Trading Blox User's Guide

Trading Blox User's Guide

By Traders... For Traders

Trading Software for Mechanical Systems Traders

Trading Blox LLC
www.tradingblox.com
508 SE Osceola St
Stuart, FL 34996
978-222-3111

Table of Contents

Part I

Part 1 – Getting Started

Section 1 – Overview - What is Trading Blox?

Trading Blox User's Guide
Program Version: 4.3.0
Help Version: Thursday, December 19, 2013

Trading Blox is a Windows-based software application that runs simulations of trading rules over historical end-of-day futures, stock, or forex price data. It determines the trades that those rules would have entered and the performance they would have exhibited.

It contains the rules for several well-known trading systems and allows users to change, test, and improve upon the specific input parameter values of the original system. Once a particular combination of parameter values has been historically tested and observed to produce acceptable results, then that same system can be traded live. With the click of a menu, Trading Blox generates the next day's entries and position sizes for you.

Each system supplied by Trading Blox has many input parameters whose values can be freely changed by the user. Some of these parameters have nothing to do with the system rules themselves, such as start and end date or the market. Your testing can span 3 months or 30 years! Other parameters affect how the system works, such as the number of days used to calculate a moving average.

By default, Trading Blox is set up with the parameter values that reflect a good starting point for testing each of the supplied systems. In addition, Trading Blox lets you to take these values and improve upon them, making it easy for you to create your own unique trading systems. It also provides the comfort and security that come from experiencing firsthand how those systems performed across various historical periods.

At Trading Blox, we understand that any trading system is only as good as the consistency with which it is applied. By harnessing the speed and reliability of modern-day computer hardware, Trading Blox significantly reduces the effort required to test and to trade a system, while eliminating the possibility of human error and missed trades. And therein lies its greatest strength: Trading Blox makes it easier to follow the trading systems you develop.

This manual is used by all of our products:
- Trading Blox Turtle
- Trading Blox Professional
- Trading Blox Builder

Some of the content may or may not be applicable to all applications. We have tried to make note of that, but if we missed something please let us know. In general, all the products have all the functionality here with the exception of the number of systems. The Trading Blox Turtle system has the Turtle system, the Donchian System, and the Triple Moving Average system, while the Trading Blox Professional system has those plus the ATR Channel Breakout, Bollinger Breakout, Bollinger Counter-trend, and Dual Moving Average. Trading Blox Builder has all systems and features listed in this manual.

Section 2 – Company Philosophy

The Trading Blox team brings together a unique blend of world-class software design talent and real-world experience in trading, system design, testing, and implementation. We are committed to developing software to help others fulfill their commitment to become better traders.

Trading Blox was conceived by traders for traders based on our observation that today's trading system software does not adequately address the needs of the user community at large. This is true not only for discretionary traders who are interested in learning to trade systematically, but also for systematic traders who don't have the tools at their disposal to help them sharpen their craft and aren't sure which way to turn.

Today's publicly available back testing software falls into two distinct categories. The first type offers a slew of indicators and advanced graphic capabilities, but only allows testing on a single stock or commodity at a time, and requires expensive add-on software to test across an entire portfolio. Even then, this type of software is cumbersome and limited at best.

At the other end of the spectrum is software that allows portfolio-level back testing with money management, but which assumes significant programming skills on behalf of the user and only offers flexibility at the price of a steep learning curve. In short, with these products comprehensive testing is possible but painful.

This leaves a sizable gap between what trading system software does today, and what is truly possible. Let's face it: Until now, the ability to conduct comprehensive, full-featured back tests has been the exclusive province of a relatively small group of die-hard programmers and large-scale investment houses.

It doesn't have to be that way.

The team at Trading Blox is uniquely positioned to change this situation, and with the release of Trading Blox Builder, Professional, and Turtle, we've done just that. By making the Trading Blox technology available publicly, others will now have the opportunity to quickly develop a better understanding of how trading systems work in depth, in detail, and in the real world.

Trading Blox is a world-class trading simulation technology that provides traders of all skill levels with access to an extraordinarily flexible and highly automated testing environment. By coupling a comprehensive, programming-free testing environment with an intuitive graphical interface, Trading Blox dramatically flattens the steep learning curve traditionally associated with trading system development. It helps the trader quickly gain the necessary level of confidence to successfully deploy and consistently trade a system.

While Trading Blox bridges the gap in existing offerings, it also provides capabilities that are not available in any other software, period. For example, you can easily instruct Trading Blox to test the pyramiding techniques and risk management rules that are an integral part of the Turtle Trading Rules, without writing a line of code. You can test a triple moving average system with sophisticated money management without a line of code. You can optimize those systems for different portfolios without a line of code.

At Trading Blox, our goal is to ensure that each and every user is completely satisfied with their entire Trading Blox experience, from purchase through trading. Moreover, we look forward to establishing long-term relationships with interested users, because it is from this group that many great ideas for future releases sprout. We value your feedback!

In addition to placing a high value on user input, everything we do is driven by the belief that quality is paramount. Every release is rigorously tested to ensure that every trade that should be taken is taken, trades which should not be taken are ignored, and that every single calculation is accurately performed and reflected in the results, down to the last penny.

If we can't produce quality software that helps traders become better traders, then we have no business being in business. With that said, we believe that the future looks very bright for you and all of us at the Trading Blox family!

Section 3 – Installing and Running Trading Blox

System Requirements

Trading Blox runs on standard Windows operating systems like Windows 7 and Windows 8. Trading Blox also runs with the Parallels, Bootcamp, or Virtual Box software on Intel based Macintosh computers. No additional software or equipment is needed other than the Virtualization Software and Microsoft Operating System. Trading Blox runs on 32 bit computers, and can also run native on x64 computers.

If using an Intel based Mac, be sure to install Parallels or Bootcamp with your choice of Windows Operating System. Trading Blox can then be install in this virtual windows environment.

A minimum screen resolution 1024 x 768 or larger is required at 96dpi. If using a larger size, such as 125% or 150% with Windows 7 a larger screen resolution will be required. A minimum of 2 GB of RAM and a 2 GHz processor speed are recommended for normal testing. If you are testing with large stock portfolios or huge amounts of intraday data, we recommend at least 8 GB or more of RAM and a quad core x64 computer running the native x64 version of Trading Blox.

Trading Blox's Parameter Stepping feature allows for the concurrent testing of multiple distinct parameter value combinations. These kinds of tests finish significantly faster on high-performance machines. Ideally a machine would have 4 GB or more of RAM and a processor speed well over 3 GHz. Because Trading Blox uses is multi threaded, a quad core or more computer is quite handy for large stepped tests.

Trading Blox is very fast, however, and will run reasonably well on an older machine. The new version 4 does not support XP or Vista.

Installation:

Trading Blox customers may download the Trading Blox installer from the www.tradingblox.com web site. Once a customer has purchased the software, they will automatically receive an email with a link to the download location, and the license keys required to install the software. The install takes a few minutes, and is completely automated.

Clicking on the download link in the email will start the download of the installer. The installer will run when the download is completed, and the installation process is automatic by default. Once the installation is completed, the customer enters their License Name and License Key from email, and the program is up and running.

Trading Blox can be launched by clicking on the Start button to bring up the Start Menu, go to Programs | Trading Blox | Trading Blox.

When Trading Blox is first installed it will be necessary for the software to reach the internet servers at Trading Blox so the software can validate the license key code. In most installation that do not use any proxy servers to reach the internet there won't be any additional changes required to get the software key validate and allow Trading Blox to run.

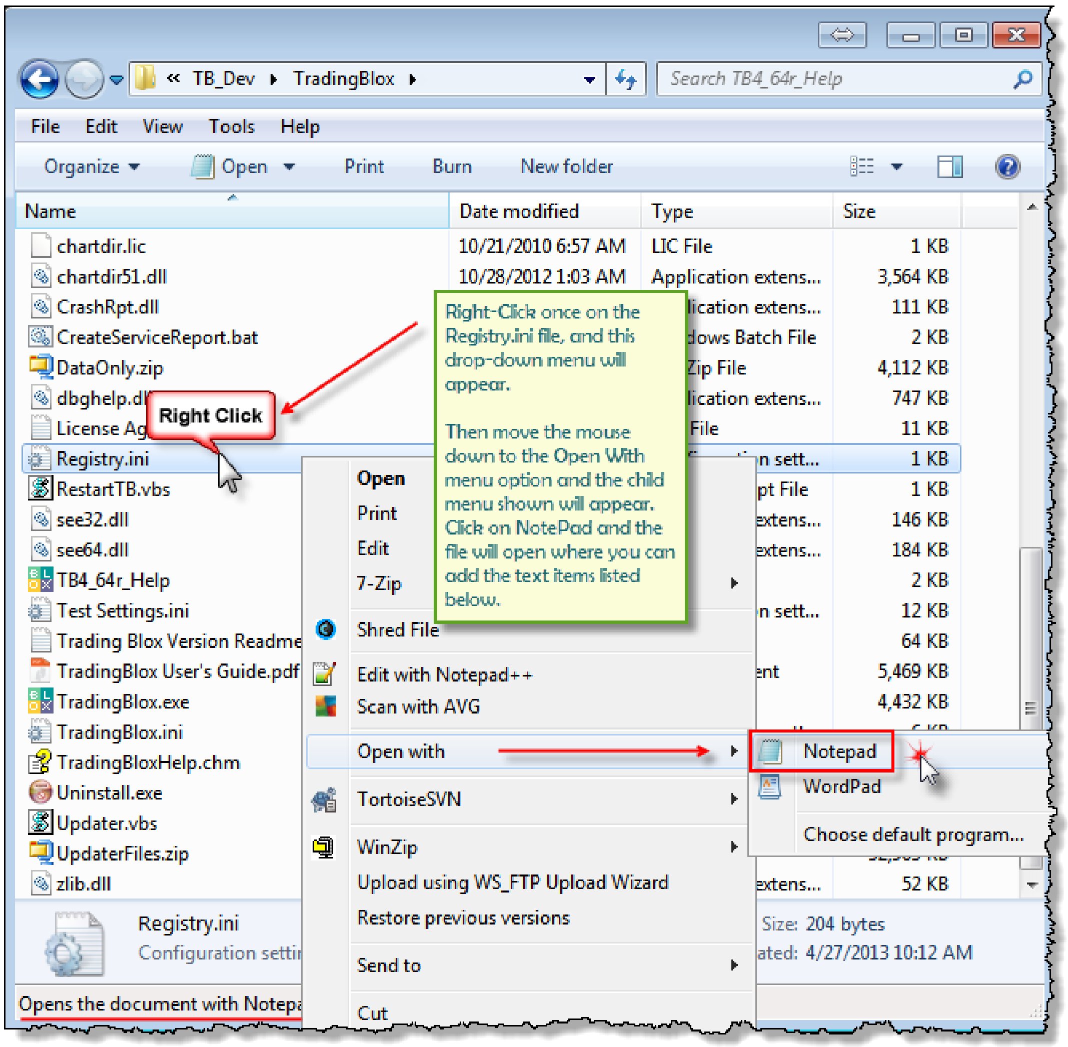
Proxy Settings:

When a Proxy server is being used for security, controlling which applications are allowed to access the internet, or any other reason, it is likely that some manual changes to the Trading Blox "**`Registry.ini`**" file will be required. These changes are simple, but they require the user to collect the following information prior to attempting to run Trading Blox.

