

TradeSim[®]

Advanced Trading Simulator and Back Tester



Application Note 2

Implementing Volatility Trailing Stops the Simple Way

-
- ✓ TradeSim Standard Edition
 - ✓ TradeSim Professional Edition
 - ✓ TradeSim Enterprise Edition
 - ✓ Metastock Version 7 and above

Last Update 20 July 2005

Contents

<u>CONTENTS</u>	<u>2</u>
<u>DISCLAIMER</u>	<u>3</u>
<u>COPYRIGHT AND LICENSING AGREEMENT</u>	<u>4</u>
Trademarks	5
Introduction	6
Volatility trailing stop on the long side	6
Trailing stop band	6
Exit condition	7
Volatility trailing stop on the short side	7
Trailing Stop Band.....	7
Exit Condition.....	7
Using the Trailing Stop with the RecordTrades function	8
Fast Trailing Stop External Function	8
Function Description	8
Trailing Stop Algorithm on the Long Side	9
Trailing Stop Algorithm on the Short Side	10
Indicator Application	10
Band Indicator Example	10
Exit Trigger Functions	12
Exit Trigger Indicator example	12
Creating Other Trailing Stop Indicators	13
Putting it all together	14
<u>REFERENCE LITERATURE</u>	<u>16</u>

Disclaimer

In no event shall Compuvision Australia or its suppliers be liable for any damage either direct or indirect, including, without limitation, damages for loss of business profits, business interruption, loss or business information or other losses arising out of the use of or inability to use the software.

The results obtained from using this software are not indicative of, and have no bearing on, any results, which may be attained in actual trading. Results of past performance are no guarantee of future performance. It should not be assumed that you would experience results comparable to that reflected by the results from this software. No assurance is given that you will not incur substantial losses, nor shall Compuvision Australia Pty Ltd be held liable if losses are incurred.

Compuvision Australia Pty Ltd is not a licensed investment advisor and so the information and results obtained by using this software is for educational purposes and of the nature of a general comment and neither purports nor intends to be, specific trading advice. The information obtained from using this software should not be considered as an offer or enticement to buy, sell or trade and is given without regard to any particular person's investment objectives, financial situation and particular needs. This software is not designed to replace your Licensed Financial Consultant or your Stockbroker. You should seek appropriate advice from your broker, or licensed investment advisor, before taking any action.

Copyright and Licensing Agreement

TradeSim is Copyright© 2000-2003 by Compuvision Australia Pty Ltd.
"TradeSim" is a trademark of Compuvision Australia Pty Ltd.

IMPORTANT - READ CAREFULLY

This license statement and limited warranty constitutes a legal agreement ("License Agreement") between you ("Licensee", either as an individual or a single entity) and Compuvision Australia Pty Ltd ("Vendor"), for the software product TradeSim® ("Software") of which Compuvision Australia Pty Ltd is the copyright holder.

BY INSTALLING, COPYING, OR OTHERWISE USING THE SOFTWARE, YOU AGREE TO BE BOUND BY ALL OF THE TERMS AND CONDITIONS OF THE LICENSE AGREEMENT.

Upon your acceptance of the terms and conditions of the License Agreement, Compuvision Australia Pty Ltd grants you the right to use the Software in the manner provided below.

If you do not accept the terms and conditions of the License Agreement, you are to promptly delete each and any copy of the Software from your computer(s).

This license agreement only applies to the software product "TradeSim" and not to any other product even if that product is similar to TradeSim and has a similar name.

The Vendor reserves the right to license the same Software to other individuals or entities under a different license agreement.

After accepting this license agreement, the Licensee is permitted to use the Software under the terms of this agreement.

Under this license agreement, the Software can only be used by those persons or entities that have purchased a license key. Payment information is available at <http://www.compuvision.com.au/PurchaseOnline.htm>.

The Software is provided "as is". In no event shall Compuvision Australia Pty Ltd be liable for any consequential, special, incidental or indirect damages of any kind arising out of the delivery, performance or use of this Software, to the maximum extent permitted by applicable law. While the Software has been developed with great care, it is not possible to warrant that the Software is error free. The Software is not designed nor intended to be used in any activity that may cause personal injury, death or any other severe damage or loss.

