TradeSim®

Advanced Trading Simulator and Back Tester

# Technical Brief 1

## Problems and Issues using the TradeSim/Metastock Plug-In

- ✓ TradeSim Standard, Professional and Enterprise Edition
- ✓ TradeSim Version 5.3.0 Beta Pre-Release
- ✓ TradeSim(DLL) Metastock Plugin Beta Version 7.0.1
- ✓ Metastock Version 7 and above

Last Update 8 April 2007

# Contents

# Introduction

Some issues may arise when using the ***TradeSim*** Plug-In. We have put together a list to help deal with the most common issues.

## Error Checking and Reporting

The RecordTrades function has extensive error checking and reporting abilities to cope with situations where errors are detected. The errors can be categorized as either Fatal or Recoverable. Fatal errors will force the RecordTrades function to abort immediately from the current operation. Recoverable errors are usually caused by invalid security data and will force the RecordTrades function to skip the invalid data. All recoverable errors are reported to the Report Log for later inspection. In the case of fatal errors and because the RecordTrades function writes trade data out immediately there may be situations where some trade data may have been written out to the trade database file before the fatal error occurred. In this situation the cause of the error(s) should be addressed otherwise the Trade Database will be incomplete.

### The Report Log

The new Report Log provides detailed user feedback. Please refer to the TradeSim User Manual for more information.

### Error Summary Table

A list of all of the internal error messages that can be generated by the RecordTrades function is given in the following table.

| Error | Type | Description | Possible Causes and Corrections |
|---|---|---|---|
| "Incorrect argument count" | Fatal | This error is usually caused by an incorrect number of arguments supplied to the RecordTrades function. | Specify the correct number of arguments. |
| "Invalid array index" | Fatal | This error is usually caused by a bad data array parameter that is supplied to the RecordTrades function. | This problem is typically caused when a data array, which contains too few data elements, is passed as a parameter to a function, which requires more data than what is available in the data array parameter. For example, using a 60-day moving average in an exploration with the data loading length set to 30 days in an exploration will flag this error. |
| "Invalid Entry Price Data Detected on dd/mmm/yyyy" | Recoverable | If the EntryPrice, Low of the day, High of the day, data arrays contain values, which are equal to or less than zero at the point of Entry into a trade, this error will be generated. This error will also be generated if the Entry Price is outside the low and high of the day at the point of entry. The date specifies the bar on the chart where the error occurred. | Typically this error occurs when there is corrupt data in the security price data or the EntryPrice array contains invalid values. |
| "Invalid Exit Price Data Detected on dd/mmm/yyyy" | Recoverable | If the ExitPrice, Low of the day, High of the day, data arrays contain values, which are equal to or less than zero at the point of Exit from a trade, this error will be generated. This error will also be generated if the Exit Price is outside the low and high of the day at the point of exit. The date specifies the bar on the chart where the error occurred. | Typically this error occurs when there is corrupt data in the security price data or the ExitPrice array contains invalid values. |

| | | | |
|---|---|---|---|
| "Invalid Initial Stop Data Detected on dd/mmm/yyyy" | Recoverable | If the InitialStop data is greater or equal to the EntryPrice at the point of entry into a long trade this error will be generated. If the InitialStop data is less than or equal to the EntryPrice at the point of entry into a short trade an error will be generated. The date specifies the bar on the chart where the error occurred. | If your InitialStop data references data in a previous bar then be careful that the resulting value does not violate the EntryPrice value at the point of entry into a trade. For example InitialStop:=ref(L,-1) may cause problems because the low of the previous day may exceed the current EntryPrice at the point of entry. |
| "Negative Initial Stop Data Detected on dd/mmm/yyyy" | Recoverable | If the InitialStop data is less than zero at the point of Entry into a trade then an error will be generated. | This error will occur if your InitialStop expression evaluates to a negative value at the point of entry into a trade. |
| "Error opening trade database file" | Fatal | This error is generated when an internal disk IO error occurs trying to open or create a trade database file. | The most likely cause is invalid filename specification. (See next section). |
| "Error writing trade database file" | Fatal | This is an internally generated error and occurs when there is an internal disk IO error writing data to a trade database file. | |
| "Error closing trade database file" | Fatal | This error is generated when an internal disk IO error occurs trying to close a trade database file. | |

## Trade Database File Naming issues

The following characters are not permitted in the specification of the trade database filename unless otherwise specified.

```
\       Back slash
/       Forward slash
:       Colon
*       Asterix
?       Question mark
<       Less than sign
>       Greater than sign
|       Pipe symbol
"       Quotes can only be used to surround the filename
```

**Note:** Do not specify a path or filename extension in the trade database filename. All trade database files are appended with a ***trb*** extension and are stored in the c:\tradesimdata directory.