Required Information:	Replace These Values Proxy Server Values:

`proxyServer`	`myProxyServerName` **This is the name administrators have given to your proxy server.**
`proxyPort`	`myProxyPortNumber` This is an integer value that is assigned to the proxy server port made available to allow access.
`proxyUserName`	`myProxyUserName` Each user is usually given a different User-Name.
`proxyPassword`	`myProxyPassword` Each user is usually given a unique password to obtain access.

When the above information has been obtained, locate the "`Registry.ini`" file in your Trading Blox installation directory:



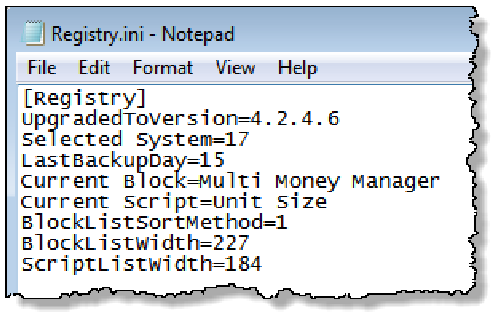
Name	Date modified	Type	Size
chartdir.lic	10/21/2010 6:57 AM	LIC File	1 KB
chartdir51.dll	10/28/2012 1:03 AM	Application extens...	3,564 KB
CrashRpt.dll		lication extens...	111 KB
CreateServiceReport.bat		dows Batch File	2 KB
DataOnly.zip		Zip File	4,112 KB
dbghelp.dl		lication extens...	747 KB
License Ag		File	11 KB
Registry.ini		n sett...	1 KB
RestartTB.vbs		pt File	1 KB
see32.dll		extens...	146 KB
see64.dll		extens...	184 KB
TB4_64r_Help			2 KB
Test Settings.ini		n sett...	12 KB
Trading Blox Version Readme			64 KB
TradingBlox User's Guide.pdf		ent	5,469 KB
TradingBlox.exe			4,432 KB
TradingBlox.ini			
TradingBloxHelp.chm			
Uninstall.exe			
Updater.vbs			
UpdaterFiles.zip			
zlib.dll		extens...	52 KB

Right-Click once on the Registry.ini file, and this drop-down menu will appear.

Then move the mouse down to the Open With menu option and the child menu shown will appear. Click on NotePad and the file will open where you can add the text items listed below.

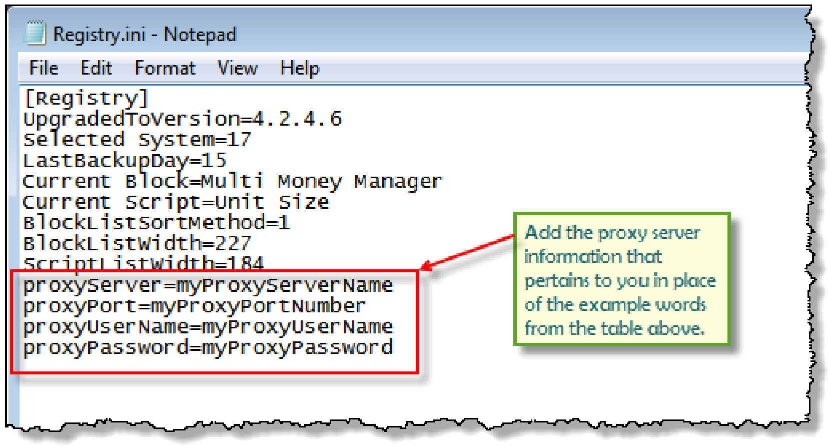
Registry.ini
Configuration setti
Size: 204 bytes
ated: 4/27/2013 10:12 AM
Opens the document with Notepa

When NotePad opens the file it will look like this:



```
[Registry]
UpgradedToVersion=4.2.4.6
Selected System=17
LastBackupDay=15
Current Block=Multi Money Manager
Current Script=Unit Size
BlockListSortMethod=1
BlockListWidth=227
ScriptListWidth=184
```

Using the table above made available for collecting the details needed, add these four lines of text using your own information on the right-side of the equal sign "=":


Registry.ini - Notepad
File Edit Format View Help
Add the proxy server information that pertains to you in place of the example words from the table above.

```
[Registry]
UpgradedToVersion=4.2.4.6
Selected System=17
LastBackupDay=15
Current Block=Multi Money Manager
Current Script=Unit Size
BlockListSortMethod=1
BlockListWidth=227
ScriptListWidth=184
proxyServer=myProxyServerName
proxyPort=myProxyPortNumber
proxyUserName=myProxyUserName
proxyPassword=myProxyPassword
```

Once the correct proxy information for you location is entered, Trading Blox should not have any trouble accessing the internet and validating the license key code.

Note:

When editing the "`Registry.ini`" file be sure to not allow any blank lines between any of the records.

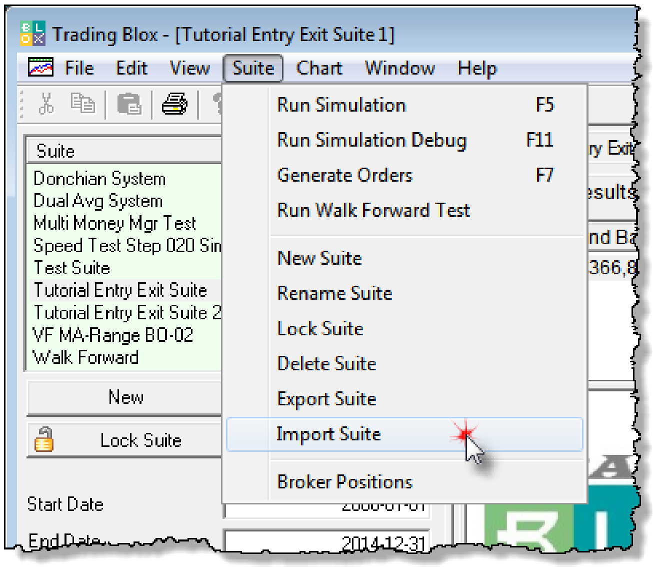
Section 4 – Importing Suite Files

Import Suite Purpose:

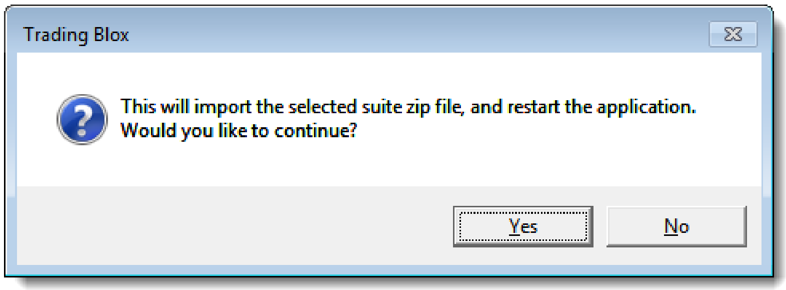
Tested Suites contains numerous files and settings that are required to enable the system perform as it was intended when it was created. In order to prevent a lot of checking and parameter setting effort Suites imported using the Suite Import menu option will retain all the settings and the files assigned when when the suite was last tested.

Import Suite Instruction:

1. Click on and then Copy the Import Suite Zip file package you wish to import into Trading Blox.
2. Click on Suite Menu and then click on the Import Suite option.



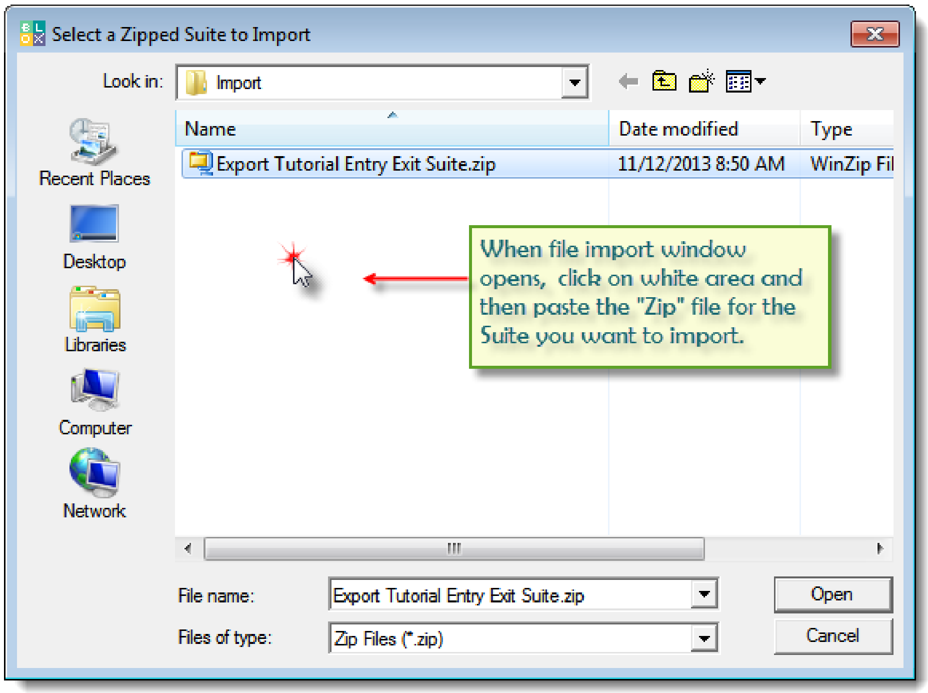
3. This next message will appear informing that you are about to import a Suite file collection and when it has imported the contents of the compressed Zip file, Trading Blox will close all active test results, close the software and then restart itself.