When errors are found in the Software, the Vendor will release a new version of the Software that may no longer contains those errors a reasonable amount of time after the Vendor is given an accurate description of those errors. Which amount of time is reasonable will depend on the complexity and severity of the errors. The Vendor will mention the release at <http://www.compuvision.com.au>, at the Vendor's option, directly contact the Licensee to announce the new release. The Licensee can then, at their option, upgrade to the latest version or to continue to use the older version the Licensee already has. In the latter case, the Licensee will no longer be entitled to technical support until the Licensee has upgraded to the latest version.

The Vendor reserves the right to charge an upgrade fee in the case of major new enhancements or additions to the Software. This major new version will then start a new version line, which will use version numbers

clearly distinguishable from the old version line. The Licensee has no obligation to upgrade to the new version line.

You must not attempt to reverse compile, modify, translate or disassemble the Software in whole or in part. You must not run the Software under a debugger or similar tool allowing you to inspect the inner workings of the Software.

The Software remains the exclusive property of the Vendor. Any Licensee, which fully complies with the terms in this license agreement, may use it according to the terms of this license agreement. You must not give copies of the Software or your license key to other persons or entities. If you have received a download password or an URL with an embedded password for downloading the Software, you must keep this password secret. You must also take reasonable steps to prevent any third party from copying the software from one of your machines without your permission.

The Vendor reserves the right to revoke your license if you violate any or all of the terms of this license agreement, without prior notice.