The following trade database file names are examples of **valid** trade database filenames.

```
"My Trades"
"MACD Long"
"ATR Volatility(S&P-ASX100)"
"Turtle Breakout System"
"Equis Bollinger Bands & Money Management Stop"
"Random Entry & ATR Volatility(S&P-ASX100)"
"Rand Entry + Bollinger Bands Exit"
```

```
"MACD(CLOSE,13,130)"
```

The following trade database file names are examples of **invalid** trade database filenames and will cause an error to be generated.

```
"Rand Entry\Bollinger Bands Exit"       { cannot use backslash '\' }
"Rand Entry.Bollinger Bands Exit"       { cannot use the period '.'}
"MACD Crossover + (3*ATR Stop)"         { cannot use the asterix '*'}
```

## RecordTrades function limitations

Due to the limitations of the Metastock external DLL interface there can be a problem when you try and rerun the same exploration with different portfolios particularly when the first security is different in each of the portfolios. The external Plug-In contains a number of variables which keep track of the database name along with the first security symbol. If you rerun a record trades exploration with a different portfolio, which has a different first security symbol name, the trade database file may not be reset or rewritten correctly. The symptoms of this problem are that the database may contain duplicate trades or exclude some trades. There are three ways around this problem:

1. Firstly, use different explorations with different trade database filenames when creating trade databases using the same trading system but with different portfolios ie “My Trades ASX100”, “My Trades ASX200” etc. The idea being that you always use the same portfolio list with the same trade database file and don't use the same trade database file name with different portfolio lists.

2. Run a different trade database exploration with a different trade database filename before you rerun the same trade database exploration.

3. Alternatively, delete the trade database (.trb) file before regenerating it. Trade database files are stored in the c:\tradesimdata directory. This will force the RecordTrades function to reset any internal variables and regenerate the trade database file correctly.

# Debugging the RecordTrades function

It is useful to be able to analyse and study the actual trade signals produced by the RecordTrades functions. These will be described in the following chapters.

## Debugging the RecordTrades function using the ShowTrades function

Starting with TradeSim version 5 and above new features have been added to the Metastock/TradeSim interface which conveniently allow the trade data and trade signals to be viewed in both tabular and graphical forms. To do this all that is required is to add an extra line of Metastock code after the call to the Initialize function. The ShowTrades function triggers the plugin to send all of the relevant trading data to the TradeSim program and display it accordingly. The syntax for the ShowTrades function is shown below.

### ShowTrades (formerly ShowChart)

```
ExtFml("TradeSim.ShowTrades", Position (LONG,SHORT),SYMBOL);
```

When called with the appropriate symbol brings up a window, which displays a variety of signals in both graphical, tabular and tree form. This information is a useful aid for debugging trade database exploration code.

> Note that the symbol of interest must be included with the security list otherwise the chart will not appear. This window is part of the external TradeSim plugin, and is not part of the main TradeSim application.

**Example**

The following example displays all of the signals for the long side on a chart for symbol ANZ when a trade database exploration is run.

```
ExtFml("TradeSim.ShowChart",LONG,"ANZ");
```

The complete code should look like the following example.

```
EntryTrigger := REF(Cross(MACD(),Mov(MACD(),9,E)),-1);
EntryPrice := OPEN;
ExitTrigger := REF(Cross(Mov(MACD(),9,E),MACD()),-1);
ExitPrice := OPEN;
InitialStop := OPEN-ref(3*ATR(10),-1);

ExtFml( "TradeSim.Initialize");
ExtFml( "TradeSim.EnableProtectiveStop",0 );
ExtFml( "TradeSim.ShowTrades",LONG,"ANZ");     { <- add this line }
ExtFml( "TradeSim.RecordTrades",
        "MACD Crossover Test", { Trade Database Filename }
         LONG,                 { Trade Position Type }
         EntryTrigger,         { Entry Trigger }
         EntryPrice,           { Entry Price }
```

```
          InitialStop,          { Optional Initial Stop }
          ExitTrigger,          { Exit Trigger }
          ExitPrice,            { Exit Price }
          START);               { Recorder Control }
```

