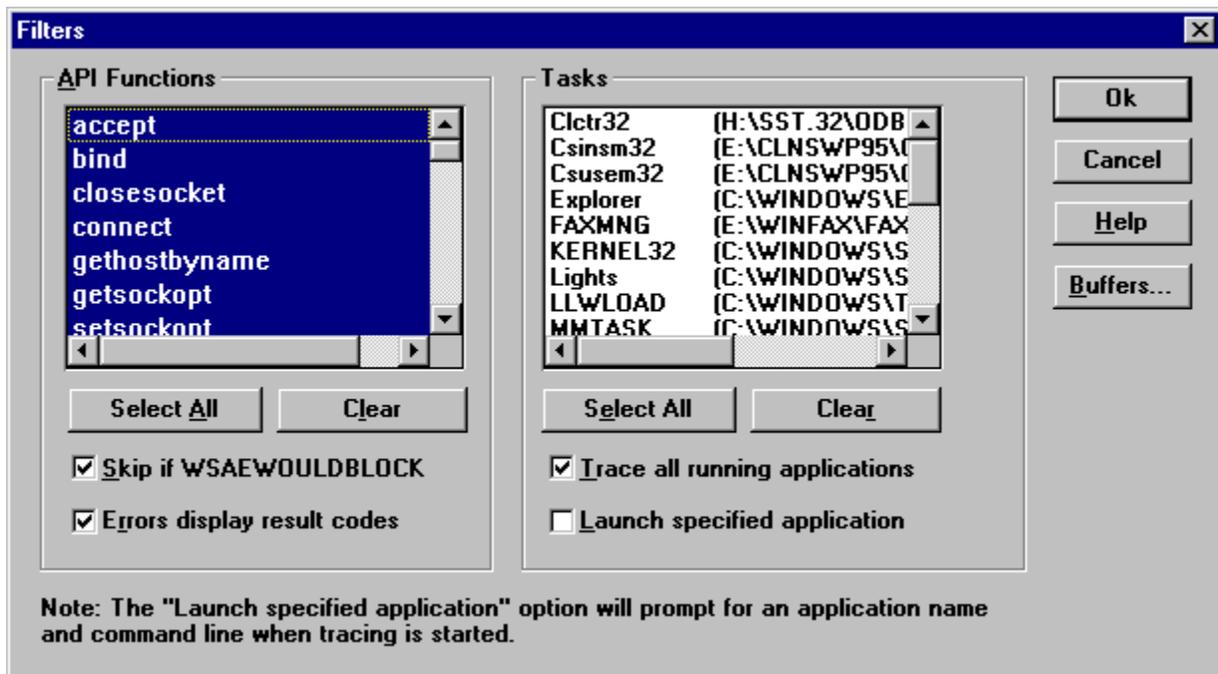
# Setting Filters

To set your trace preferences, select Filters... in the Trace menu, or press **Ctrl + F**.



*Shortcut:* Select the Filters icon.

The Filters dialog box is displayed. Select any field or button for more information on using this dialog box. Note that the dialog box may be slightly different, depending on your TracePlus application.



## Specifying the API Location

This function is provided so that you can use TracePlus/Winsock with multiple TCP/IP stacks. TracePlus loads the *WINSOCK.DLL* file you specify in the Trace Options dialog box.

## Specifying the API Help File Location

Select API Help File... (File menu) to specify the location of the API help file, if this file exists on your system. Specify the file as in a standard Open dialog box.

Once you specify the location of the API help file, you can access API help while the debugger is running. For more information on accessing API help, see [Getting API Help](#).

## Prepending File Name and Line Number

You can embed **SSTPrintf()** and **SSTPrintfEx()** in your program. You can then specify that these formatted text strings be included in the log file. **SSTPrintfEx()** formats the string with the filename and line number prepended to the string.

# The API View

The API View displays trace records created each time a Winsock API function is called by a Windows application.

The **Filters** dialog box contains settings which affect whether a trace record is generated. You can select specific applications and Winsock API functions. You can also filter only Winsock API functions that fail.

## Organization of the API View

The trace records appear as one or more lines in the scrolling listbox, and are numbered sequentially, starting at 1. If a Winsock API function fails, TracePlus will automatically call `WSAGetLastError()` and retrieve the underlying Winsock error code, unless specified otherwise (see the **Filters** dialog box).

The data for each packet is colored according to its status:



The function completed successfully.



The function failed.



A notification was received from `WSAAsyncxxx()`.

The trace records can be selectively displayed in one of three ways:

- **Tree** - Trace records appear as a single-line node of a tree. If you double-click or press Enter when the trace record is selected, the trace record will expand to show more information about the trace record.
- **Double Line** - The trace record appears as two lines. The first line displays the API function and its return code, and the second line displays the timestamp, elapsed time, and the application that called the function. If you double-click or press Enter when the trace record is selected, the **Event Details** dialog box is displayed..
- **Single line** - The trace record appears as a single line, displaying the API function and its return code. Like the double line trace record, double-clicking or pressing Enter will display the **Event Details** dialog box.

## Timestamps

The timestamp that appears on the second line of the trace record (and in the Event Details dialog box) has a resolution of 1 millisecond, and will match a packet displayed in the Data View, or a trace record in the Socket View (if one is generated).

# The Data View

The Data View displays contents of packets either sent or received by the following Winsock API functions:

- `recv()`
- `recvfrom()`
- `send()`
- `sendfrom()`

## Organization of the Data View

The packets appear as one or more lines in the scrolling listbox. The formatting of the packets can be configured by changing the settings in the **Data View Settings** dialog box.

The data for each packet is colored according to its source:



Sent from an application (The packet was passed to either the `send()` or `sendfrom()` API function).



Received by an application. (The packet was returned by the `recv()` or `recvfrom()` API function).

## Timestamps

The timestamp that appears at the far left of the packet data display has a resolution of 1 millisecond, and matches the time of Winsock API function where the packet originated. You can locate the specific API function by finding the trace record in the **API View** whose timestamp matches this value.

# The Socket View

The Socket View tracks the state changes of multiple sockets over their lifecycle. To detect state changes, the following Winsock API functions are monitored:

- `accept()`
- `bind()`
- `closesocket()`
- `connect()`
- `listen()`
- `socket()`

## Organization of the Socket View

A trace record is issued each time one of the above functions is called successfully. The Socket display is configured as a grid with 8 columns:

- Time - The time that the socket changed state.
- Task - The process that called the Winsock API function which changed the socket state.
- Socket ID - The socket number assigned by the Winsock subsystem.
- Type - Can be one of two types: TCP, or UDP.
- Service - The name of the service (returned by `getservbyport()`) that is communicating via the socket.
- Local IP:Port - The IP address and port number on the local side of the socket.
- Remote IP:Port - The IP address and port number of the remote process that is connected to the socket.
- State - The state of the socket when the trace record was issued.

## Timestamps

The timestamp in the first column of the grid has a resolution of 1 millisecond, and matches the time of Winsock API function where the packet originated. You can locate the specific API function by finding the trace record in the **API View** whose timestamp matches this value.