This will import the selected suite zip file, and restart the application.
Would you like to continue?

4. In most cases this isn't an issue unless you have some test results you wish to print or locate in the Results folder.
5. Click the "Yes" button if you want to proceed at this time. Click "No" if you want to go back and do something with the results displayed.
6. If you click "Yes" a file dialog window will appear providing you with a location where you can

paste the suite file package you just copied. Click anywhere on the open white space and the Paste the file package into the open dialog.



Select a Zipped Suite to Import

Look in: Import

Name	Date modified	Type
Export Tutorial Entry Exit Suite.zip	11/12/2013 8:50 AM	WinZip Fil

When file import window opens, click on white area and then paste the "Zip" file for the Suite you want to import.

File name: Export Tutorial Entry Exit Suite.zip

Files of type: Zip Files (*.zip)

7. You should see the name of the suite import package file in the white are of the dialog and in the Open File name area.
8. Click on the Open button and Trading Blox will unpack the Zip file and then import all of the contents into the Trading Blox folders as required.
9. After Trading Blox has closed and then reappeared again, you should see the Suite file name you just imported.

Caution:

Files exported with the same name as those that are normally provided with a Trading Blox installation will be overwritten when the Suite is imported in another installation. This concern is for users who might have made modifications to some of the Blox or System module files and didn't change the name of file in some way that it is not longer the default name.

Special files that are not normally part of the Trading Blox process, like special index or weighting files that are not contained with a Blox module will not be exported and then will not be available for export.

Section 5 – Exporting Suite Files

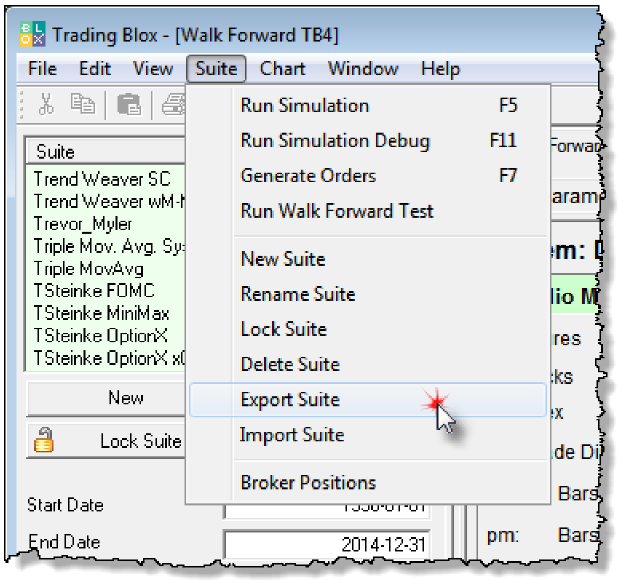
Export Suite Purpose:

Suite construction most often contains numerous files and settings that can have a significant effect on how a system performs and the results it creates.

In order to reduce the time it takes to resolve questions and issue it is necessary that Suites that generate questions and issues be contained in their entirety so that the process used to emulate an incomplete Suite process doesn't become the reason the issues don't appear as they do in the trader's installation.

Export Suite Instruction:

1. Click on the Suite that is creating the questions and issues and run a simulation.
2. Look to see if the issues are present and if they are present, click on the Suite Menu and then click on the Export Suite option shown in the image above.
3. When the dialog appears, look for the Zip file, not the folder file that has the same name as the Suite.



4. Click on the "suite-Same-Name.zip" file and then select copy, or press the "Control+C" at the same time.
5. Now add that file to your email and send along with any notes you believe to be import in helping us see the same problem you see produced there. This means any files that are generated where the error appears will be helpful in keeping the time to resolution short.

Caution:

Files exported with the same name as those that are normally provided with a Trading Blox installation will be overwritten when the Suite is imported in another installation. This concern is for users who might have made modifications to some of the Blox or System module files and didn't change the name of file in some way that it is not longer the default name.

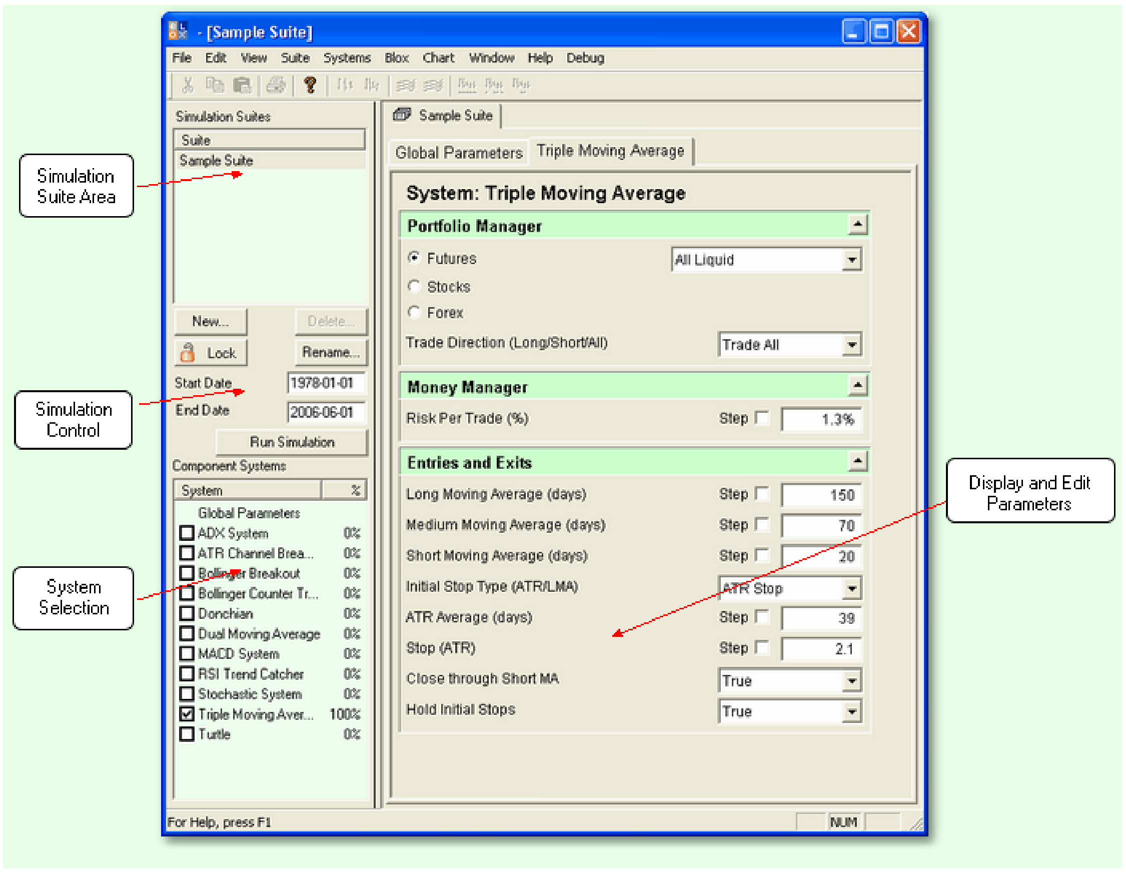
Special files that are not normally part of the Trading Blox process, like special index or weighting files that are not contained with a Blox module will not be exported and then will not be available for export.

Part II

Part 2 – Using Trading Blox -- an overview

Trading Blox uses sophisticated graphical elements to allow users to test without programming. Depending on which version of the software you purchased, some options may be different for your program.

The Trading Blox interface has four main sections:



Start Date 1978-01-01
End Date 2006-06-01

System	%
Global Parameters	
☐ ADX System	0%
☐ ATR Channel Brea...	0%
☐ Bollinger Breakout	0%
☐ Bollinger Counter Tr...	0%
☐ Donchian	0%
☐ Dual Moving Average	0%
☐ MACD System	0%
☐ RSI Trend Catcher	0%
☐ Stochastic System	0%
☑ Triple Moving Aver...	100%
☐ Turtle	0%

System: Triple Moving Average

Portfolio Manager

[✓ Futures] Stocks Forex — All Liquid

Trade Direction (Long/Short/All): Trade All

Money Manager

Risk Per Trade (%) Step ☐ 1.3%

Entries and Exits

Long Moving Average (days)	Step ☐	150
Medium Moving Average (days)	Step ☐	70
Short Moving Average (days)	Step ☐	20
Initial Stop Type (ATR/LMA)		ATR Stop
ATR Average (days)	Step ☐	39
Stop (ATR)	Step ☐	2.1
Close through Short MA		True
Hold Initial Stops		True

Suite List Area
The *Suite List Area* shows the current Simulation Suites. Each Simulation Suite represents a group of settings for a complete simulation. The settings include which systems are part of the simulation, the parameters for each system, the global parameters and assumptions like slippage, commissions, etc. Trading Blox's Simulation Suites let you work on different trading ideas at the same time.

Simulation Control Area
The *Simulation Control Area* controls the simulation dates and initiates historical simulations. Pressing the *Run Simulation* button, choosing the Suite | Run Historical Simulation menu or pressing the F5 key will start a test.