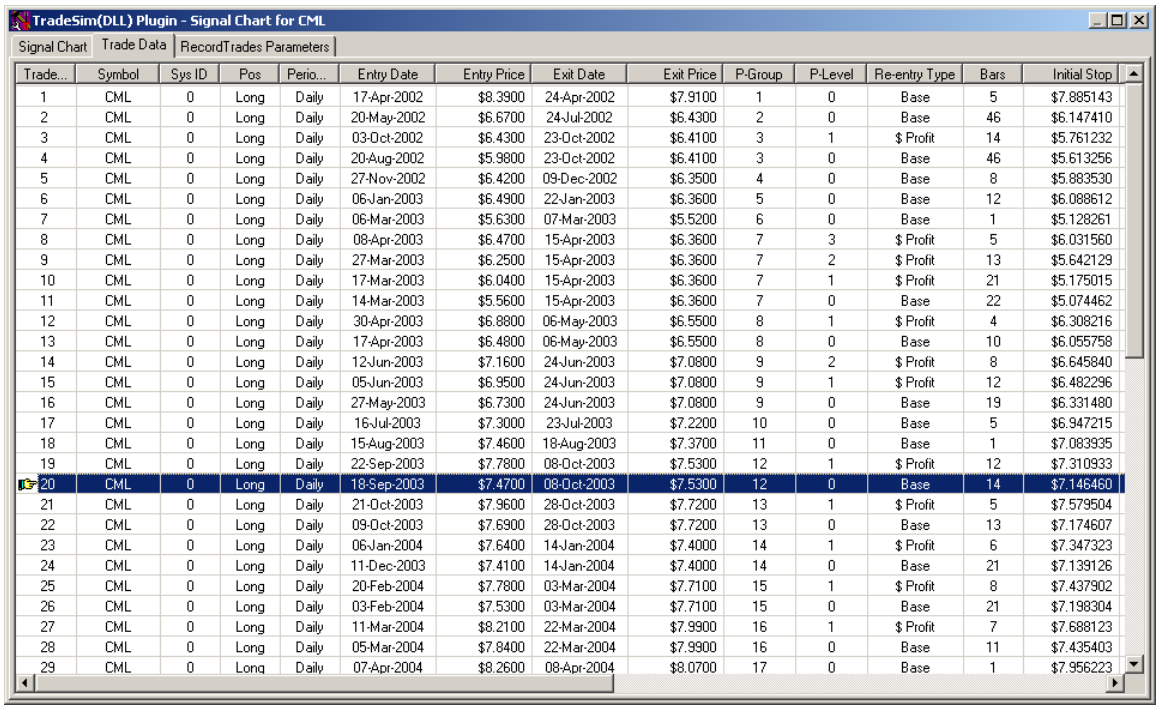
After running the exploration a new window should popup with a number of tabbed windows on it. A detailed description of the information provided in this window is described in the TradeSim User Manual.

### Trade Signals Displayed on a Bar Chart

Clicking on the "Chart" tab shows all of the trade data superimposed on a bar chart. Double clicking on any line of trade data will automatically highlight the trade data in the trade data table. Please refer to the TradeSim User Manual for more information.