## Starting a Trace

To start a trace, select Start in the Trace menu, or press **Ctrl + S**.



*Shortcut:* Select the Start icon.

Before you start the trace, set your trace preferences and filters.

Note that you can drag single files from the Windows File Manager and drop them on the TracePlus window or the TracePlus icon. The specified *.EXE* file will be launched when the trace is started.

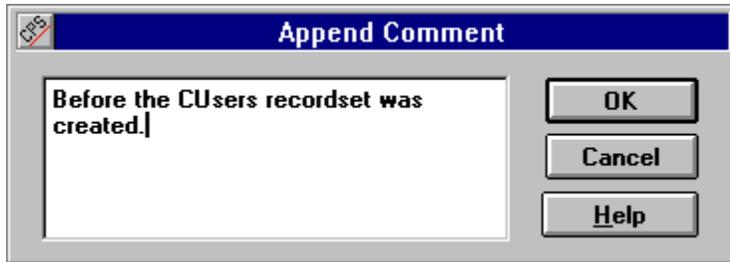
## Appending a Comment

Select Append Comment... (Trace menu) to add a comment to the end of a trace while the trace is running.



*Shortcut:* Select the Comment icon, or press **Ctrl + Shift + A**.

A dialog box similar to the following is displayed.



In the text box, type your comment and select the OK button. To exit the dialog box without appending the comment, select the Cancel button.

## Pausing a Trace

To pause a trace, select Pause in the Trace Menu, or press **Ctrl + A**.

Alternatively, If you selected the **Stop trace when activated** option in the Trace Options dialog box, tracing will stop when you activate the TracePlus program.



*Shortcut:* Select the Stop icon.

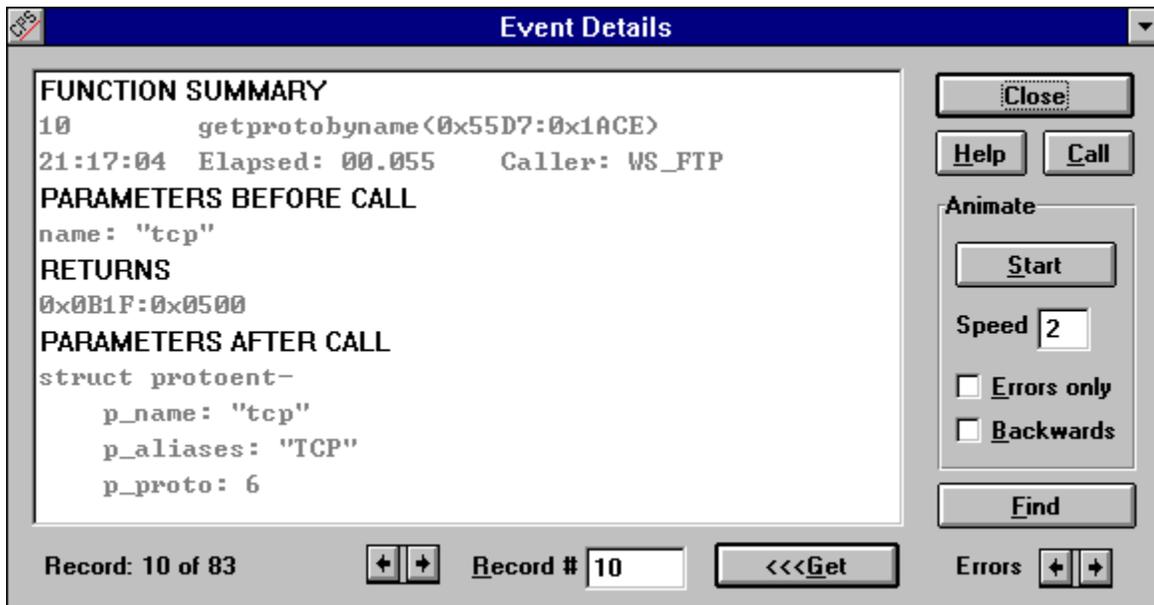
# Viewing Trace Event Details

Select Event Details (Trace menu) to view details of trace events. Alternatively, double click on a trace record in the TracePlus window, or press **Ctrl + E** .



*Shortcut:* Select the Details icon in the tool bar.

The Event Details dialog box similar to the one below is displayed. Click on the fields and buttons below for more information on how to use this dialog box.



The Event Details dialog box can remain on the screen as long as the trace buffer is not cleared either by selecting the Reset Events item in the Trace menu or selecting the Clear icon.

## Continuing a Trace

Select Continue in the Trace menu after you have stopped/paused the trace to save the current trace and continue tracing events.



*Shortcut:* Select the Continue icon.

## **Clearing the Current View**

To clear the current View, select Reset View in the Trace menu. The other view(s) will retain their current contents.

If the API View is cleared and Event Details dialog box is currently visible, it will be closed.

## Clearing all Views

To clear the trace log and buffers, select Reset Events in the Trace menu.



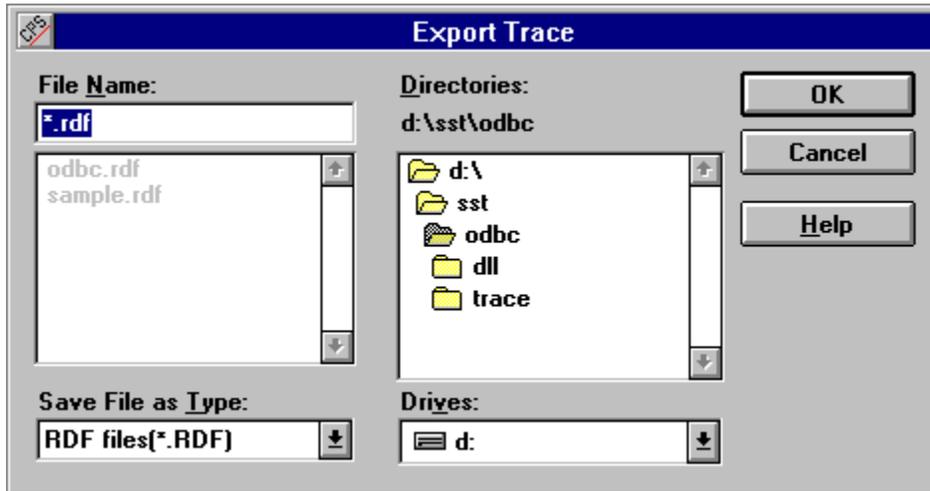
*Shortcut:* Select the Clear icon.

If the Event Details dialog box is currently visible, it will be closed.

## Exporting Trace Log Files

Select Export Trace... in (Trace menu) to export trace log files from a customer site to a file that can be imported into TracePlus running on your PC. This allows you to debug a customer's system.

The Export Trace dialog box is displayed.