System List Area
The *System List Area* determines which systems are *Active* for testing and editing and lets you select active systems for editing. Clicking in the check box for a system will activate or deactivate that

system. Clicking on an *Active* system will bring that system's parameter editor to the top of the *Edit and Display Area*.

Edit and Display Area
The *Edit and Display Area* displays both system-level and test-level parameter editors. It also displays test results and trades after historical simulations complete. The contents of the *Edit and Display Area* change as the user selects various systems for editing or windows using the tabs at the top of the *Edit and Display Area.*

Section 1 – Working with Systems

You can use the systems that come with Trading Blox, you can get systems from other people and import them, or if you have the Pro or Builder versions of Trading Blox, you can create your own. This section of the manual describes how to create your own systems by assembling Blox. You should also be familiar with this section if you want to modify an included system, or a system you have purchased from another source.

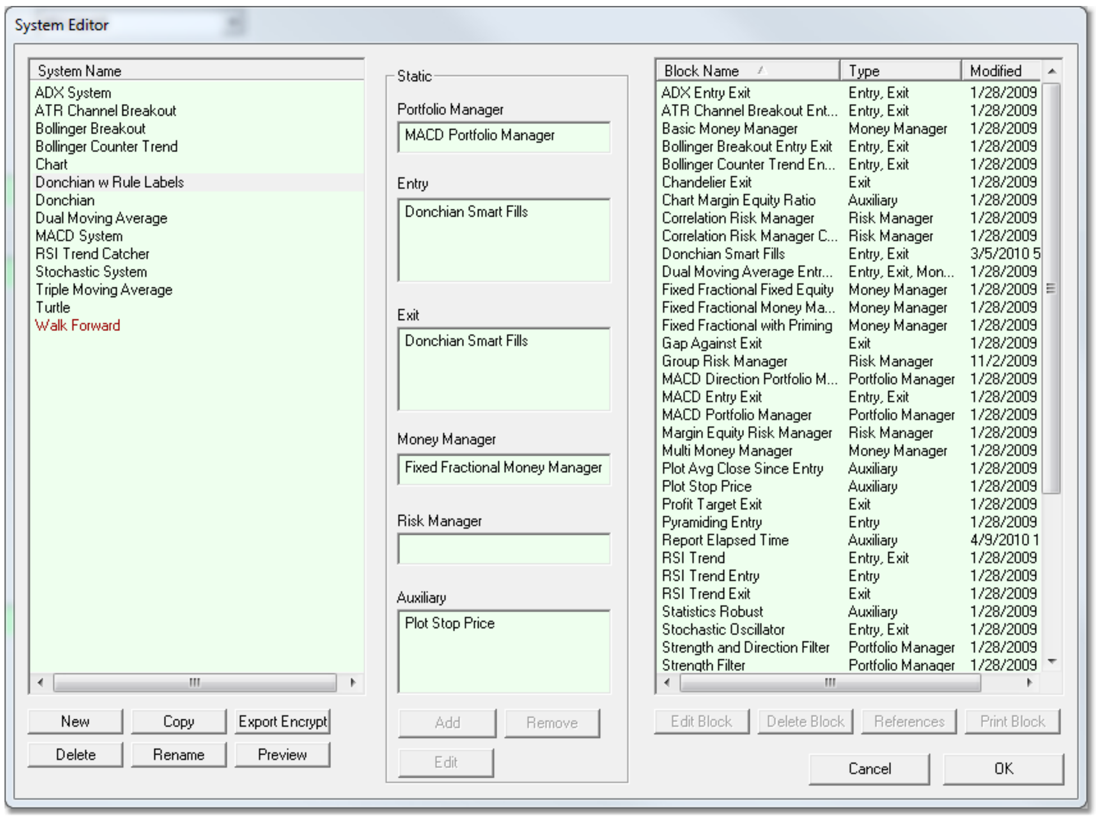
What is a system?

Systems are a collection of Blox. There are five Block types that we use, and each has a particular purpose in a trading methodology. As described earlier, the basic components of a trading system are:

System Component	Corresponding Block Type
What to Trade	Portfolio Manager
When to Trade	Entry Blox
Whether to Trade	Risk Manager
How Much to Trade	Money Manager
When to get out	Exit Blox

The System Editor

Selecting "Edit Systems" from the System Menu will bring up the System Editor window:


System Editor

System Name
ADX System
ATR Channel Breakout
Bollinger Breakout
Bollinger Counter Trend
Chart
Donchian w Rule Labels
Donchian
Dual Moving Average
MACD System
RSI Trend Catcher
Stochastic System
Triple Moving Average
Turtle
Walk Forward

Static
Portfolio Manager: MACD Portfolio Manager
Entry: Donchian Smart Fills
Exit: Donchian Smart Fills
Money Manager: Fixed Fractional Money Manager
Risk Manager:
Auxiliary: Plot Stop Price

Block Name	Type	Modified
ADX Entry Exit	Entry, Exit	1/28/2009
ATR Channel Breakout Ent...	Entry, Exit	1/28/2009
Basic Money Manager	Money Manager	1/28/2009
Bollinger Breakout Entry Exit	Entry, Exit	1/28/2009
Bollinger Counter Trend En...	Entry, Exit	1/28/2009
Chandelier Exit	Exit	1/28/2009
Chart Margin Equity Ratio	Auxiliary	1/28/2009
Correlation Risk Manager	Risk Manager	1/28/2009
Correlation Risk Manager C...	Risk Manager	1/28/2009
Donchian Smart Fills	Entry, Exit	3/5/2010 5
Dual Moving Average Entr...	Entry, Exit, Mon...	1/28/2009
Fixed Fractional Fixed Equity	Money Manager	1/28/2009
Fixed Fractional Money Ma...	Money Manager	1/28/2009
Fixed Fractional with Priming	Money Manager	1/28/2009
Gap Against Exit	Exit	1/28/2009
Group Risk Manager	Risk Manager	11/2/2009
MACD Direction Portfolio M...	Portfolio Manager	1/28/2009
MACD Entry Exit	Entry, Exit	1/28/2009
MACD Portfolio Manager	Portfolio Manager	1/28/2009
Margin Equity Risk Manager	Risk Manager	1/28/2009
Multi Money Manager	Money Manager	1/28/2009
Plot Avg Close Since Entry	Auxiliary	1/28/2009
Plot Stop Price	Auxiliary	1/28/2009
Profit Target Exit	Exit	1/28/2009
Pyramiding Entry	Entry	1/28/2009
Report Elapsed Time	Auxiliary	4/9/2010 1
RSI Trend	Entry, Exit	1/28/2009
RSI Trend Entry	Entry	1/28/2009
RSI Trend Exit	Exit	1/28/2009
Statistics Robust	Auxiliary	1/28/2009
Stochastic Oscillator	Entry, Exit	1/28/2009
Strength and Direction Filter	Portfolio Manager	1/28/2009
Strength Filter	Portfolio Manager	1/28/2009

New System
You can create new systems and delete systems using the system editor.

You build systems by selecting a system on the left, and adding Blox from the available blox list on the right.

Some of the lists accept multiple Blox and some lists can accept only one Block. The Portfolio Manager, the Risk Manager, and the Money Manager can accept only one Block per system. The Entry Block and Exit Block can accept multiple Blox per system. The reason for this is that you may want to have multiple entry/exit ideas executing at the same time.

Copy System
Select the system to copy, press the copy system button, and enter a new name.

Rename System
Select the system to rename, press the rename system button, and enter a new name.

Preview
Prepares a print preview of the system, including all the blox, scripts, variables, indicators, etc.

Delete Systems
Caution: If you delete a systems, you cannot recover the system other than to recreate it. The good news is that since a system is only a collection of Blox, recreating the system is as simple as creating a new system and dragging the required Blox into the system. It is still good practice to back up your systems on a regular basis. The systems are stored in a folder called "Systems" in your Trading Blox folder.

If a System contains a Block of type that can only have one Block, you must remove the Block before adding a new one. A system can have multiple Entry and Exit Blocks.

To delete a Block from the system, select the System, select the Block within the System, and click "Remove Block from System" You will only be deleting this block from the selected system. You will not be deleting this block from your list of available blocks.

The Blox required for a system to be able to trade are Entry Signals, Exit Signals, and Money Management. The Entry and Exit Block need to call the Broker object to enter and exit trades, and the Money Manager Block needs to set the trade quantity for each trade. You can use the Basic Money Manager to get started quickly.

After you have modified a System (changed, added, or deleted the Blox contained in the System) click the OK button to save and exit, or click Cancel to cancel all changes. To edit a Block or view the code, double click on the block name. This will bring you directly to the editor with that Block selected.

Export and Encrypt System
This button will export the system and attached blox to a special encrypted file. It will be put in the Export folder, which will be displayed. You can then send this file to another Trading Blox user. They will be able to use and test with the system, change parameters, etc, but will not be able to view or edit the system or blox.

To use one of these exported systems, put the .tbz file in your Import folder before starting up Trading Blox.

Note: Be sure that the name of the System and the name of all Blox in the system are unique. We recommend that you use your name, or some other unique identifier, in the blox and system names. In this way, they will not conflict with other blox or systems that may already be in an environment prior to importing the encrypted system.

Do not change the name of the exported .tbz file once it is created. It is linked to the system name and blox and changing the name will cause the system to not load. When testing an export/import process into the same environment, make a renamed copy of the system and then export, delete the copy, and import. In this way the system name is not duplicated in the same environment.

Import System
If someone sends you an encrypted system, a .tbz file, you place that in your Import folder. When you startup Trading Blox, this system will be listed and available for testing, but you will not be able to view or edit the system or the blox. Note that system names in Trading Blox must be unique, so if this imported system has the same name as an existing system, it will not load. Delete or rename the existing system and restart Trading Blox.

Add
Use this button, or a right click, to add a block to a system.

Remove
Use this button, or right click, to remove a block from a system.

Edit
Use this button to edit a block in the Blox Editor.