### Trade Signals Displayed in Tabular form

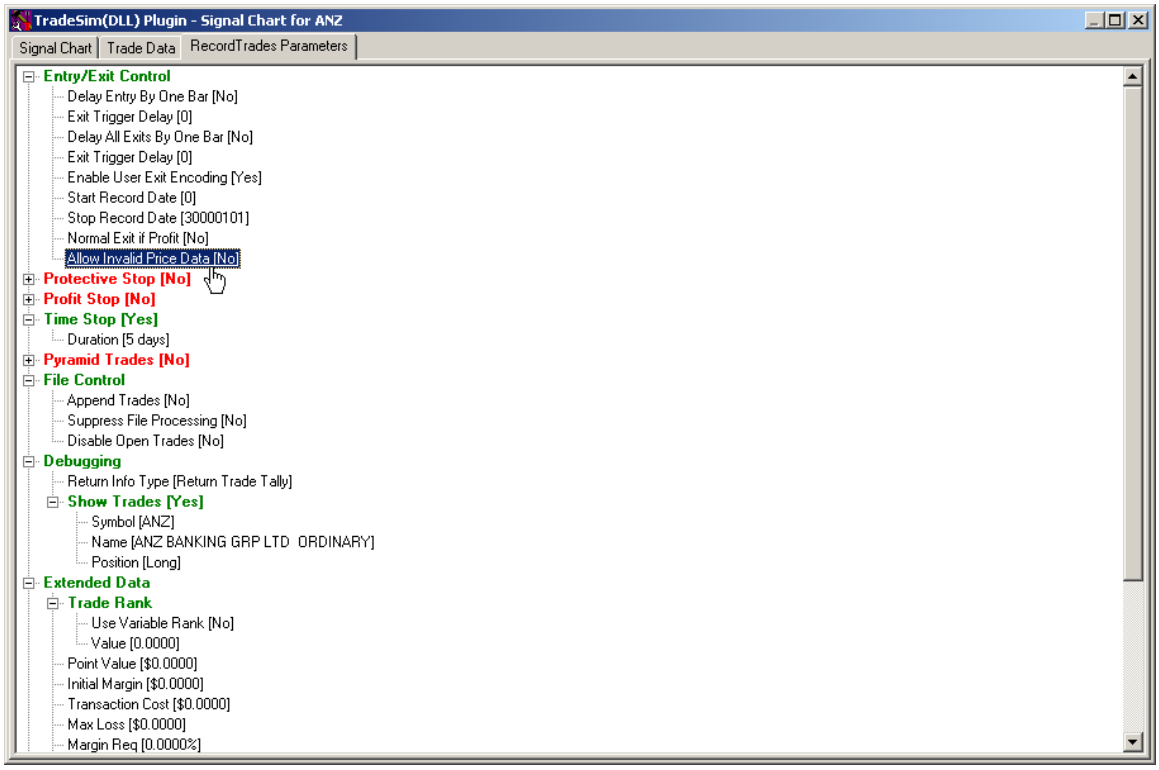
Clicking on the "Trade Data" tab shows all of the trade data displayed in tabular form. Double clicking on any line of trade data will automatically display the trade data in the chart window. Please refer to the TradeSim User Manual for more information.