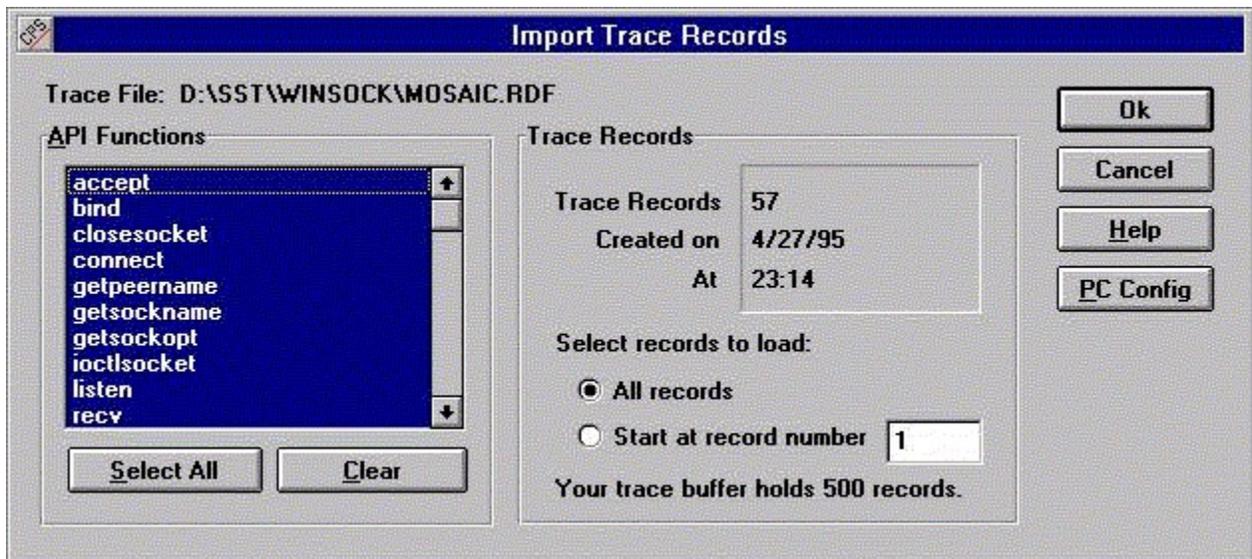


Specify the destination for the .RDF file as in a standard Save As dialog box.

## Importing Trace Log Files

Select Import Trace... (Trace menu) to import trace logs from another copy of TracePlus, such as one from a customer site that has been exported as a .RDF file. This feature allows you to load traces made on other PCs into TracePlus. Once the log is loaded, you can view and manipulate it as if the trace was created on your own PC..

When you load an exported .RDF file onto your system and select Import Trace..., a dialog box similar to the following is displayed. This dialog box displays API functions and records from the remote PC. Select buttons, options, and fields for more information on using this dialog box.



## Exiting TracePlus

Select Exit in the File menu or press **Alt + F4**.

## Finding a Trace Record

To find a record in the Event Details dialog box, select the Find button. To search for strings in the trace log from the main screen, Select Find... (Trace menu) or press **F3**.

The Find dialog box is displayed.



You can search by any numeric or alphabetic characters. If you search alphabetically, you can do a case-sensitive search (if the word for which you are searching contains capital letters) by selecting the Match Case option. You can also search up or down through records.

## Getting API Help

Before you access API you must first specify the location of the API help file by selecting API Help File... (File menu) and specifying the file in the API Help File dialog box.

Select *[API\_Name]*API... (Help menu) to access help on the API.



*Shortcut:* Select the API Help icon, or press **Ctrl + F1**.

Select Specific *API\_Name* Function (Help menu) to access help on a specific API function.



*Shortcut:* Select the API Function Help icon, or press **Shift + F1**.

Note that the public domain *WINSOCK.HLP* is included with TracePlus/Winsock and has hypertext links to the API functions that appear in the trace log. By pressing **Shift + F1** in the main window, or pressing the Call button in the Details dialog box, you can get detailed help on that Winsock function. Select Specific *API\_Name* Function (Help menu) to access help on a specific API.

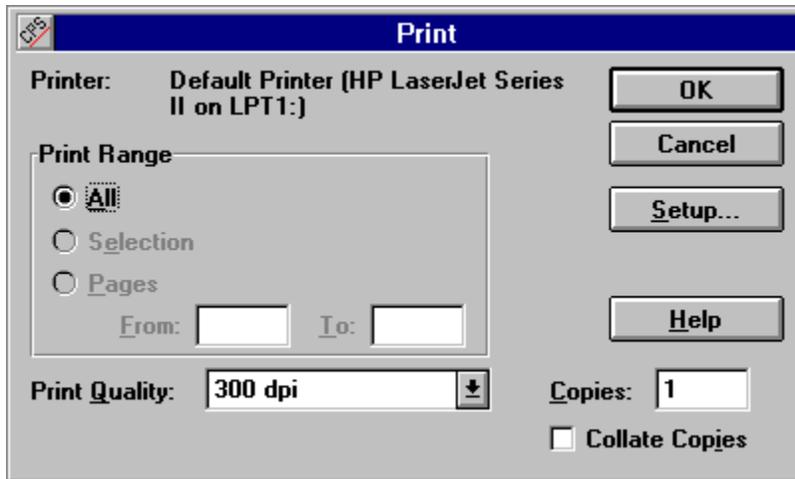
## Printing the contents of a View

Select Print in the File menu, or press **Ctrl + P**, or to print the contents of the current View.

Note: This feature is not available for the Socket View.



*Shortcut:* Select the Print icon.

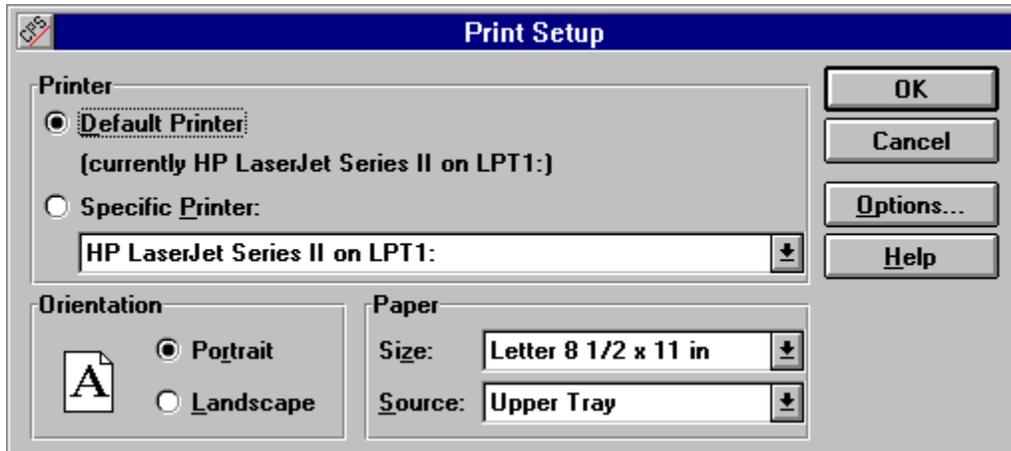


The default print range is all. You cannot choose the Selection or Pages options. You can select the print quality, the number of copies you want to print, and if you want the copies collated.