Edit Block
Select a block, and click on this button to edit the block in the Blox Editor. Same function as the Edit button in the middle. Available only in the Blox Builder Edition.

Delete Block
Select a block, and click on this button to delete the block from the system and delete the file as well. Blox cannot be delete if they are in a system. Use with caution as blox cannot be recovered once deleted.

References
Select a block, and click on this button to see a list of system references. All the systems the block is in will be listed.

Print Block
Prints the block to the printer. Includes all the scripts, variable, indicators, and parameters in the block.

Global Suite Systems
If the system name is the same as a suite, it will be a global suite system. This system allows blox to be attached directly to the suite, and have access to data from all systems. The global suite system scripts run after all the system scripts of same name run.

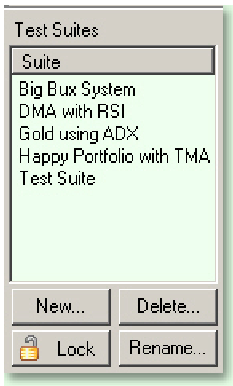
Scripts available for use in a global suite system:
Before Simulation, Before Test, After Trading Day (access to final test equity), After Test, After Simulation. These scripts have no system object context.
Entry Order Filled, Exit Order Filled, Can Add Unit, Can Fill Order -- note that these order scripts run for all orders placed or filled regardless of originating block or system. The scripts also have access to the system, instrument, and order object from the block in which the order was placed or filled.

Section 2 – Simulation Suites

A Simulation Suite embodies a complete group of settings for a simulation. The settings include which systems are part of the simulation, the parameters for each system, and global parameters like slippage, commissions, etc.

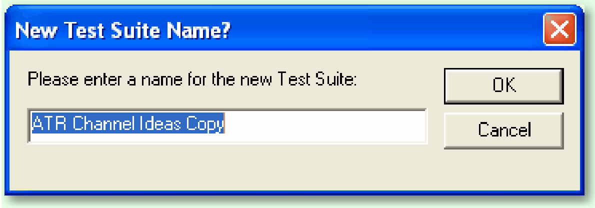
Trading Blox's Simulation Suites let you work on different trading ideas at the same time. For example, you might have one suite that contains the settings for the systems you are currently trading, and several others that contain settings for ideas and ongoing research.

The current Simulation Suite is specified in the Suite List Area:



Selecting another Simulation Suite will replace the contents of the System List Area and Edit Area with the editors corresponding to that Simulation Suite.

You can create a new Simulation Suite by selecting a Simulation Suite and then pressing the *New...* button or selecting *New Suite...* from the Suite menu. This will copy the parameters and settings from the currently selected Simulation Suite and bring up a dialog for entering the name:



Please enter a name for the new Test Suite:

ATR Channel Ideas Copy

The dialog allows you to create a new suite with a different name. Selecting *OK* will create a new suite with the name you specify. This new Suite will contain the same settings as the Simulation Suite that was selected when the New... button was pressed.

You can delete Simulation Suites by selecting one and then pressing the *Delete* button or selecting *Delete Suite...* from the Suite menu.

You can rename an existing Simulation Suite by selecting it and pressing the *Rename...* button or selecting *Rename Suite...* from the Suite menu.

Locking Suites
Simulation Suites can be locked which will disable all editing controls for the system parameters and settings. This will help prevent inadvertent changes to a system.

To lock a Simulation Suite, simply press the *Lock* button.

or select *Lock Suite* from the Suite menu.

When a Simulation Suite has been locked, the button changes to an *Unlock* button which can be used to unlock the Simulation Suite. Similarly, the *Lock Suite* menu item changed to *Unlock Suite* when the current suite is a locked Simulation Suite. Note that this does not prevent changes to the system or the blox, so to truly lock a suite a second copy of the entire Trading Blox folder should be used. An example would be for production order generation where changes could be disastrous.

The **Walk Forward** Suite can be used to walk forward test any system.

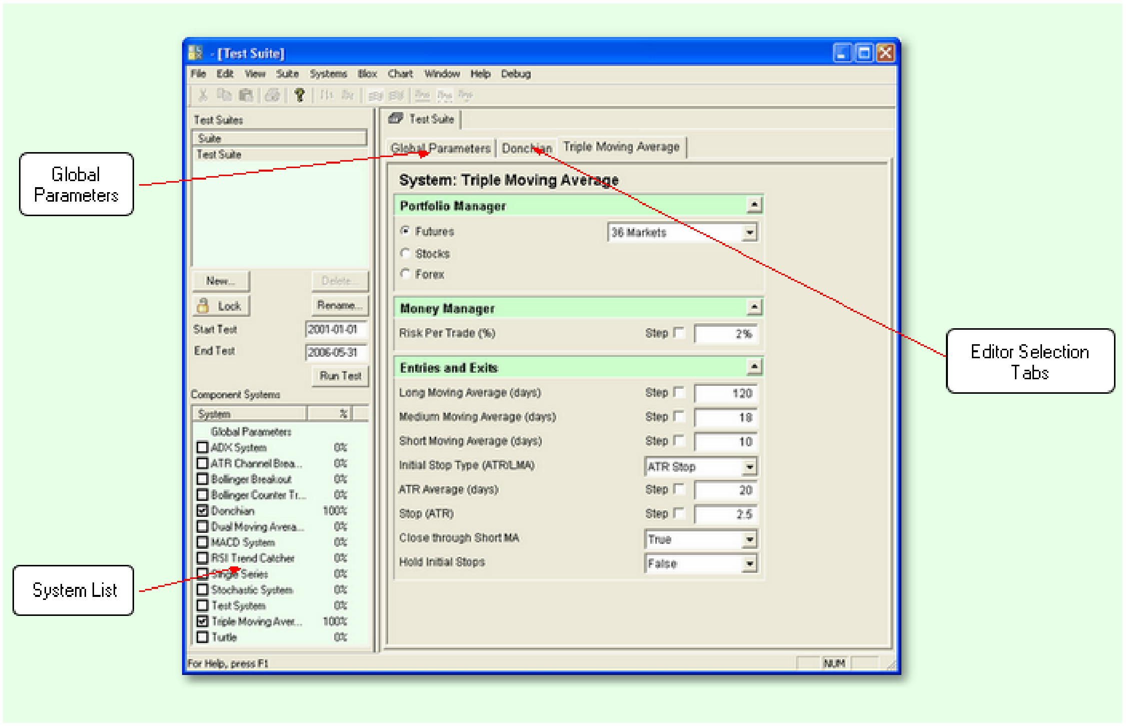
This process uses the measure of goodness as defined in preferences, so the measure can be a custom statistic combination of many goodness measures.

1) Set the Run (Index) parameter to step from 1 to 5, step 1. (Or more as needed)
2) Set the Optimization Run parameter to step true to false. (Required)
3) Using this suite, select your chosen system and set your optimization parameters to step as desired. Can step just about any parameters in your system, except dates.
4) Set your start and end dates to make sure you have enough data loaded to complete the walk forward test, given the parameters. Only data loaded between the start and end dates can be used for the test. A Jan 1 start date works well, since the end date for the yearly option is hard coded to Dec 31.
5) This will likely be considered a large simulation run, so turn on all reporting for the large simulation run, and turn off the multi parameter graphs.

Further discussion on the process and options can be found here:
http://www.tradingblox.com/forum/viewtopic.php?t=7323

Or visit the walk forward section of the manual.

Section 3 – Simulation Parameters Editor



Start Test 2001-01-01
End Test 2006-05-31

Component Systems

System	%
Global Parameters	
☐ ADX System	0%
☐ ATR Channel Brea...	0%
☐ Bollinger Breakout	0%
☐ Bollinger Counter Tr...	0%
☑ Donchian	100%
☐ Dual Moving Avera...	0%
☐ MACD System	0%
☐ RSI Trend Catcher	0%
☐ Single Series	0%
☐ Stochastic System	0%
☐ Test System	0%
☑ Triple Moving Aver...	100%
☐ Turtle	0%

System: Triple Moving Average

Portfolio Manager

(● Futures) ○ Stocks ○ Forex — 36 Markets

Money Manager

Risk Per Trade (%)	Step ☐	2%

Entries and Exits

Long Moving Average (days)	Step ☐	120
Medium Moving Average (days)	Step ☐	18
Short Moving Average (days)	Step ☐	10
Initial Stop Type (ATR/LMA)		ATR Stop
ATR Average (days)	Step ☐	20
Stop (ATR)	Step ☐	2.5
Close through Short MA		True
Hold Initial Stops		False

Trading Blox showing the Triple Moving Average System parameters.

The Trading Blox application includes separate System Parameter Editors for each of the systems where users can change test values like the number of days in a moving average, the trading portfolio, stop size, etc.

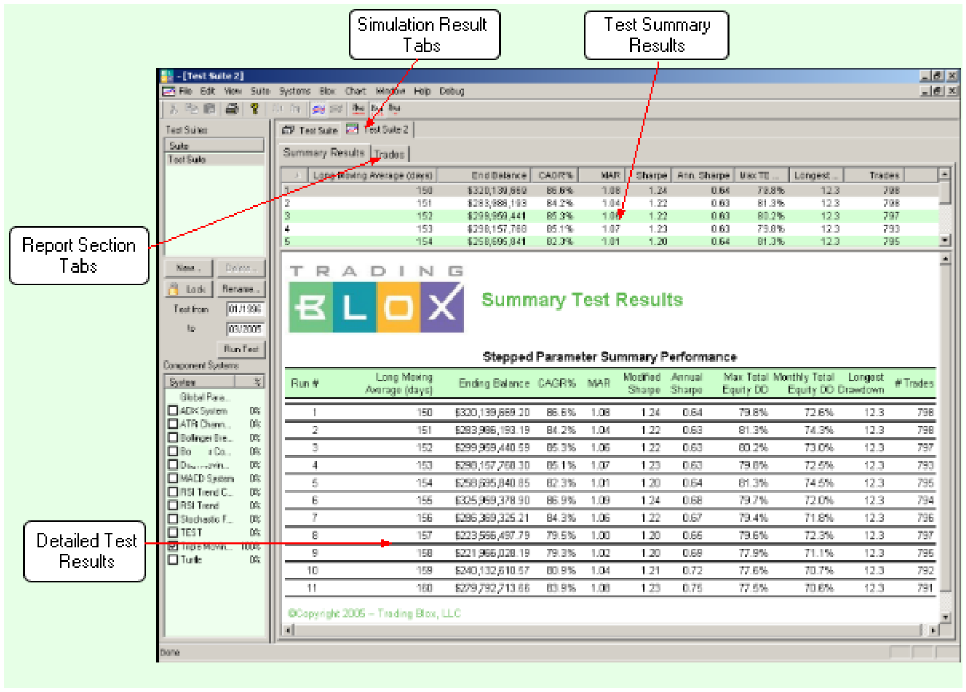
Since the rules for each system are different, the associated parameter values for each system are likewise different. So each system has its own separate parameter editor for editing those system specific values. For information on the parameter values specific to each system, see the Built-in Systems section.