| Trade... | Symbol | Sys ID | Pos | Perio... | Entry Date | Entry Price | Exit Date | Exit Price | P-Group | P-Level | Re-entry Type | Bars | Initial Stop |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | CML | 0 | Long | Daily | 17-Apr-2002 | $8.3900 | 24-Apr-2002 | $7.9100 | 1 | 0 | Base | 5 | $7.885143 |
| 2 | CML | 0 | Long | Daily | 20-May-2002 | $6.6700 | 24-Jul-2002 | $6.4300 | 2 | 0 | Base | 46 | $6.147410 |
| 3 | CML | 0 | Long | Daily | 03-Oct-2002 | $6.4300 | 23-Oct-2002 | $6.4100 | 3 | 1 | $ Profit | 14 | $5.761232 |
| 4 | CML | 0 | Long | Daily | 20-Aug-2002 | $5.9800 | 23-Oct-2002 | $6.4100 | 3 | 0 | Base | 46 | $5.613256 |
| 5 | CML | 0 | Long | Daily | 27-Nov-2002 | $6.4200 | 09-Dec-2002 | $6.3500 | 4 | 0 | Base | 8 | $5.883530 |
| 6 | CML | 0 | Long | Daily | 06-Jan-2003 | $6.4900 | 22-Jan-2003 | $6.3600 | 5 | 0 | Base | 12 | $6.088612 |
| 7 | CML | 0 | Long | Daily | 06-Mar-2003 | $5.6300 | 07-Mar-2003 | $5.5200 | 6 | 0 | Base | 1 | $5.128261 |
| 8 | CML | 0 | Long | Daily | 08-Apr-2003 | $6.4700 | 15-Apr-2003 | $6.3600 | 7 | 3 | $ Profit | 5 | $6.031560 |
| 9 | CML | 0 | Long | Daily | 27-Mar-2003 | $6.2500 | 15-Apr-2003 | $6.3600 | 7 | 2 | $ Profit | 13 | $5.642129 |
| 10 | CML | 0 | Long | Daily | 17-Mar-2003 | $6.0400 | 15-Apr-2003 | $6.3600 | 7 | 1 | $ Profit | 21 | $5.175015 |
| 11 | CML | 0 | Long | Daily | 14-Mar-2003 | $5.5600 | 15-Apr-2003 | $6.3600 | 7 | 0 | Base | 22 | $5.074462 |
| 12 | CML | 0 | Long | Daily | 30-Apr-2003 | $6.8800 | 06-May-2003 | $6.5500 | 8 | 1 | $ Profit | 4 | $6.308216 |
| 13 | CML | 0 | Long | Daily | 17-Apr-2003 | $6.4800 | 06-May-2003 | $6.5500 | 8 | 0 | Base | 10 | $6.055758 |
| 14 | CML | 0 | Long | Daily | 12-Jun-2003 | $7.1600 | 24-Jun-2003 | $7.0800 | 9 | 2 | $ Profit | 8 | $6.645840 |
| 15 | CML | 0 | Long | Daily | 05-Jun-2003 | $6.9500 | 24-Jun-2003 | $7.0800 | 9 | 1 | $ Profit | 12 | $6.482296 |
| 16 | CML | 0 | Long | Daily | 27-May-2003 | $6.7300 | 24-Jun-2003 | $7.0800 | 9 | 0 | Base | 19 | $6.331480 |
| 17 | CML | 0 | Long | Daily | 16-Jul-2003 | $7.3000 | 23-Jul-2003 | $7.2200 | 10 | 0 | Base | 5 | $6.947215 |
| 18 | CML | 0 | Long | Daily | 15-Aug-2003 | $7.4600 | 18-Aug-2003 | $7.3700 | 11 | 0 | Base | 1 | $7.083935 |
| 19 | CML | 0 | Long | Daily | 22-Sep-2003 | $7.7800 | 08-Oct-2003 | $7.5300 | 12 | 1 | $ Profit | 12 | $7.310933 |
| 20 | CML | 0 | Long | Daily | 18-Sep-2003 | $7.4700 | 08-Oct-2003 | $7.5300 | 12 | 0 | Base | 14 | $7.146460 |
| 21 | CML | 0 | Long | Daily | 21-Oct-2003 | $7.9600 | 28-Oct-2003 | $7.7200 | 13 | 1 | $ Profit | 5 | $7.579504 |
| 22 | CML | 0 | Long | Daily | 09-Oct-2003 | $7.6900 | 28-Oct-2003 | $7.7200 | 13 | 0 | Base | 13 | $7.174607 |
| 23 | CML | 0 | Long | Daily | 06-Jan-2004 | $7.6400 | 14-Jan-2004 | $7.4000 | 14 | 1 | $ Profit | 6 | $7.347323 |
| 24 | CML | 0 | Long | Daily | 11-Dec-2003 | $7.4100 | 14-Jan-2004 | $7.4000 | 14 | 0 | Base | 21 | $7.139126 |
| 25 | CML | 0 | Long | Daily | 20-Feb-2004 | $7.7800 | 03-Mar-2004 | $7.7100 | 15 | 1 | $ Profit | 8 | $7.437902 |
| 26 | CML | 0 | Long | Daily | 03-Feb-2004 | $7.5300 | 03-Mar-2004 | $7.7100 | 15 | 0 | Base | 21 | $7.198304 |
| 27 | CML | 0 | Long | Daily | 11-Mar-2004 | $8.2100 | 22-Mar-2004 | $7.9900 | 16 | 1 | $ Profit | 7 | $7.688123 |
| 28 | CML | 0 | Long | Daily | 05-Mar-2004 | $7.8400 | 22-Mar-2004 | $7.9900 | 16 | 0 | Base | 11 | $7.435403 |
| 29 | CML | 0 | Long | Daily | 07-Apr-2004 | $8.2600 | 08-Apr-2004 | $8.0700 | 17 | 0 | Base | 1 | $7.956223 |

## Parameter Tree

Clicking on the "RecordTrades Parameters" will display all of the RecordTrades parameters in tree form. Parameter tree roots marked red are ones that are not set active. Please refer to the TradeSim User Manual for more information.



# Debugging the RecordTrades function within Metastock using Indicators and Experts

This was the older way of trying to debug the RecordTrades function for versions of TradeSim earlier than version 5 and TradeSim(DLL) earlier than version 6. It is recommended that the new ShowTrades function be used instead and this section is only included for backwards compatibility with the earlier versions of TradeSim and the plugin.