Press the **Setup** button to access the printer options available to you from the printer manufacturer.

# Setting up Your Printer

Select Printer Setup in the File Menu to access the Print Setup dialog box and set up your printer.



Select the Options... button in the dialog box to access the printer options available to you from the printer manufacturer.

## **Saving the Configuration**

Select Save Configuration (Configure menu) to save the window position, Trace Options dialog box settings, and the Filters dialog box settings (except for tasks, or applications).

## Saving Trace Information

You can save the current contents of the API View in ASCII format to a file by selecting Save Trace As... from the **File** menu.

Trace records are formatted according to the *Level of Detail* setting in the **Options** dialog box.

- The High setting formats the trace record as it appears in the **Event Details** dialog box.
- The Low setting formats the trace record as a two line summary. This format is the same as viewing trace records in the API View with the *Exploding* and *Single Line* options turned off.

The trace log is prepended by a comments section which describes the conditions of the trace, i.e.:

- The version of TracePlus/ODBC that generated the trace log.
- System information from the PC that generated the trace log.
- Trace options in effect at the time the trace was started.

### Saving Trace Information as HTML

You can save the current contents of the API View as a Web page by selecting Save Trace As HTML... from the **File** menu.

The trace record formatting options are the same as those specified for ASCII files.

The resulting .HTM file can be uploaded to your Web server or displayed in a Web browser such as the Netscape Navigator or the Internet Explorer.

# Setting Trigger Conditions

To start a trace using trigger conditions, select Trigger Conditions... in the Trace menu, or press **Ctrl + T**.

The Trigger Conditions dialog box is displayed. Select any field or button for more information on using this dialog box.

Trigger Conditions

Trigger conditions determine when a trace starts, according to the conditions below.

Logic **Any of the conditions is true**

Trigger on specific Winsock function  
getpeername

Trigger on return value  
WSAECONNREFUSED

Trigger on value in buffer

send()      Position **From the start**      Offset **0**

recv()      Bytes **ASCII**      **STAT**

Hex strings are formatted as "nn nn nn" where n is a hex digit (0-F). Separate each byte with a space.

OK  
Help  
Cancel

Select the Ok button, or press **Enter** to save your selections and exit the dialog box.

# Using SSTPrintf and SSTPrintfEx

TracePlus provides two functions, **SSTPrintf()** and **SSTPrintfEx()**, that allow you to insert your own messages into the trace log. Both functions have the same functionality as the C function **printf()**. Additionally, if you are running two or more TracePlus applications, you can specify which TracePlus application will log your message.

## Description

You need the following files to use **SSTPrintf()** and **SSTPrintfEx()**:

- *SSTDEBUG.H*: This header file contains the prototypes for the functions.
- *SSTDEBUG.LIB*: This library file includes the exports for *SSTDEBUG.DLL*.
- *SSTDEBUG.DLL*: This *.DLL* must be either in your path or in the same directory as the application you are writing.

## For Visual Basic Developers

Include the following declaration at the top of the module:

- Declare Function SSTPrint Lib "SSTDEBUG" ( ByVal DebuggerType as Integer, ByVal Message as String )

## For Powerbuilder Developers

Include the following declaration at the top of the module:

- Function INTEGER SSTPrint ( INTEGER DebuggerType, STRING MessageString ) LIBRARY "SSTDEBUG"

## SST Function Descriptions

The following functions are available in *SSTDEBUG.DLL*:

### SSTPrintf()

```
BOOL FAR __cdecl SSTPrintf( WORD DebuggerType, LPSTR szFormat, ... );
```

### Arguments

**int DebuggerType**

May be one of the following:

- **DEBUGGER\_TYPE\_WINSOCK**
- **DEBUGGER\_TYPE\_ODBC**
- **DEBUGGER\_TYPE\_SQL\_SERVER**
- **DEBUGGER\_TYPE\_WINDOWS**
- **DEBUGGER\_TYPE\_ALL**

Note by OR'ing the **DebuggerType** flags together, you can send the same comment to multiple TracePlus applications. This feature is commonly used to create checkpoints while tracing multiple APIs, such as ODBC and Winsock.

**char \*szFormat**

A null terminated string containing any valid **printf** style formatting string. See your C language

reference for information on valid format specifications.

### **argument**

One or more optional parameters that will be formatted according to the format specification.

### **Returns**

No return value.

### **Comments**

**SSTPrint()** will create an entry in the trace log:

- **1125:** This is a comment.
- **12:35:00:** Task: GENERIC

### **Sample Code**

```
#include "sstdebug.h"
```

```
int PASCAL WinMain( HANDLE hInstance,HANDLE hPrevInstance,LPSTR lpCmdLine,int  
nCmdShow )
```

```
{  
MSG msg;                /* message          */
```

```
SSTPrintf( "The value of hInstance is: 0x%X",hInstance );
```

```
while( TRUE ) {  
    if ( PeekMessage( &msg,NULL,0,0,PM_REMOVE ) )  
        if ( msg.message == WM_QUIT )  
            break;  
    else  
        if ( !TranslateAccelerator( hMainWnd,hAccel,&msg ) ) {  
            TranslateMessage( &msg );  
            DispatchMessage( &msg );  
        }
```

```
    BackgroundProcessing();  
}
```

```
return( msg.wParam );    /* Returns the value from PostQuitMessage */  
}
```

### **SSTPrintfEx()**

```
BOOL FAR __cdecl SSTPrintfEx( WORD DebuggerType,LPSTR szFilename,int LineNumber,LPSTR  
szFormat, ... );
```

### **Arguments**

**int DebuggerType**

May be one of the following:

- **DEBUGGER\_TYPE\_WINSOCK**
- **DEBUGGER\_TYPE\_ODBC**
- **DEBUGGER\_TYPE\_SQL\_SERVER**
- **DEBUGGER\_TYPE\_WINDOWS**
- **DEBUGGER\_TYPE\_ALL**

Note by OR'ing the **DebuggerType** flags together, you can send the same comment to multiple TracePlus applications. This feature is commonly used to create checkpoints while tracing multiple APIs, such as ODBC and Winsock.

#### **LPSTR szFilename**

The filename of the source code where this function is being called.

#### **int LineNumber**

The source code line number where this function is being called.

#### **char \*szFormat**

A null terminated string containing any valid **printf** style formatting string. See your C language reference for information on valid format specifications.

#### **argument**

One or more optional parameters that will be formatted according to the format specification.

#### **Returns**

No return value.