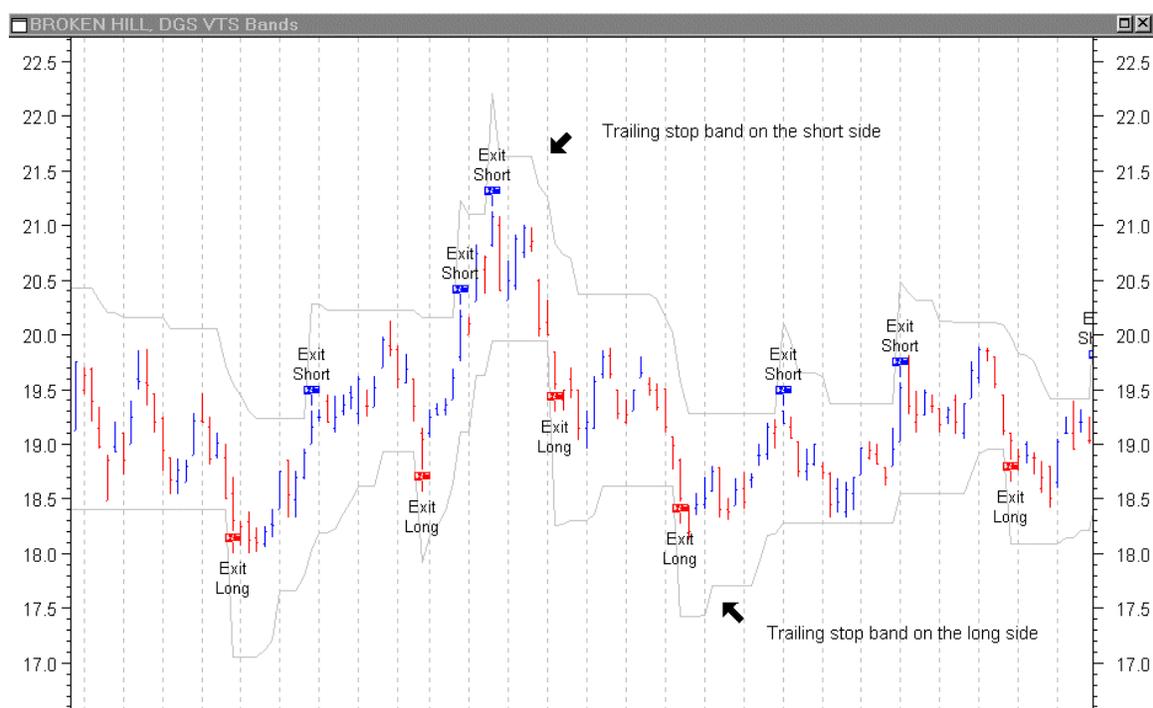
Trademarks

1. TradeSim® is a registered trademark of Compuvision Australia Pty Ltd.
2. Metastock® is a registered trademark of Equis International.
3. Microsoft Windows® is a registered trademark of Microsoft Corporation.
4. Microsoft Excel® is a registered trademark of Microsoft Corporation.
5. Word for Windows ® is a registered trademark of Microsoft Corporation.
6. TradeStation® is a registered trademark of Omega Research Corporation.

Introduction

Volatility trailing stops are useful in developing a trading system, which adheres to strict money management guidelines. The main benefits of a trailing stop is that the stop moves in favour of the trade whilst minimising risk and maximising profits.

The volatility-trailing stop is usually used in risk based position size models. This type of stop has a three-fold purpose. Firstly it is used to establish an initial risk or money management stop. Secondly the initial risk is used to derive the position size for the trade. Thirdly the trailing stop is used to protect capital of the trade whilst minimising risk and maximising profit. An example of its use as an indicator is shown below. Note that the bands precede the exit conditions by one bar. This is because the exit conditions reference the previous days trailing stop value. This reason for this will become clearer later when we discuss the indicators in a bit more detail.



Volatility trailing stops can be implemented using the Metastock indicator language. However the use of the PREV function in defining this indicator pays a considerable penalty regarding computation speed so bare this in mind if you decide to use these types of stops in your indicators.

Volatility trailing stop on the long side.

Trailing stop band

The following indicator formula defines a volatility-trailing stop band on the long side.

```

Period:=10;           { ATR period }
ARC:=3.0;            { Average range constant }
Stop:=C-ARC*ATR(Period);
BandLong:=If(L<=PREV,           { is stop reached?}

```

```

    Stop,                { yes: restart plot }
    If (Stop>PREV,       { no: if new stop>prev stop?}
        Stop,           { yes: plot new higher stop }
        PREV)           { no: plot previous stop}
);

```

The BandLong formula above can be condensed into one line as follows.

```
BandLong:=If (L<=PREV, Stop, If (Stop>PREV, Stop, PREV));
```

The ATR period defines the interval for which the Wilders Average True Range function is computed. Different values can be used depending on the market conditions. Slowly changing markets would benefit from using a longer period. However a 10 bar period is a fairly good compromise. The Average Range Constant (ARC) can be adjusted according to market conditions. Values between 1.8 and 3.0 should be used. A too lower value for the ARC should be avoided because it will whipsaw you in and out of the market too often. Very large values of the ARC should be avoided because you will not get stopped out in time and most of the profit will be lost. The aim is to try to protect capital and maximize profits without being whipsawed out of the markets too often.

Exit condition

The indicator above generates a trailing stop band on the long side. An exit condition is generated on the day when the low of the day crosses the stop band of the following day. To write this as an indicator we need to use the REF function to reference the stop band from the previous day. The following indicator demonstrates how this is done.

```
ExitLong:=(L <= ref (BandLong, -1));
```

Volatility trailing stop on the short side.

Trailing Stop Band

The following indicator formula defines a volatility-trailing stop on the short side.

```

Period:=10;
ARC:=3;
Stop:=C+ARC*ATR(Period);
BandShort:=If( H >= PREV ,      { is stop reached? }
    Stop ,                      { yes: restart plot }
    If (Stop < PREV,          { no: if new stop>prev stop? }
        Stop,                { yes: plot new stop }
        PREV)                { no: plot previous stop}
);

```

The BandShort formula above can be condensed into one line as follows.

```
BandShort:=If( H >= PREV , Stop , If (Stop < PREV, Stop, PREV));
```

Exit Condition

The indicator above generates a trailing stop band on the short side. An exit condition is generated on the day when the high of the day crosses the stop band of the following day. To write this as an indicator we

need to use the REF function to reference the stop band from the previous day. The following indicator demonstrates how this is done.

```
ExitShort:=(H >= ref(BandShort,-1));
```