System List
The left side of the screen contains a list of the systems contained in Trading Blox. Users can activate particular systems for testing by clicking on the check box next to the system in the system list. Each active system will have a tab next to the "Global Parameters" tab.

Editor Selection Tabs
Active systems have an associated editor tab. Clicking a tab brings up the System Parameter Editor for that system.

Global Parameter Editor
Trading Blox allows users to edit global parameters like starting balances, slippage and commission amounts, etc. using the Global Parameter Editor. For information on individual global parameters, see the Global Parameter Reference section.

Section 4 – Simulation Results



	Long Moving Average (days)	End Balance	CAGR%	MAR	Sharpe	Ann. Sharpe	Max TE ...	Longest ...	Trades
1	150	$320,139,669	86.6%	1.08	1.24	0.64	79.8%	12.3	798
2	151	$283,986,193	84.2%	1.04	1.22	0.63	81.3%	12.3	798
3	152	$299,959,441	85.3%	1.06	1.22	0.63	80.2%	12.3	797
4	153	$298,157,768	85.1%	1.07	1.23	0.63	79.8%	12.3	793
5	154	$258,695,841	82.3%	1.01	1.20	0.64	81.3%	12.3	795

Summary Test Results

Stepped Parameter Summary Performance

Run #	Long Moving Average (days)	Ending Balance	CAGR%	MAR	Modified Sharpe	Annual Sharpe	Max Total Equity DD	Monthly Total Equity DD	Longest Drawdown	# Trades
1	150	$320,139,669.20	86.6%	1.08	1.24	0.64	79.8%	72.6%	12.3	798
2	151	$283,986,193.19	84.2%	1.04	1.22	0.63	81.3%	74.3%	12.3	798
3	152	$299,959,440.59	85.3%	1.06	1.22	0.63	80.2%	73.0%	12.3	797
4	153	$298,157,768.30	85.1%	1.07	1.23	0.63	79.8%	72.5%	12.3	793
5	154	$258,695,840.85	82.3%	1.01	1.20	0.64	81.3%	74.5%	12.3	795
6	155	$325,969,378.90	86.9%	1.09	1.24	0.68	79.7%	72.0%	12.3	794
7	156	$286,369,325.21	84.3%	1.06	1.22	0.67	79.4%	71.8%	12.3	796
8	157	$223,566,497.79	79.5%	1.00	1.20	0.66	79.6%	72.3%	12.3	797
9	158	$221,966,028.19	79.3%	1.02	1.20	0.69	77.9%	71.1%	12.3	795
10	159	$240,132,610.57	80.9%	1.04	1.21	0.72	77.6%	70.7%	12.3	792
11	160	$279,792,713.66	83.9%	1.08	1.23	0.75	77.5%	70.6%	12.3	791

©Copyright 2005 – Trading Blox, LLC

The *Test Results* are displayed when a historical simulation test is finished. Each *Test Results* window has the following areas:

Results Tabs
Each *Simulation Results* window has an associated test results tab. Trading Blox automatically selects the tab which corresponds to the most recent test after that test finishes. The tabs are named after the Simulation Suite

Test Summary List
The top of the *Summary Results* tab contains a list of the results for each of the distinct parameter combinations run for the simulation. For simulations with no parameter steps, this list will have only a single line. Clicking on any of the lines in the *Test Summary List* takes you to the top of the detailed results for that test.

Detailed Test Report
The bottom of the *Summary Results* tab contains a detailed test report. This report is generated using the HTML format, the same format as internet web pages. This allows much more sophisticated formatting than the simple ASCII Text file output used by most system testing programs. Each of these HTML-based reports are also saved as files in the *Results Folder* for later viewing. You can define which reports show up in the Detailed Test Report section using the Report Content Specification section of the Trading Blox Preferences.

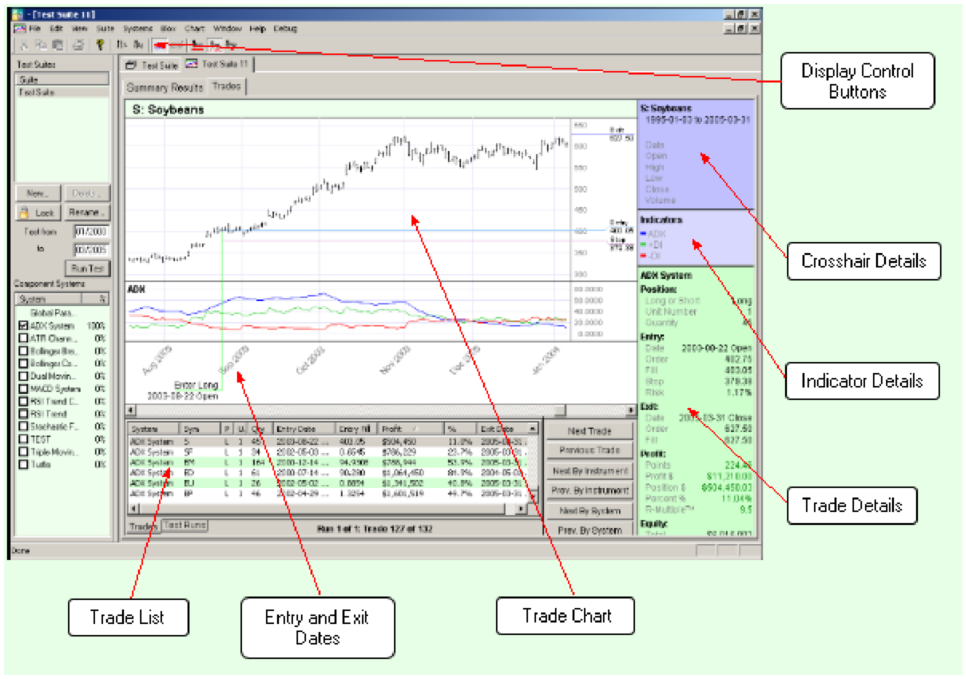
Note that this HTML version of the summary results is redundant to the Test Summary Listing above. You can right click on this list to export directly to excel. This report does not print for runs greater than 5000 because the resulting file would be too large for windows to handle.

Report Section Tabs
Each *Test Results* window has one tab for the *Summary Results* and one for the *Trades*. Clicking on

a tab will take you to that section.. The next section, Test Trades Report, shows the *Trades* Tab.

Section 5 – Test Trades Report

The *Test Trades Report* shows each of the trades for a given *Test*, when you have "Log Trades" turned on in Test Reporting preferences. If you have this unchecked, then you will see no "Trades" tab. Each *Test Trades Report* has the following areas:



Display Control Buttons

The tool bar has buttons which allow single-click access to common menu actions for controlling the chart display including: expanding and collapsing the width of a price bar; showing and hiding the trade indicators; and controlling how many trades are displayed at the same time.

Trade Chart

The selected trade is plotted on a daily bar chart graph of the prices for the particular market or stock issue. The indicators for the system that generated the trade are also plotted on the graph so you can see how the system indicators control trade entry and exit. The Chart menu and associated buttons on the tool bar control which trades are displayed on the graph.

Entry and Exit Dates

The entry and exit dates for the selected trade are shown below the chart date legend. These show the exact date and time of entry (Open, Day, or Close).

Trade List

A complete list of each trade for the test is shown below the chart. When you click on a trade in the list, Trading Blox will select that trade and display the appropriate chart. You can sort the list by clicking on any of the column headers. For example, you can easily determine the largest percentage losing trades by clicking on the Profit % column.

An Exit Rule is added to positions that are forced to exit by the application:

- **Broker Position Exit** is caused by a new broker position that must exit the exiting open position, or a broker position of OUT.
- **End of Data** is an exit on the last bar of data when Running a Simulation. These positions are not

exited when Generating Orders.

- **Test End Exit** is an exit at the end date of the test (based on the test start and end dates), when the last bar of data is not reached.

Crosshair Details
When you move the mouse over the chart area, Trading Blox draws a crosshair on the chart and draws the details of the bar under the current mouse position in the Crosshair Details pane. It also displays the price associated with the current vertical mouse position. These values update immediately as you move the mouse across the chart. You can take manual control of the cross hairs by double clicking on the chart. The arrows will then be enabled for moving the cross hairs. To resume the automatic mouse control of the cross hairs just double click in the chart area again.

Indicator Details
The Indicator Details pane shows the values for the system indicators as of the close one day prior to the day associated with the bar under the current mouse position. The indicators let you graphically see the reasons a particular trade was initiated, e.g. a moving average crossover, a breakout, etc.