#### **Comments**

**SSTPrintfEx()** will create an entry in the trace log:

- **1125:** This is a comment.
- **12:35:00:** Task: GENERIC

#### **Sample Code**

```
#include "sstdebug.h"
```

```
int PASCAL WinMain( HANDLE hInstance,HANDLE hPrevInstance,LPSTR lpCmdLine,int
nCmdShow )
```

```
{
MSG msg;                /* message          */
```

```
SSTPrintfEx( __FILE__,__LINE__,"The value of hInstance is: 0x%X",hInstance );
```

```
while( TRUE ) {
    if ( PeekMessage( &msg,NULL,0,0,PM_REMOVE ) )
        if ( msg.message == WM_QUIT )
            break;
```

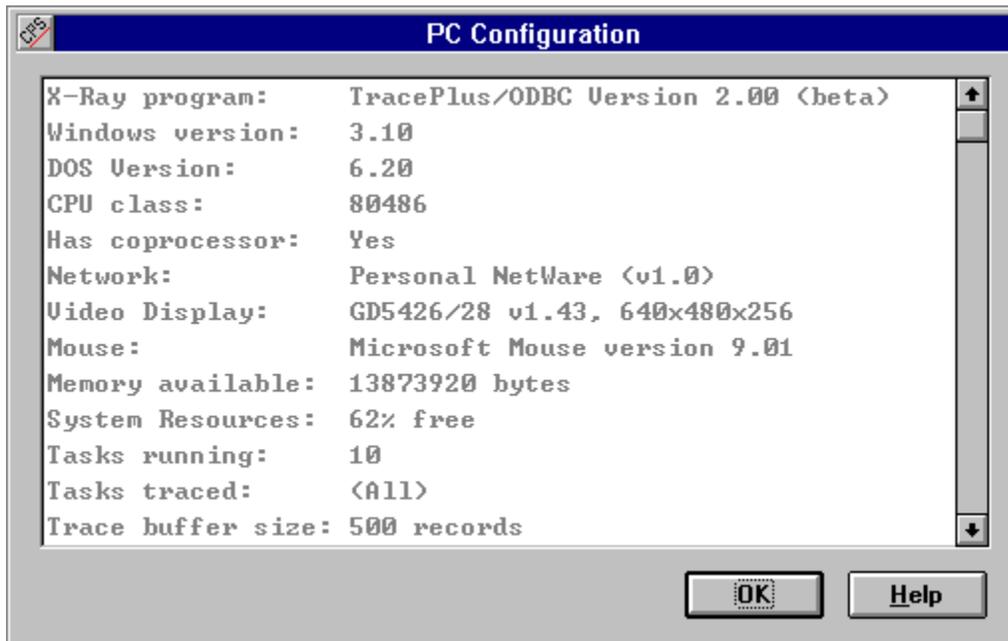
```
        else
            if ( !TranslateAccelerator( hMainWnd,hAccel,&msg ) ) {
                TranslateMessage( &msg );
                DispatchMessage( &msg );
            }

        BackgroundProcessing();
    }
return( msg.wParam );    /* Returns the value from PostQuitMessage */
}
```

# PC Configuration

To view the remote PC configuration from which a log file was exported/imported, select the PC Config button in the Import Trace Records dialog box. Alternatively, after you close the Import Trace Records dialog box, select PC Configuration... (View menu).

The PC Configuration dialog box is displayed.



# Viewing API Performance

The following TracePlus applications have the ability to record, display, and export performance information:

- TracePlus/ODBC
- TracePlus/SQL Server
- TracePlus/Winsock

Select API Performance... (View menu) to display the total calls, total time, average time per call, and the fastest and slowest recorded times for each call.

The following dialog box is displayed. Click on option fields and buttons for more information about how to use this dialog box.

The screenshot shows a dialog box titled "ODBC Performance" with a table of performance data. The table has columns for Function, # Calls, Total Time, Average Time, High Time, and Low Time. The data is sorted by the number of calls. Below the table is a "Sort By" section with radio buttons for Function, Calls, Total Time, Average, High, and Low. The "Calls" radio button is selected. On the right side of the dialog, there are buttons for "OK", "Save As...", and "Help".

Function	# Calls	Time (in seconds)			
		Total	Avg	High	Low
SQLBindCol	46	0.064	0.001	0.001	0.002
SQLColAttributes	46	0.061	0.001	0.001	0.002
SQLDescribeCol	46	0.177	0.003	0.002	0.034
SQLNumResultCols	46	0.052	0.001	0.001	0.002
SQLGetInfo	10	0.012	0.001	0.001	0.002
SQLSetStmtOption	4	0.007	0.001	0.001	0.002
SQLDriverConnect	1	9.238	9.238	9.238	9.238
SQLExecute	1	2.358	2.358	2.358	2.358
SQLExtendedFetch	1	0.204	0.204	0.204	0.204
SQLGetFunctions	1	0.001	0.001	0.001	0.001
SQLAllocEnv	1	0.096	0.096	0.096	0.096
SQLAllocConnect	1	0.002	0.002	0.002	0.002
SQLPrepare	1	0.963	0.963	0.963	0.963
SQLSetConnectOption	1	0.020	0.020	0.020	0.020

Sort By

Function  Calls  Total Time  Average  High  Low

# Viewing Winsock Information

The Winsock Information dialog displays the following information about the Winsock implementation:

- Highest compatible Winsock version.
- Copyright notice returned from WSASStartup().
- Maximum open sockets per process.
- Maximum size of a UDP packet.
- All allocated sockets. Any socket allocated by an application monitored by TracePlus will be included in the list. The list is updated in realtime as the socket(s) change state.

There is a caveat when using the *Open Sockets by Task* feature:

- Only the tasks selected in the Filters dialog box will be monitored for open sockets.

Note that sockets opened by an application that are not subsequently closed (i.e. the application UAEd, logic error, etc.) will remain in the open sockets list until the Delete button is pressed. At that time all sockets belonging to tasks that no longer exist will be removed from the list.

There are four states that will be displayed for an open socket:

- 1 ALLOCATED
- 2 OPEN
- 3 LISTEN
- 4 CONNECTED

# Keyboard

Following are the TracePlus keyboard shortcuts.

Access API help on ODBC	Ctrl + F1
Access API help on SQL Server	Ctrl + F1
Access API help on Windows	Ctrl + F1
Access API help on Winsock	Ctrl + F1
Access help on specific API function	Shift + F1
Access TracePlus help	F1
Append a comment to a trace	Ctrl + Shift + A
Clear trace log and buffer	Ctrl + R
Continue the trace	Ctrl + C
Display event details	Ctrl + E
Display low memory (Windows)	Ctrl + V
Display resource usage (Windows)	Ctrl + U
Exit TracePlus	Alt + F4
Find a record	F3
Pause the trace	Ctrl + A
Print	Ctrl + P
Set trace functions and tasks	Ctrl + F
Set trace preferences	Ctrl + O
Start SQL Capture (ODBC/SQL Server)	Ctrl + Q
Start the trace	Ctrl + S
Unload DLLs (Windows)	Ctrl + D

# Toolbar

The following command icons are available on the tool bar:



Saves the contents of the current trace log as an ASCII file.



Prints the contents of the screen.



Displays the details of a specified event.



Opens the Filters dialog box, so that you can select the functions and tasks (applications) you want to trace.



Opens the Trace Options dialog box, so that you can set trace preferences.



Allows you to append a comment to the current trace.



Clears the trace log and buffer.