Using the Trailing Stop with the RecordTrades function

To use the trailing stop with the RecordTrades function is fairly straightforward. The example below illustrates the use of the trailing stop on the long side.

```
EntryTrigger := Cross(MACD(),Mov(MACD(),9,E));
EntryPrice := CLOSE;
ExitTrigger := (L <= ref(BandLong,-1));
ExitPrice := CLOSE;
InitialStop := BandLong;
```

The following example below illustrates the use of the trailing stop on the short side.

```
EntryTrigger := Cross(Mov(MACD(),9,E),MACD());
EntryPrice := CLOSE;
ExitTrigger := (H >= ref(BandShort,-1));
ExitPrice := CLOSE;
InitialStop := BandShort;
```

Fast Trailing Stop External Function

In the previous section we used the Metastock indicator language to construct a trailing stop function. One shortcoming of this trailing stop is the use of the notoriously slow PREV function, which is used no less than three times in the trailing stop indicator. This may not be a problem when overlaying the indicator on a single chart but doing a Trade Database exploration containing hundreds of securities with a maximum of ten thousand records loaded for each security can tax the computers capabilities quite dramatically so much so that it can take quite along time to complete an exploration.

To overcome the speed problem the external Plug-In contains a hard coded shell of a primitive trailing stop function. The advantage of hard coding this function is that it no longer relies on the PREV function but will still allow the user to use the Metastock indicator language to construct the type of trailing stop function so it is not just limited to the popular ATR type of trailing stop.

Function Description

The external trailing stop function shell takes on the following syntax.

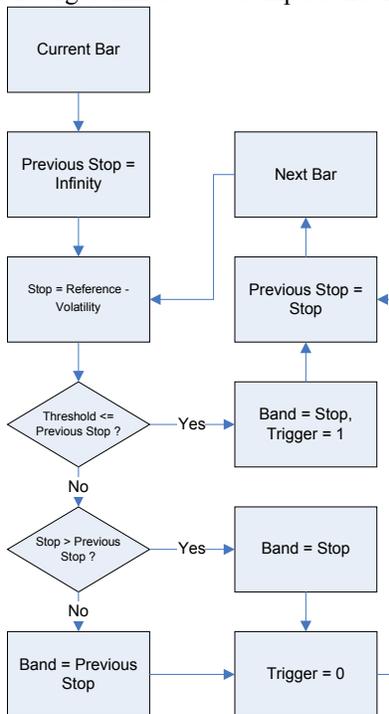
```
ExtFml("TradeSim.TrailingStop", { 1. Name of external function }
    Mode, { 2. Mode}
    TradePosition, { 3. Trade Position Type }
    VolatilityFunction, { 4. Volatility Function }
    RefPoint { 5. Reference Point }
    ThresholdPoint { 6. Threshold Point }
```

};

Parameter	Description	Allowable Values
1	The name of the external function along with the name of the external DLL that contains it	"TradeSim.TrailingStop "
2	The Mode parameter determines whether the function is used to generate a trailing stop band indicator or a binary stop trigger function.	BAND – displays a Trailing Stop band indicator. TRIGGER – generates a binary trigger where a value of '1' indicates a stop trigger.
3	The TradePosition parameter specifies whether the trade is on the long or short side.	LONG or SHORT
4	The VolatilityFunction parameter is a user defined volatility function for determining the type of trailing stop.	For example 3*ATR(10) gives a 10 bar ATR trailing stop with an average range constant of 3.
5	The RefPoint parameter determines the point or price where the trailing stop is calculated. It is typically set to the closing price however it could be set to HIGH for the long side or to LOW for the short side.	CLOSE, OPEN, HIGH, LOW or any arbitrary data array.
6	The ThresholdPoint parameter determines the point at which the stop band has been breached and the the trailing stop band is reset.	LOW, CLOSE, HIGH, or any arbitrary data array. Typically you would use LOW for the long side and HIGH for the side short side.