Two new functions added to version 4.0.0 of the TradeSim DLL facilitate debugging the RecordTrades function.

## SuppressFileProcessing

```
ExtFml("TradeSim.SuppressFileProcessing");
```

This function is primarily used when trying to diagnose the ***RecordTrades*** function or simply creating an Indicator or Expert of the actual entry and exit triggers using the ***RecordTrades*** without actually creating a trade database. Calling this function before the ***RecordTrades*** function inhibits the ***RecordTrades*** function from generating a trade database file. The syntax for using this function is as follows:-

## SetReturnInfoType

```
ExtFml("TradeSim.SetReturnInfoType",INFO_TYPE);
```

By default the RecordTrades function returns the trade tally. This enables the total trade tally for each security to be reported in the exploration report when a trade database exploration is run. However when overlaying the RecordTrades function as an indicator on a chart the trade tally is of limited use. A more useful indicator would be to plot the actual entry or exit triggers or create an expert, which can be overlayed on a chart. The RecordTrades function can be made to report back the trigger information instead of the trade tally and a long with the ***SuppressFileProcessing*** can allow the ***RecordTrades*** function to be used as an indicator or expert without actually creating a trade database file. The syntax for using this function is as follows:-

The INFO_TYPE parameter can be one of the following four values.

- **TradeTally**
  The ***RecordTrades*** function returns the total trade tally.
- **EntryTriggers**
  The ***RecordTrades*** function returns the actual entry triggers. A value of '1' denotes an actual entry trigger and a value of '0' denotes no entry trigger.
- **ExitTriggers**
  The ***RecordTrades*** function returns the actual exit triggers. A value of '1' denotes an actual exit trigger and a value of '0' denotes no exit trigger.
- **AllTriggers**
  The ***RecordTrades*** function returns all actual entry and exit triggers. To avoid ambiguity with entry and exit triggers that occur at the same time the following encoding scheme is used.
  - '0' - No entry or exit trigger.
  - '1' - Actual entry trigger.
  - '2' - Actual exit trigger.
  - '3' - Both entry and exit trigger on the same day.

## Debugging the RecordTrades function using an Indicator.

You can use the previous functions to debug or analyse the information generated by the RecordTrades function without actually creating a trade database file.

We will start off by using our simple MACD trading system indicator and modifying it as follows.

```
EntryTrigger:=Cross(MACD(),Mov(MACD(),9,E));
EntryPrice:=CLOSE;
ExitTrigger:=Cross(Mov(MACD(),9,E),MACD());
ExitPrice:=CLOSE;

dummy:=ExtFml("TradeSim.Initialize");
dummy:=ExtFml("TradeSim.SuppressFileProcessing");          { add this line }
dummy:=ExtFml("TradeSim.SetReturnInfoType",AllTriggers);   { add this line }

ExtFml( "TradeSim.RecordTrades",
        "Debugging example",    { Trade Data Filename }
         LONG,                  { Trade Position Type }
         EntryTrigger,          { Entry Trigger }
         EntryPrice,            { Entry Price }
         0,                     { Optional Initial Stop }
         ExitTrigger,           { Exit Trigger }
         ExitPrice,             { Exit Price }
         START);                { Start Symbol }
```