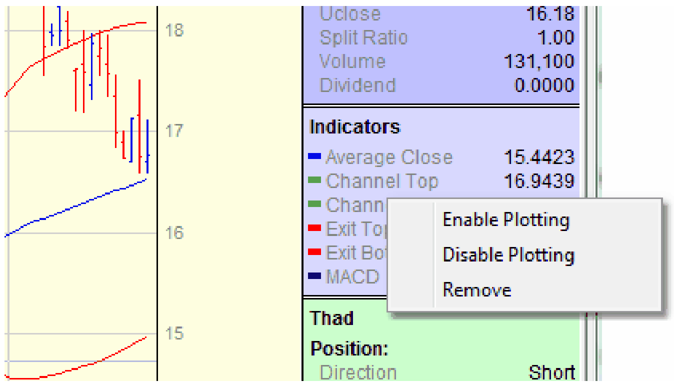
The trade chart has a pop up menu if you **right click** on any indicator or plotting series.

If the series is 'display' but not 'plotting' (as defined in the Blox Editor) you can **Enable Plotting** for this chart.

Likewise you can **Disable Plotting,** or

Remove the indicator from the list completely. Once removed, you cannot then enable or disable plotting.

These actions are for the current chart only, and do not affect other charts, or future charts created by a new simulation.



```
Uclose            16.18
Split Ratio        1.00
Volume          131,100
Dividend         0.0000
Indicators
Average Close   15.4423
Channel Top     16.9439
Chann
Exit To
Exit Bo
MACD
Thad
Position:
Direction         Short
```

Entry and Exit Prices
The trade Entry, Exit, and initial Stop are plotted so you can visually see the exact trade prices.

Trade Details
The bottom right pane shows the details of each trade including: entry and exit dates and prices, profit, profit %, entry risk %, trade duration, etc.

"En Trading" is the Entry Trading Equity (system.tradingEquity) for the day prior to the trade entry date. So this is the equity used for trade size computation if using system.tradingEquity, as the Fixed Fractional Money Manager does.

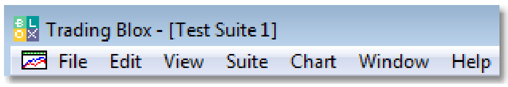
"Ex Closed" and "Ex Total" is the test closed and total equity on the exit date of the trade.

Pressing Z will zoom the chart and R will reduce the chart, once a trade has been selected.
Pressing Shift while scrolling the chart will freeze the scale.

If the screen is too small to display all the information at once, the cross hair and indicator info will be removed when a trade is selected, so that the green trade information section has the full display. When the cross hairs are again engaged, the cross hair and indicator information will again present on top of the trade information.

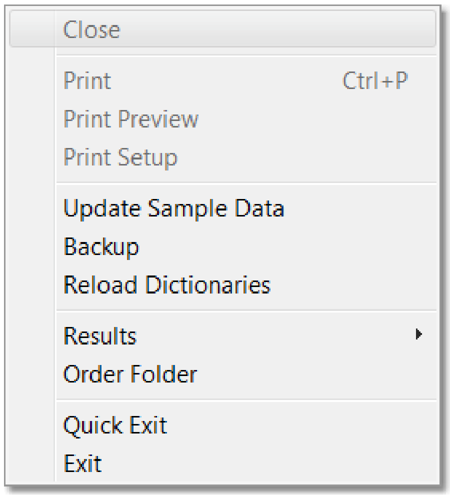
Section 6 – Menu Reference

The Trading Blox Menu Bar contains the following menus. Depending on which version of the software you own, some of these menus may be absent.



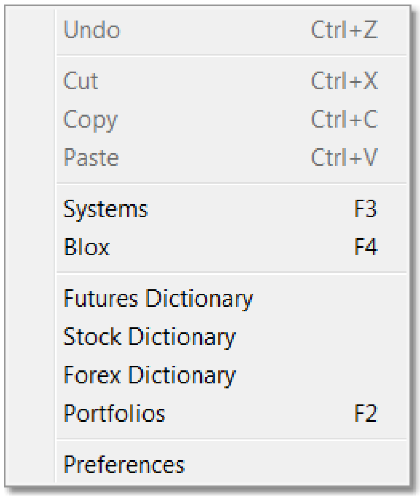
Each of these menus is explained in detail below.

File Menu



Close	Closes the current window.
Print...	Prints the Summary Results. Printing of charts is not currently possible in Trading Blox.
Print Preview	Previews the Printing for Summary Results.
Print Setup...	Sets up the printer settings.
Update Sample Data	Updates the sample Futures and Forex data. The data is downloaded from the internet and installed in the Trading Blox / Data folder.
Backup	Creates a back up of certain files using the backup.bat batch file. Backups are zipped and date/time stamped and located in the Backups folder.
Reload Dictionaries	In the case a dictionary file is changed external to Trading Blox, this menu will reload all dictionary files and exchanges and currency files.
Results	Contains sub menus for all the results files and the results folder.
Order Folder	Opens the folder in which the order file has been generated and stored.
Exit	Exits the Trading Blox application.

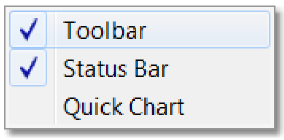
Edit Menu



Undo	Undoes the last editing command.

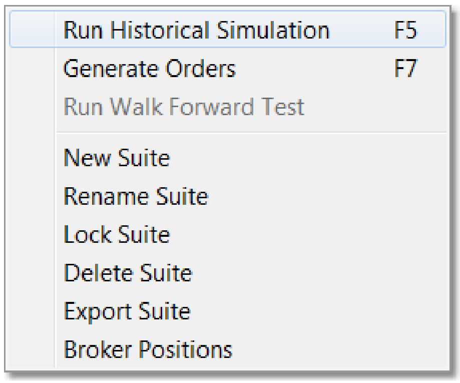
Cut	Removes the selected text and copies it to the clipboard.
Copy	Copies the selected text to the clipboard.
Paste	Pastes the contents of the clipboard to the current cursor location.
Systems	Opens the System Editor.
Blox	Opens the Blox Editor.
Futures Dictionary	Opens the Futures Dictionary Editor.
Stock Dictionary	Opens the stock dictionary in notepad or excel. The stock dictionary can be located in the Stock Data Folder, or Dictionaries folder, and can be called stockinfo.txt or stockinfo.csv. There is no GUI editor for this dictionary.
Forex Dictionary	Opens the Forex Dictionary Editor.
Portfolios	Opens the Portfolio Editor.
Preferences	Opens the Preferences Editor.

View Menu



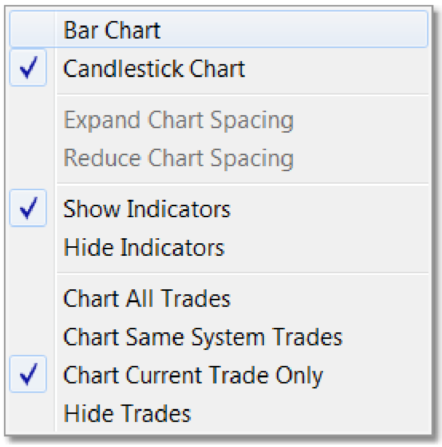
Toolbar	Shows or hides the Toolbar.
Status Bar	Shows or hides the Status Bar (bottom of main window).
Quick Chart	Creates charts of any markets. Multiple charts can be open at any one time. Only available in the Builder Edition.

Suite Menu



Run Historical Simulation	Runs a simulation with the current suite, systems, and settings.
Generate Orders	Generates orders with the current suite.
Run Walk Forward	Run a Walk Forward test on the current system and settings. Optimization size and Out of Sample size are set in global parameters as calendar days.
New Suite	Creates a new Simulation Suite.
Rename Suite	Renames the currently selected Simulation Suite.
Lock Suite	Locks the current Simulation Suite. Text changes to Unlock Suite if the current Simulation Suite is already locked.
Delete Suite	Deletes the current Simulation Suite
Export Suite	Exports the suite, systems, blox, and portfolio files into folders in the Export folder. In this way a suite can be zipped and sent to a friend or moved to another computer.
Broker Positions	Opens the broker positions editor

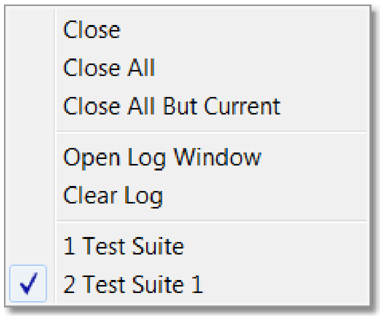
Chart Menu



Bar Chart	Sets the chart type to Bar Chart.
Candlestick Chart	Sets the chart type to Candlestick Chart.
Expand Chart Spacing	Expands the spacing between bars or candles. Also press Z for Zoom.

Reduce Chart Spacing	Reduces the spacing between bars or candles. Also press R for Reduce.
Show Indicators	Shows the indicators for systems on the Trade Chart.
Hide Indicators	Hides the indicators for systems.
Chart All Trades	Charts every trade for a given instrument even if they are from different systems.
Chart Same System Trades	Charts only trades from the same system as the Current Trade.
Chart Current Trade Only	Charts only the current trade which is the trade selected in the Trade List below the Trade Chart itself.
Hide Trades	Do not show the trades on the chart.

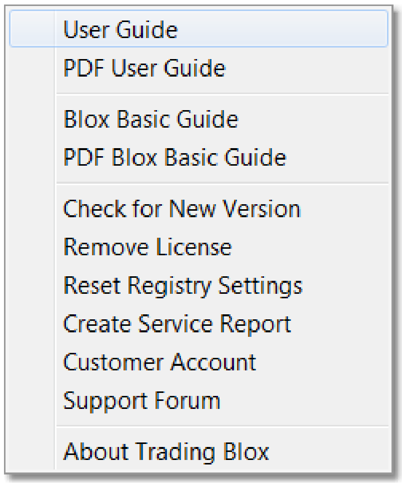
Window Menu



Close	Closes the current Results Window.
Close All	Closes all the open results windows, and reset the data.
Close All But Current	Closes all the windows but the currently selected window.
Open Log Window	Opens the log window on the bottom of the screen so messages PRINTED in blox scripting will show.
Clear Log	Clears the log window.
Test Suite	The suite window -- the system tab with the global parameters and system parameters tab.
Test Suite 1	This is the first test run for the suite Test Suite.