Starts trace, and clears the current trace



Stops tracing events.



Continue tracing events. It saves the current trace.



Displays the Winsock API help file (if you have previously specified the location).



Displays help on a specific API function.

# Index

<a href="#">≡</a>
<a href="#">#</a>
<a href="#">A</a>
<a href="#">B</a>
<a href="#">C</a>
<a href="#">D</a>
<a href="#">E</a>
<a href="#">F</a>
<a href="#">G</a>
<a href="#">H</a>
<a href="#">I</a>
<a href="#">J</a>
<a href="#">K</a>
<a href="#">L</a>
<a href="#">M</a>
<a href="#">N</a>
<a href="#">O</a>
<a href="#">P</a>
<a href="#">Q</a>
<a href="#">R</a>
<a href="#">S</a>
<a href="#">T</a>
<a href="#">U</a>
<a href="#">V</a>
<a href="#">W</a>
<a href="#">X</a>
<a href="#">Y</a>
<a href="#">Z</a>

## **A**

[API Functions](#)

[API functions](#)

[Appending a Comment](#)

## **B**

[Backwards](#)

[Browse Records](#)

[Buffer Options Data](#)

[Buffer Options Field](#)

[Buffers](#)

[Buffers](#)

[Buffers](#)

## **C**

[Call](#)

[Cancel](#)

[Clear](#)

[Clearing all Views](#)

[Clearing the Current View](#)

[Close](#)

[Contents](#)

[Continuing a Trace](#)

## **D**

[DBWin](#)

[Detail List Box](#)

[Display buffers in ASCII only](#)

[Display buffers in Hex ASCII](#)

[Display Formatting](#)

[Display Options Field](#)

[drag and drop](#)

## **E**

[Errors display result codes](#)

[Errors Only](#)

[Errors](#)

[Exiting TracePlus](#)

[Exporting Trace Log Files](#)

## **F**

[File](#)

[Finding a Trace Record](#)

[Functions](#)

## **G**

[Get](#)

[Getting API Help](#)

## **I**

[Importing Trace Log Files](#)

[Include Types](#)

[Index](#)

## **K**

[Keyboard](#)

## **L**

[Launch Application](#)

[Launch specified application](#)

[Level of detail](#)

[Log Exceptions Only](#)

[Log File](#)

[Logic](#)

## **M**

[Machine name](#)

[Menus](#)

[Monochrome Display](#)

## **N**

[Notes on Win32 versions](#)

## **O**

[Ok](#)

## **P**

[Packet Options](#)

[Pausing a Trace](#)

[PC Configuration](#)

[Prepending File Name and Line Number](#)

[Printer](#)

[Printing the contents of a View](#)

## **R**

[Refresh](#)

[Running TracePlus for the First Time](#)

## **S**

[Save As](#)

[Saving the Configuration](#)

[Saving Trace Information](#)

[Select All](#)

[Select records to load](#)

[Send Receive Display Options](#)

[Setting Data View Preferences](#)

[Setting Filters](#)

[Setting Trace Preferences](#)

[Setting Trigger Conditions](#)

[Setting up Your Printer](#)

[Skip if WSAEWOULDBLOCK](#)

[Sort By](#)

[Sort](#)

[Source](#)

[Specifying the API Help File Location](#)

[Specifying the API Location](#)

[Speed](#)

[Start](#)

[Starting a Trace](#)

## **T**

[Tasks](#)

[Tasks](#)

[The API View](#)

[The Data View](#)

[The Socket View](#)

[Toolbar](#)

[Trace all running applications](#)

[Trace log only](#)

[Trace log](#)

[Trigger on return value](#)

[Trigger on specific Winsock function](#)

[Trigger on value in buffer](#)

**U**

[Using SSTPrintf and SSTPrintfEx](#)

**V**

[Viewing API Performance](#)

[Viewing Trace Event Details](#)

[Viewing Winsock Information](#)

## API Functions

Select the trace functions you want to view in the trace log.

For more information about these functions, refer to the documentation that accompanies the Winsock API. TracePlus does not trace the Winsock function **WSAGetLastError()**.

## **Backwards**

Select this option if you want to see the log in reverse order while animation is enabled.

## **Browse Records**

Select the Browse scroll buttons to view records ahead or behind the current record.

## **Buffer Options (Data View)**

The buffer size of the Data View uses the concept of displayable lines of text, rather than actual trace records (used in the API View).

The Data View will display up to the number of text lines specified. If the newest packet causes the number of lines to exceed the specified maximum, the oldest packets will be deleted from the display (FIFO).

## Buffer Options Field

The number in parentheses below the Buffer Size option is the amount of memory in kilobytes.

**Buffer Size:** Enter the number of records, between 25-2000 inclusive, that you want to buffer.

**Circular:** When the number of records is one more than the number you specified in the **Buffer Size** option, the buffer drops the oldest record, giving the appearance of an endless buffer. The larger the buffer size, the more records it can hold before losing the oldest record.

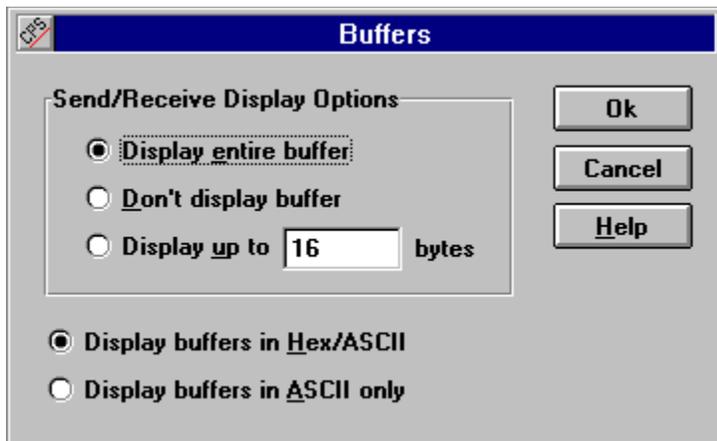
**Stop when filled:** The buffer stops accepting record when the number of records you specified in the **Buffer Size** option is reached. The trace is automatically stopped.

# Buffers

Select this button to specify the way that data packets are formatted in the **Event Details** dialog box. The following Winsock API functions display data packets:

- recv()
- recvfrom()
- send()
- sendto()

Select any field or button below for more information on using this dialog box.



TracePlus will consume much more memory if you are tracing applications that send and receive data in large packet sizes (2048, 4096, etc.). To minimize memory usage, set a limit for buffer sizes, or turn off the buffer display altogether. Alternatively, you can set the trace buffer size to a smaller value, typically one half of the current buffer size.

## Call

Select this button to display a description of the function shown in the details list box as outlined in the API help file. This button is grayed if the API help file is not accessible by TracePlus.

## **Cancel**

Select this button to exit the dialog box without making any changes.

## **Clear**

Select this button to cancel your choice(s) in the list box.

## **Close**

Select this button to close the dialog box.