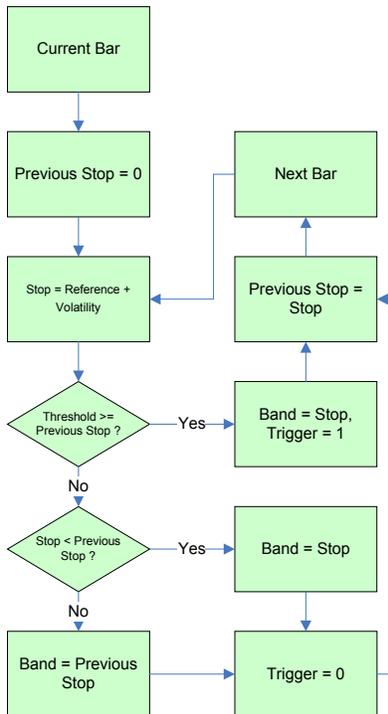
Trailing Stop Algorithm on the Long Side

The following flowchart illustrates the algorithm used to compute the trailing stop on the long side.



Trailing Stop Algorithm on the Short Side

The following flowchart illustrates the algorithm used to compute the trailing stop on the short side.



Indicator Application

The following indicator example plots two ATR trailing stop bands on a chart. The top band corresponds to the trailing stop on the short side. The bottom band corresponds to the trailing stop on the long side.

Band Indicator Example

```

Period:=Input("Period",3,50,10);
ARC:=Input("Average Range Constant",0.5,10,3);

Volatility:=ARC*ATR(Period);

{ Trailing Stop Band Indicator on the Long side }

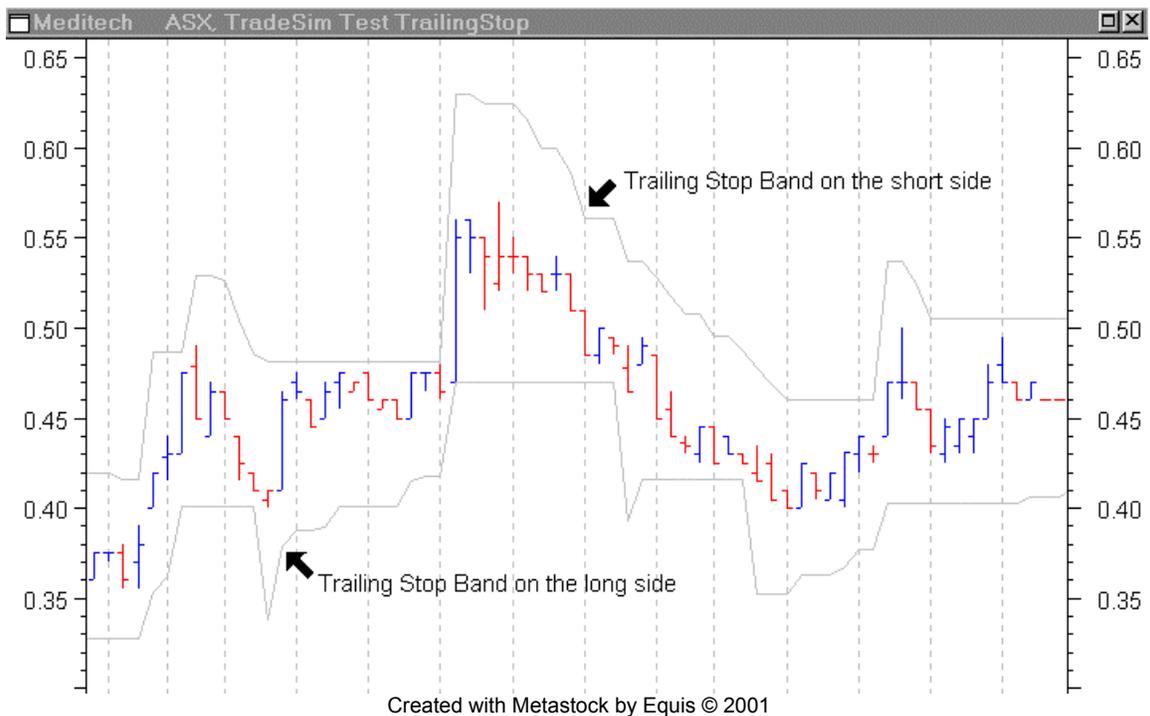
BandLong:=ExtFml("TradeSim.TrailingStop",
BAND,
LONG,
Volatility,
CLOSE
LOW);
  