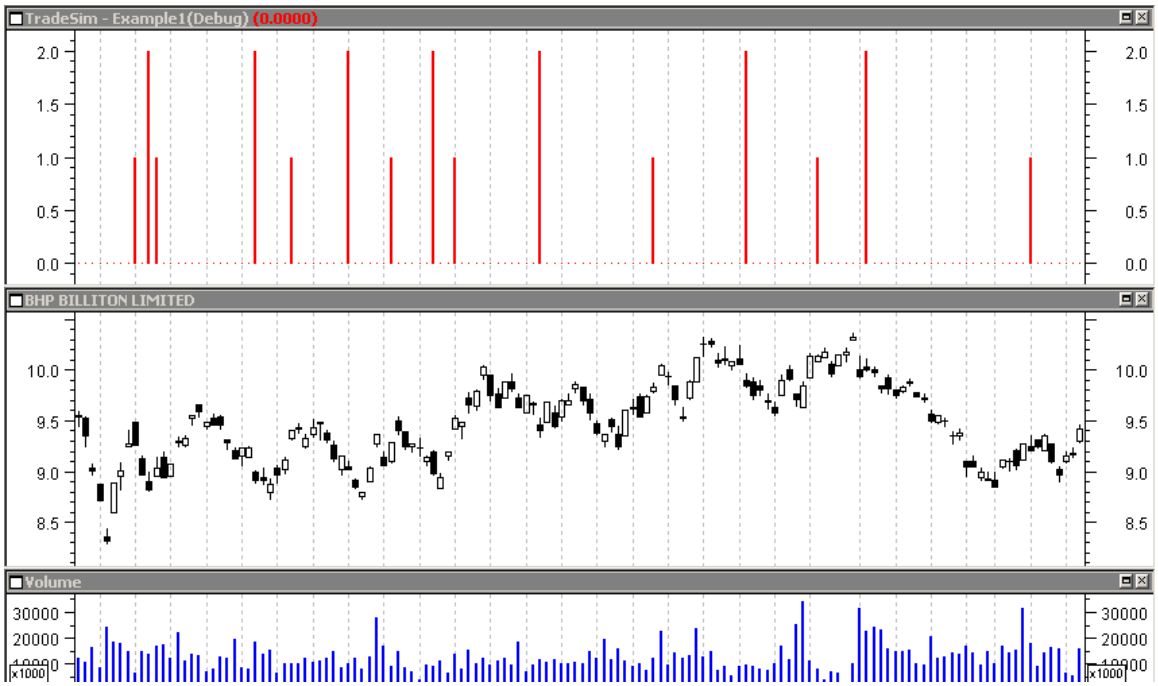
Notice the addition of the dummy assignments so that the calls to the external functions are not plotted when the indicator is overlayed on a chart. The last call to the RecordTrades function plots the actual entry and exit triggers that would have been recorded to the trade database file had not the SuppressFileProcessing function been called.

The results of overlaying this indicator on a chart are shown on the next screen shot. Note that for convenience the indicator chart has been changed to display bars rather than lines. A value of one indicates a valid entry trigger whilst a value of two indicates a valid exit trigger. Although absent from the display a value of three indicates both entry and exit triggers coinciding on the same date. The bars displayed in the indicator represent the actual trade entry and exit triggers that would have been written to a trade database if file generation were enabled.

You may have asked why not just plot the Entry and Exit trigger functions? This is because not all of the entry and exit triggers are recorded to the trade database and using the RecordTrades functions to display the entry and exit triggers allows you to visually display the actual triggers that would be written to a trade database file if file generation were enabled. As an exercise try commenting out the call to the SuppressFileProcessing function and then deleting and the overlaying indicator on the chart again. Then check the resulting trade database file by loading it into TradeSim and comparing the entry and exit triggers with those displayed on the indicator. Alternatively you could use the following indicator to plot all of the Entry and Exit Triggers.

```
EntryTrigger:=Cross(MACD(),Mov(MACD(),9,E));
ExitTrigger:=Cross(Mov(MACD(),9,E),MACD());

EntryTrigger+ExitTrigger*2;
```




## Debugging the RecordTrades function using an Expert.

You can use a similar procedure to the previous section and use an Expert instead of an Indicator to debug the RecordTrades function. You need to create two symbols - an Entry Trigger symbol, and an Exit Trigger Symbol. If you are not familiar with Experts please consult your Metastock User Manual for more information.

### The Entry Trigger Symbol Formula

```
EntryTrigger:=Cross(MACD(),Mov(MACD(),9,E));
EntryPrice:=CLOSE;
ExitTrigger:=Cross(Mov(MACD(),9,E),MACD());
ExitPrice:=CLOSE;

dummy:=ExtFml("TradeSim.Initialize");
dummy:=ExtFml("TradeSim.SuppressFileProcessing");          { add this line }
dummy:=ExtFml("TradeSim.SetReturnInfoType",EntryTriggers);  { add this line }

ExtFml( "TradeSim.RecordTrades",
        "Debugging example",    { Trade Data Filename }
         LONG,                  { Trade Position Type }
         EntryTrigger,          { Entry Trigger }
         EntryPrice,            { Entry Price }
         0,                     { Optional Initial Stop }
         ExitTrigger,           { Exit Trigger }
```

```
         ExitPrice,              { Exit Price }
         START);                 { Start Symbol }
```