## DBWin

Outputs trace information to DBWin (a sample debugger available with Microsoft Windows SDK and Microsoft C/C++ (Visual C++™)). This option also supports output to WinScope™ by Periscope. TracePlus is compatible with any program capable of displaying messages sent by the function **OutputDebugString( )**.

Be aware that this option can slow your system down if many trace events are being logged.

## Detail List Box

This window displays detail of a specified trace. You can specify the trace for which you want more detail by double clicking on a trace record in the TracePlus window, or by entering a number in the Record text box and pressing the Get button. Alternatively, you can scroll through the list box, or select the Animate button to view other records, or select the Find button to find a record alphanumerically.

The following information is displayed for every trace detail:

- **FUNCTION SUMMARY:** A summary of the function, including record number, function, and the time it was called.
- **PARAMETERS BEFORE CALL:** Parameters in the code before the function is executed.
- **RETURNS:** Error codes or values that are returned by the function.
- **PARAMETERS AFTER CALL:** Values that may have been changed by the function.

### TracePlus/VBX+Windows additions

- **VERSION INFORMATION:** This section is displayed when a LoadLibrary() trace record is generated. It contains the information received by calling the Microsoft Version API, and includes version information, copyright and trademark notices, target OS, language type,
- **VALID PROPERTY SETTINGS:** If the VBX control property is an enumerated property, TracePlus will display the the list of valid settings for that property. Currently, this does not apply to standard properties.
- **DATA ACCESS STRUCTURE:** This is the Microsoft defined "C" structure that sent between the bound control and the data control, specifying the action to take by the receiving control.
- **VISUAL BASIC DECLARATION:** This is the predefined syntax for a VBX event procedure, as specified by the VBX control.
- **NEW PROPERTY VALUE:** The value that will be assigned to a property (displayed when a VBSetControlProperty() trace record is generated).
- **VB MESSAGE DETAILS:** The values of each parameter passed to the VBX control procedure, corresponding to the standard Windows message format, i.e. (HWND hWnd,unsigned msg,LPARAM lParam,LPARAM lParam).
- **VBX EVENT SUMMARY:** The parameter values passed to the VBFireEvent() API function.

## **Display buffers in ASCII only**

This option displays buffer in text format.

## **Display buffers in Hex ASCII**

This option displays buffer information in DEBUG style format.

# Display Formatting

Data packets can be formatted in one of three ways:

- 1 Automatic - When a packet is ready to be displayed, TracePlus will check the packet to determine the type of data in the packet. If all the data in the packet is ASCII (printable), the packet will be displayed as text. Otherwise, the packet will appear in hex.
- 2 Hex/ASCII - This is the standard hexadecimal packet display provided by TracePlus. Packet data is displayed in rows of 16 hexadecimal digits, with the ASCII representation of each hex digit to the right of each row.
- 3 ASCII packets only - If TracePlus determines that any of the data contained in a packet is not ASCII (printable), the packet will not be displayed.

## Display Options Field

**Minimize when trace started:** Minimizes the TracePlus application window to an icon.

**Stop trace when activated:** Stops the trace when you activate the TracePlus window.

**Always on top:** Keeps the TracePlus window on top of all other applications, so that it is always visible.

# Errors

Select the Errors scroll buttons to scroll through errors.

## **Errors display result codes**

If you are using TracePlus/Winsock, this checkbox selects whether TracePlus calls `WSAGetLastError()` to obtain the error code when a Winsock API function fails. If this option is not selected, TracePlus will display the word `ERROR` when a API function fails.

## **Errors Only**

Select this option if you want to view error messages while animation is enabled.

## File

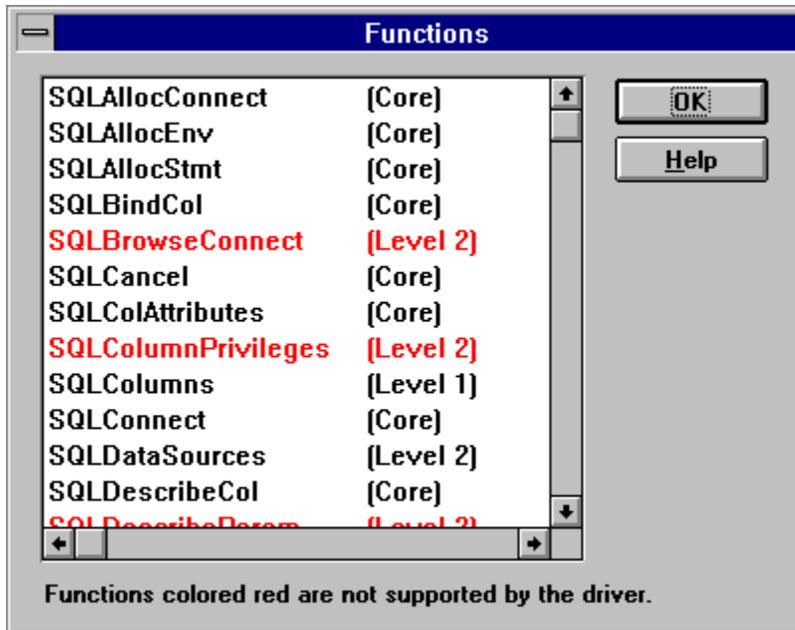
Outputs trace information to a log file. However, you can enter another file in the Log File text box, or choose another log file by selecting the Browse button.

The default is **Append to existing file**, which appends new trace information to the log file you specify. If you deselect the default, the log file is deleted and a new file is created when you run a trace on another function.

Note that if you do not specify the log file, you can save the contents of a trace buffer in ASCII format to a file by selecting Save Trace As... (File menu).

# Functions

Select this button to view the ODBC functions supported by the ODBC driver for the selected data source.



Supported functions are in black. Unsupported functions are in red.

Select Ok or press **Enter** to exit the dialog box and return to the Data Source Information dialog box.

## **Get**

Select this button to retrieve details for a trace you specify in the Record text box. The record is the first field that is displayed in the TracePlus window. You can also use the scroll buttons to scroll backwards and forwards through record numbers.

## Include Types

Select **System**, **Views**, or **Aliases** for the type of column you want to view.

Select Ok or press **Enter** to exit the dialog box and return to the ODBC Driver Information dialog box.

## Launch Application

This option enables you to launch applications from TracePlus and start tracing at the same time. The dialog box only appears if you have selected **Launch Application** from the **Filters** dialog box.

Each time you launch an application from TracePlus, the application filename is remembered by TracePlus. This list of previously launched applications appears underneath the **Application** text box. The list is sorted according to their priority, i.e. the most recent applications will appear at the top of the list.

### Specifying an Application to Launch

Select the **Browse** button, or enter the application filename in the **Application** text box. You can enter arguments to the application in the **Command Line** text box.

### Selecting a recently used Application

If the application you want to launch is in the list, double-click on it with the mouse. The application filename will be copied to the **Application** text box. Press Ok to launch the application.

## Launch specified application

Select this option if you want to run a trace on an application that is not displayed in the Tasks list box. To select the application, select the Choose application... button, or enter the application in the **Application** text box. You can enter arguments to the application in the **Command Line** text box.