```

```
{ Trailing Stop Band Indicator on the Short side }
```

```
BandShort:=ExtFml( "TradeSim.TrailingStop",  
BAND,  
SHORT,  
Volatility,  
CLOSE,  
HIGH);
```

```
{ Display the bands }
```

```
BandLong;  
BandShort;
```



When overlaying the above indicator on the chart the bands appear to pre-empt the stop triggers. This is perfectly normal, as the trigger condition is not known until the next day. You can shift the bands one bar to the right to match up the crossing of the band with the trigger condition by replacing the last two lines with the following lines;

```
Ref (BandLong, -1) ;  
Ref (BandShort, -1) ;
```

Exit Trigger Functions

When the Mode parameter in the trailing stop function is set to TRIGGER the function returns a binary value based on a stop trigger event.

When TradePosition=LONG a value of '1' indicates an exit on the long side. It is equivalent to writing the BandLong indicator above as follows;

```
ExitLong:=(ThresholdPoint <= ref(BandLong,-1));
```

Typically you set ThresholdPoint to be LOW.

When TradePosition=SHORT a value of '1' indicates an exit on the short side. It is equivalent to writing the BandShort indicator above as follows;

```
ExitShort:=(ThresholdPoint >= ref(BandShort,-1));
```

Typically you set ThresholdPoint to be HIGH.