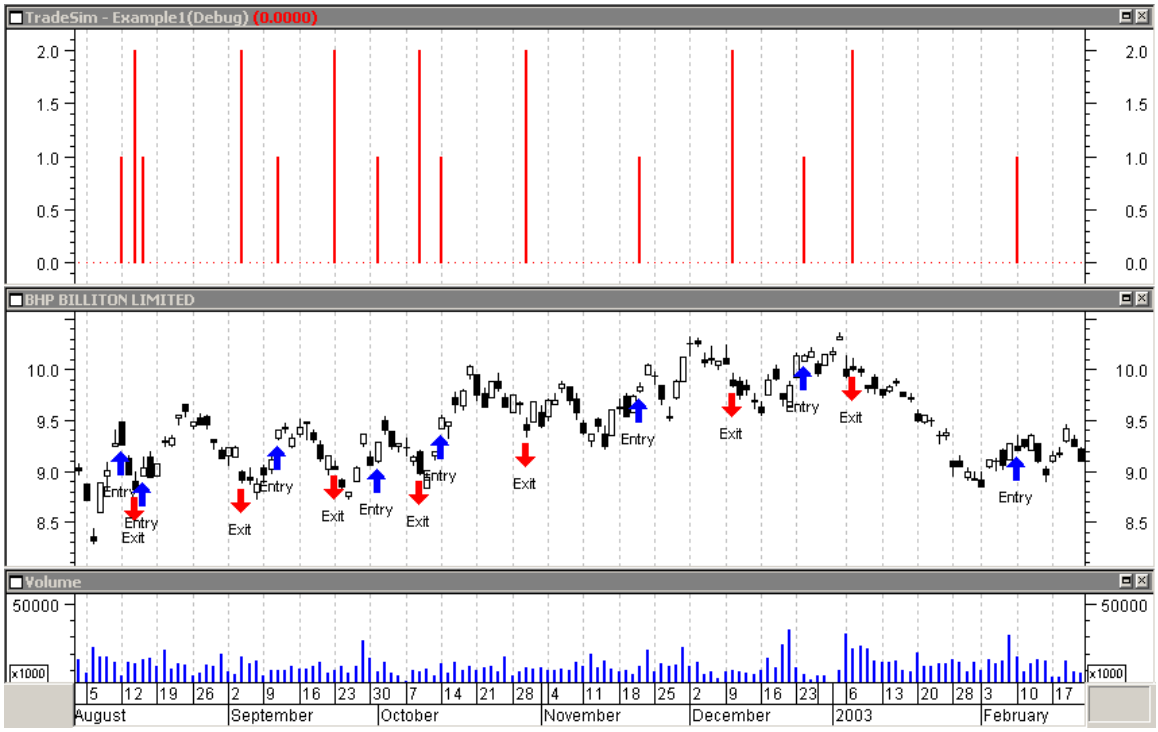
## The Exit Trigger Symbol Formula

```
EntryTrigger:=Cross(MACD(),Mov(MACD(),9,E));
EntryPrice:=CLOSE;
ExitTrigger:=Cross(Mov(MACD(),9,E),MACD());
ExitPrice:=CLOSE;

dummy:=ExtFml("TradeSim.Initialize");
dummy:=ExtFml("TradeSim.SuppressFileProcessing");          { add this line }
dummy:=ExtFml("TradeSim.SetReturnInfoType",ExitTriggers);  { add this line }

ExtFml( "TradeSim.RecordTrades",
        "Debugging example",    { Trade Data Filename }
         LONG,                  { Trade Position Type }
         EntryTrigger,          { Entry Trigger }
         EntryPrice,            { Entry Price }
         0,                     { Optional Initial Stop }
         ExitTrigger,           { Exit Trigger }
         ExitPrice,             { Exit Price }
         START);                { Start Symbol }
```

The results of attaching the Expert to the chart can be seen in the following screen shot. Note that the debug indicator has been included for comparison. Also note that the entry and exit symbols of the expert match up with the trigger bars on the indicator.



# Reference Literature

This list of references is by no means exhaustive but represents material, which is either recommended, or for general reading.

1) Compuvision Australia. TradeSim User Manual.
2) Equis. ***Metastock for Windows 95/98 & NT***. This is the user manual that comes with Metastock Version 7.0 and is a prerequisite for using TradeSim.