## Level of detail

Select **High** to output information details to printer, DBWin, or the file you specified in the option above. Select **Low** to display only two lines of information to the Event Details dialog box.

Remember that this option does not change the format of the TracePlus main window. This options affects either the **File** or **Printer** output options.

## **Log Exceptions Only**

Select this option if you want to log functions that return errors only.

## Log File

If you selected **File** in the Output Options field, and you do not want to accept the default log file, enter the name of the file to which you want trace information stored, or choose another log file by selecting the Browse button.

Note that if you do not specify the log file, you can save the contents a trace buffer in ASCII format to a file by selecting Save Trace As... (File menu).

# Logic

The **Logic** option determines the number of conditions necessary to trigger **TracePlus** to start recording Winsock activity. There are three Logic settings:

- 1 No triggering - The log will start at the first Winsock call made by a selected application after the trace is manually started.
- 2 Any of the conditions is true - The log will start when any one of the selected trigger conditions occur.
- 3 All of the conditions are true - The log will start only when all of the selected trigger conditions have occurred (not necessarily at the same time).

## **Machine name**

The name of the workstation is automatically displayed in this field under Windows 95, Windows NT and Windows for Workgroups. This differentiates users when multiple workstations are traced by writing to the same log file on a network drive.

# Menus

Insert Menu help text here

## **Monochrome Display**

For a monochrome monitor. This option is grayed if you do not have a monochrome video card installed in your computer.

## Ok

Select the Ok button or press **Enter** to set your changes and exit the dialog box.

# Packet Options

You can limit the amount of packet data displayed in the Data View by specifying the maximum amount of bytes that will be displayed in a packet.

Additionally, you can choose to exclude packets entirely, if the packet size exceeds a certain threshold.

There are two settings that determine the maximum display size of a packet:

- 1 Display entire packet - Show each byte of a packet, regardless of packet size.
- 2 Display up to - If the packet size exceeds the specified amount, show only the specified number of bytes.

The option **Display only packet size** causes any packet exceeding the specified size to be represented in the Data View as a one line description of the packet size, i.e. "*Packet size is xxxx bytes*". This is particularly useful in filtering large binary file transfer packets common to HTTP and FTP protocols.

# Printer

Outputs trace information to your printer.

## **Refresh**

Refresh the list box.

## **Save As**

Select this button to save the performance statistics as a text file.

## Select All

To select individual list box items, press **Ctrl** as you click with the left mouse button. To select a range of list box items, click on the first item with the left mouse button, and then press **Shift** as you click on the last item with the left mouse button. To select all items in the list box, select the Select All button.

## Select records to load

Select All to load all records, or specify the record number at which you want to start. Note that the total number of records is specified for your convenience.

## Send/Receive Display Options

**Display entire buffer:** Displays the entire buffer in the selected format.

**Don't display buffer:** Displays "N/A" instead of the contents of the buffer.

**Display up to nnnn bytes:** Displays the first nnnn bytes of the buffer. Buffers are limited to 4096 bytes. If a buffer is larger than 4096 bytes, it will be truncated to 4096.

## Skip if WSAEWOULDBLOCK

If you select the **Skip if WSAEWOULDBLOCK** option, Winsock API functions returning the Winsock error code WSAEWOULDBLOCK are not logged. This error code is returned if the **recv()**, **recvfrom()**, **send()**, and/or **sendto()** functions fail because no data could be received or sent.

# Sort

Select the option by which you want to sort information

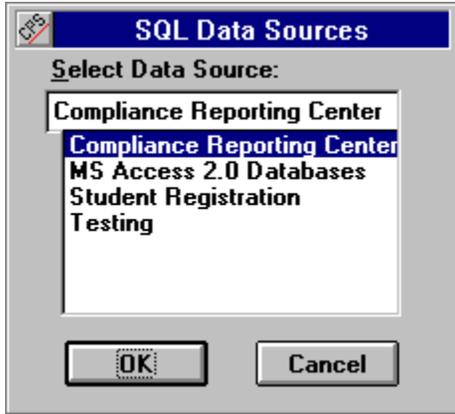
- If you choose address, addresses are listed in ascending order.
- Module refers to the module that allocates the memory.
- Size refers to the amount of memory the block occupies.

## Sort By

Select the sort option by which you want to display performance statistics.

## Source

When the ODBC Driver Information dialog box is displayed, the SQL Data Sources dialog box is also automatically displayed to allow you to select the data source. You can also select the Source button to display the SQL Data Sources dialog box.



Select the data source and select Ok or press **Enter**.

## Speed

Enter the number from 1 to 15. The higher the number, the slower the animation (that is, automated scrolling through the detail list box). The range of values equates to .2 through 3 seconds. You can initiate animation by selecting the Start key.

## Start

Select this button if you want to automate scrolling through the detail list box. To set the animation delay, enter the number of seconds delay in the seconds text box.

## Tasks

Select the applications for which you want trace information.

# Trace all running applications

Select this option if you want to trace all current and new applications.

- Note that this option is only available in 16 bit versions of TracePlus.

32 bit versions of TracePlus have an option called *Trace Processes started from Desktop*. This feature will automatically trace an application under two conditions:

- The application has a main window (i.e., not a Win32 console application).
- The trace will be initiated at the point where the main window receives the WM\_CREATE message. This means that any API calls made before this message is received will not be traced.

The procedure for tracing 32 bit applications launched from the desktop is as follows:

- 1 Make sure that a checkmark is next to the item *Trace processes started from desktop* in the Filters dialog box.
- 2 Unselect the option *Launch specified application*
- 3 Press OK (not Cancel) to close the **Filters** dialog box.
- 4 Click on the *Start* button on the toolbar, or press Ctrl+S.
- 5 Launch the Win32 application that you want to trace. Make sure that it creates a main window.

## **Trace log only**

Displays trace information to a window.

## Trigger on return value

This condition exists when a Winsock function fails and calling `WSAGetLastError()` returns the specified Winsock error code.

Use this function in conjunction with the logic *All the conditions are true*, and the condition *Trigger on a specific Winsock function*, to start the log when a specific Winsock function is called and fails with a specific error code.

## Trigger on specific Winsock function

This condition is considered to have occurred if the specified Winsock function has been called by any application selected in the **Filters** dialog box.

## Trigger on value in buffer

This is the most powerful of the trigger conditions, giving you the ability to start a log by looking for a specific string of ASCII text or byte pattern found in a data packet. The following Winsock functions deal with data packets:

- `recv()`
- `recvfrom()`
- `send()`
- `sendto()`

Search criteria can be specified in ASCII or hexadecimal. ASCII strings are case sensitive, i.e. uppercase and lowercase characters must match.

Search criteria can be specified in ASCII or hexadecimal. Hexadecimal strings consist of one or more bytes defined as two hex digits (0-9, A-F). Multiple bytes are separated by spaces, i.e. "0D 0A" is equivalent to a carriage return/line feed combination. The digits A-F are not case sensitive.