Exit Trigger Indicator example

The following example uses the Trailing Stop function to plot a binary waveform. To distinguish the ExitLong from the ExitShort trigger the ExitLong trigger is negated so that it appears as a '-1' on the chart in red, whereas an ExitShort trigger appears as a +1 on the chart in blue.

```
Period:=Input("Period",3,50,10);
ARC:=Input("Average Range Constant",0.5,10,3);

Volatility:=ARC*ATR(Period);

{ Trailing Stop Exit on the Long side }

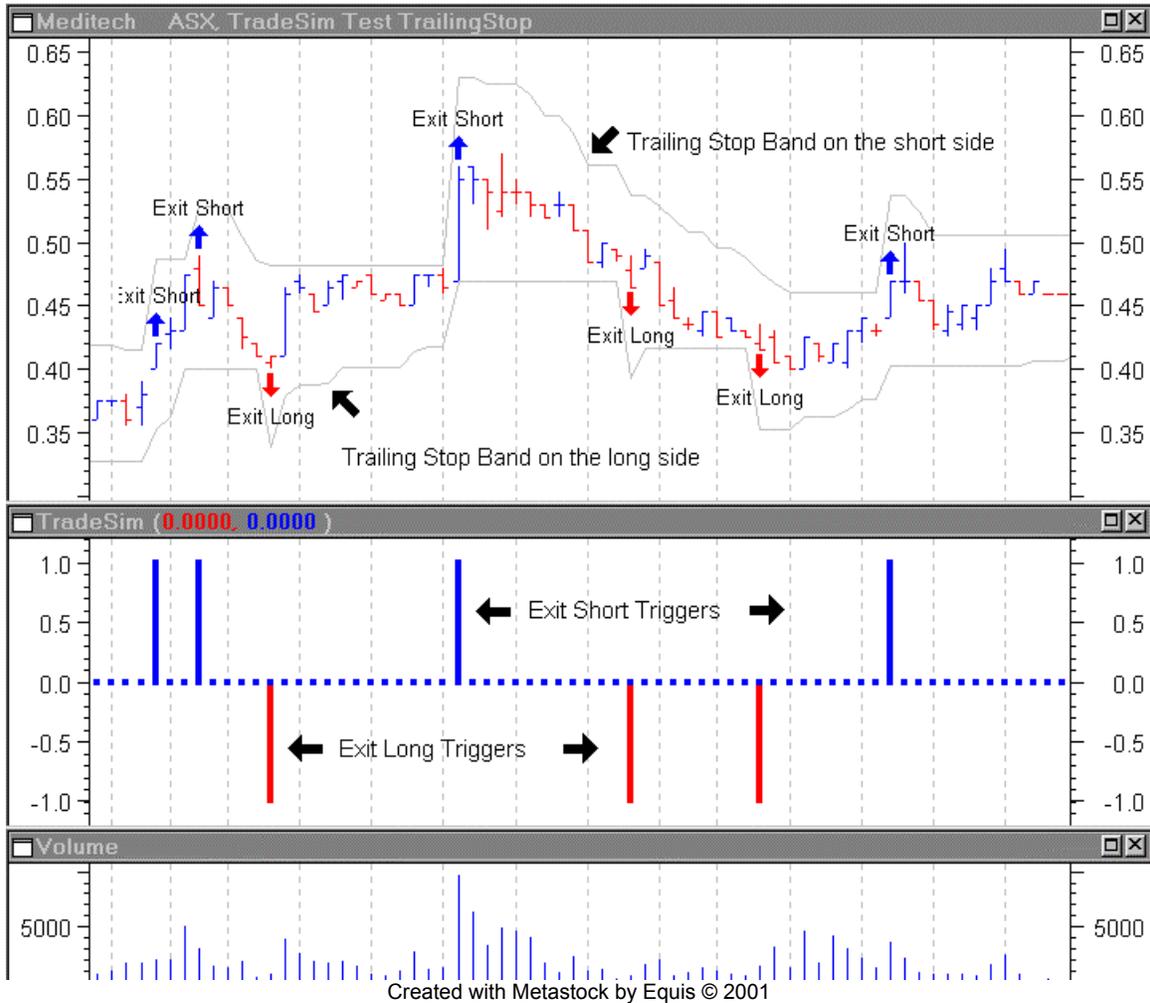
ExitLong:=-ExtFml( "TradeSim.TrailingStop",
TRIGGER,
LONG,
Volatility,
CLOSE,
LOW);

{ Trailing Stop Exit on the Short side }

ExitShort:=+ExtFml( "TradeSim.TrailingStop",
TRIGGER,
SHORT,
Volatility,
CLOSE,
HIGH);
```

```
{ Plot the triggers }
```

```
ExitLong;  
ExitShort;
```



Creating Other Trailing Stop Indicators

The Trailing Stop external function is not limited to creating ATR style-trailing stop functions only. You can quite easily use other functions such as standard deviation or even moving averages to construct a different style of trailing stop. The idea is to use a function that represents the volatility of the underlying security. The following example shows how to create a Two Standard Deviation Trailing Stop function using the external TrailingStop function.

```
{ Two Standard Deviation Trailing Stop Exit on the Long side }
```

```
Volatility:=2*Stdev(CLOSE,10);
```

```
BandLong:=ExtFml( "TradeSim.TrailingStop",
```

```

BAND,
LONG,
Volatility,
CLOSE,
LOW);

```

Putting it all together

The following example illustrates the use of the Trailing Stop function in a complete trade database exploration for long trades only.

```

Period:=10;
ARC:=3;  { Average Range Constant }
Volatility:=ARC*ATR(Period);

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

{ Trailing Stop Exit on the Long side }
ExitTrigger:=ExtFml("TradeSim.TrailingStop",
TRIGGER,
LONG,
Volatility,
CLOSE,
LOW);

ExitPrice := CLOSE;
InitialStop := 0;           { optional initial stop }

ExtFml("TradeSim.Initialize");  { Initialize internal variables }
ExtFml("TradeSim.RecordTrades",
"Trailing Stop Example", { Trade Data Filename }
LONG,                    { Trade Position Type }
EntryTrigger,            { Entry Trigger }
EntryPrice,              { Entry Price }
InitialStop,             { Initial Stop }
ExitTrigger,             { Exit Trigger }
ExitPrice,               { Exit Price }
START);                  { Trade Recorder Control }

```


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.