The Window menu changes to reflect all of the currently open results windows. They are named exactly as they appear in the Results Tabs.

Help Menu

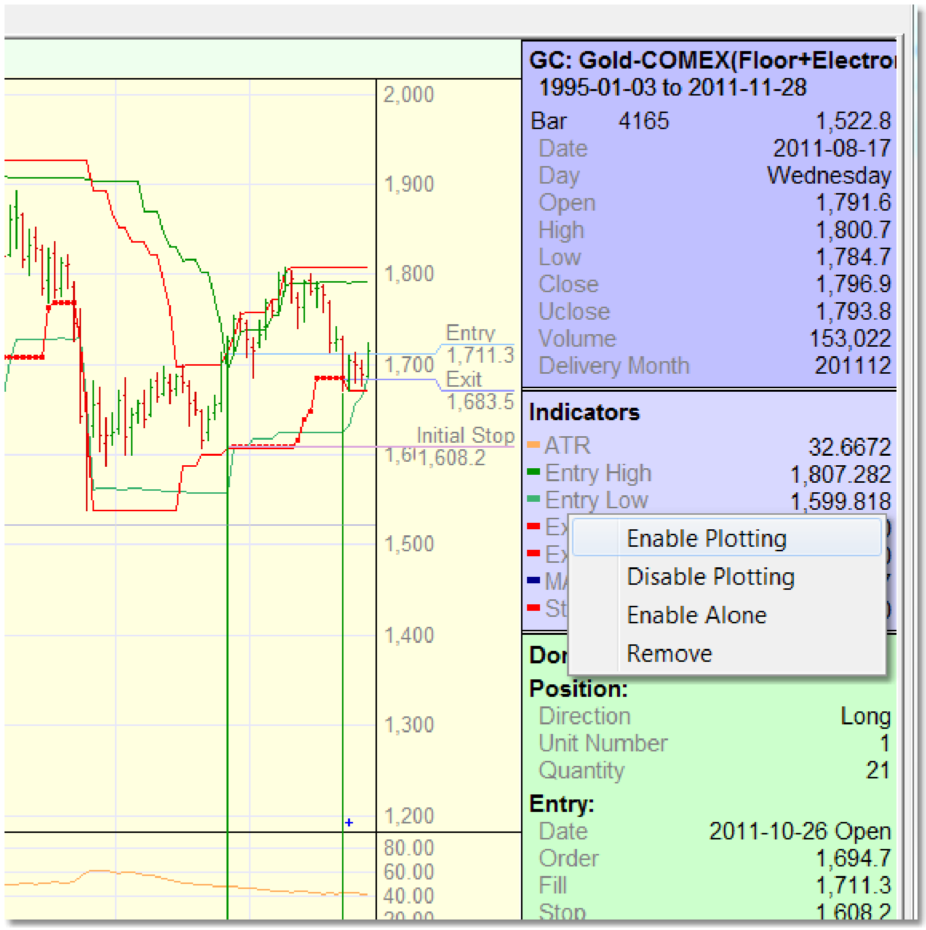


User Guide	Opens the Trading Blox User Guide
PDF User Guide	Opens the PDF version of the User Guide
Blox Basic Guide	Opens the Blox Basic Builder Guide
PDF Blox Basic Guide	Opens the PDF version of the Blox Basic Guide
Check for New Version	Checks for the current version of Trading Blox
Remove License	Removes the license from this machine. You can then install on a new or different machine.
Reset Registry Settings	This will clear and reset all the preferences and settings that are stored in the registry. This is like a full uninstall and reinstall, but without removing the application. If you have missing or invisible windows, this might be an option to try.
Create Service Report	Creates a service report. You can then send the file "TradingBloxServiceReport.zip" to customer support if required.
Customer Account	Access the Customer Login section of the website
Support Forum	Launches the Support Forum on the Internet
About Trading Blox	Shows About Window

The trade chart has a pop up menu if you right click on any indicator or plotting series.

If the series is 'display' but not 'plotting' (as defined in the Blox Editor) you can set to plotting for this chart. Likewise you can disable plotting, or remove the indicator from the list completely. Once removed, you cannot then enable or disable plotting. These actions are for the current chart only, and do not affect other charts, or future charts created by a new simulation.

Selecting the Enable Alone menu item will cause only the selected indicator to plot in that graph area, and the others in the same graph area to disable plotting.



GC: Gold-COMEX(Floor+Electro
1995-01-03 to 2011-11-28

Bar	4165	1,522.8
Date		2011-08-17
Day		Wednesday
Open		1,791.6
High		1,800.7
Low		1,784.7
Close		1,796.9
Uclose		1,793.8
Volume		153,022
Delivery Month		201112

Indicators

ATR	32.6672
Entry High	1,807.282
Entry Low	1,599.818

Position:

Direction	Long
Unit Number	1
Quantity	21

Entry:

Date	2011-10-26 Open
Order	1,694.7
Fill	1,711.3
Stop	1,608.2

Historical Data Setup

Part III

Part 3 – Historical Data Setup

Formats

Trading Blox supports MetaStock and ASCII text formats. The sample data is provided in ASCII since it is human readable and easier to debug. Also, the MetaStock format does not include the Unadjusted Close or the Delivery Month for futures data, nor can you add extra fields to be imported and used.

There are comprehensive tutorials on data setup.
CSI Data: http://www.tradingblox.com/tradingblox/howto-ua.htm
Worden Data: http://www.tradingblox.com/tradingblox/howto-worden.htm

File Names

Metastock
With the Metastock format you don't need to deal with file names. If Trading Blox can read the Master file, it will locate all the file in the directory. For futures, be sure that the symbol is correct in the Futures Dictionary.

ASCII Text
Futures and Forex: The file name is entered in the Futures or Forex Dictionary. Be sure the file name entered here matches the file name in the selected folder. If the file name starts with the symbol, you can try to find the file using the Futures Dictionary "Find Symbol" button.

Stocks: The stock files must be named SYMBOL.TXT or SYMBOL.CSV where SYMBOL is the symbol of the stock. For example, a file name for AAPL needs to be AAPL.CSV or AAPL.TXT and the file name for MSFT needs to be MSFT.TXT or MSFT.CSV.

ASCII Data Files

For ASCII format, the files must be comma delimited. Also be sure the dates are in sequential order from earliest date to latest date.

Daily Data

The fields must be set up as follows, with no header.

Futures:
YYYYMMDD, Open, High, Low, Close, Volume, Open Interest, Delivery Month (YYYYMM), Unadjusted Close, Extra Data 1 ... 8

Stocks and Forex:
YYYYMMDD, Open, High, Low, Close, Volume, Unadjusted Close, Extra Data 1 ... 8

Everything after DOHLC is optional. You can create a file that only contains DOHLC, or you can create a file that includes more. But if you want to include more, it must be in the right order. So if you want to include extra data 1 but you don't have the futures month, then leave an extra comma there.

If you are using CSI Unfair Advantage, the "ASCII Field Layout" for futures is DOHLCVINU. Also be sure the files are ASCII format, with .txt extension, and your data range is As Available to As Available.

Intraday Data

For intraday data, a **header must be used** with at least the first two items "Date,Time", such as:

Date,Time,Open,High,Low,Close,Volume

Thereafter the format of the data is:

Futures:
YYYYMMDD, HHMM, Open, High, Low, Close, Volume, Open Interest, Delivery Month (YYYYMM), Unadjusted Close, Extra Data 1 ... 8

Stocks and Forex:
YYYYMMDD, HHMM, Open, High, Low, Close, Volume, Unadjusted Close, Extra Data 1 ... 8

Trading Blox accepts the following optional date formats: YYYYMMDD, YYMMDD, MM/DD/YYYY, MM/DD/YY.
Trading Blox accepts the following optional time formats: HHMM, HH:MM.

The Dictionaries

In order for Trading Blox to read your data, there needs to be a dictionary file. This can be in the Dictionary folder, or in the same folder as the data. This file contains the market specifics for the instruments.

If you don't have a Stock Dictionary in your stock data folder, Trading Blox will look for a Master file. If found, it will assume Metastock format, and if not found it will assume Text format. If text format, it will load all the files ending in .txt into the system. Since it has no information about these stocks, the symbol will be used as the description. If you would like to create a stockinfo.txt file, click here.

If you don't have a Futures Dictionary in your futures data folder, Trading Blox will give you some warnings on startup, but allow you to create a new one. Open the Futures Dictionary once you start up, and start creating new markets for all the instruments you are using. If you have CSI UA, you can use the Synch UA button and the Find File button to make this process much easier. Or you can copy the futuresinfo.txt file from our sample data folder into you new data folder as a way to get started more quickly. If you do copy the file in, be sure to restart so Trading Blox can read the new file.

Stock Dictionary: StockInfo.txt
Futures Dictionary: FuturesInfo.txt
Forex Dictionary: ForexInfo.txt.

See Data and Folder Preferences to change the folder to your data's location.

Interest Rates Files

The rates files are located in the Forex Data Folder as set in preferences. They are required for forex testing when accounting for the cost of carry, and also the base currency rate is required when earning interest. So in the case of a USD base currency, the USD_Rates.txt file is used for earning interest as well as paying margin on stocks. The Lending Rate is used when earning interest, and the Borrowing Rate is used for paying margin.

The file can be sparsely populated, and for any given test date, the last available date in the file will be used for the rates. The format for this file is as follows. If only one rate is in the file, it is the Lending Rate, and the Borrowing Rate is the Lending Rate plus .5%. Note that the concepts of Borrowing and Lending are from the consumer's perspective.

Date, Lending Rate, Borrowing Rate

Example:
20081008,0.015,0.0205
20081029,0.010,0.0155
20081031,0.0045,0.0155
20081216,0.0015,0.0125

Exchange Definition File

The exchanges.csv file contains the abbreviation and description of all the exchanges that are used in Trading Blox. A warning will appear if the exchange is used by a market, but missing from this file. To add an exchange to this file simply open the file in excel or notepad, add the exchange and description, and save. Restart Trading Blox.

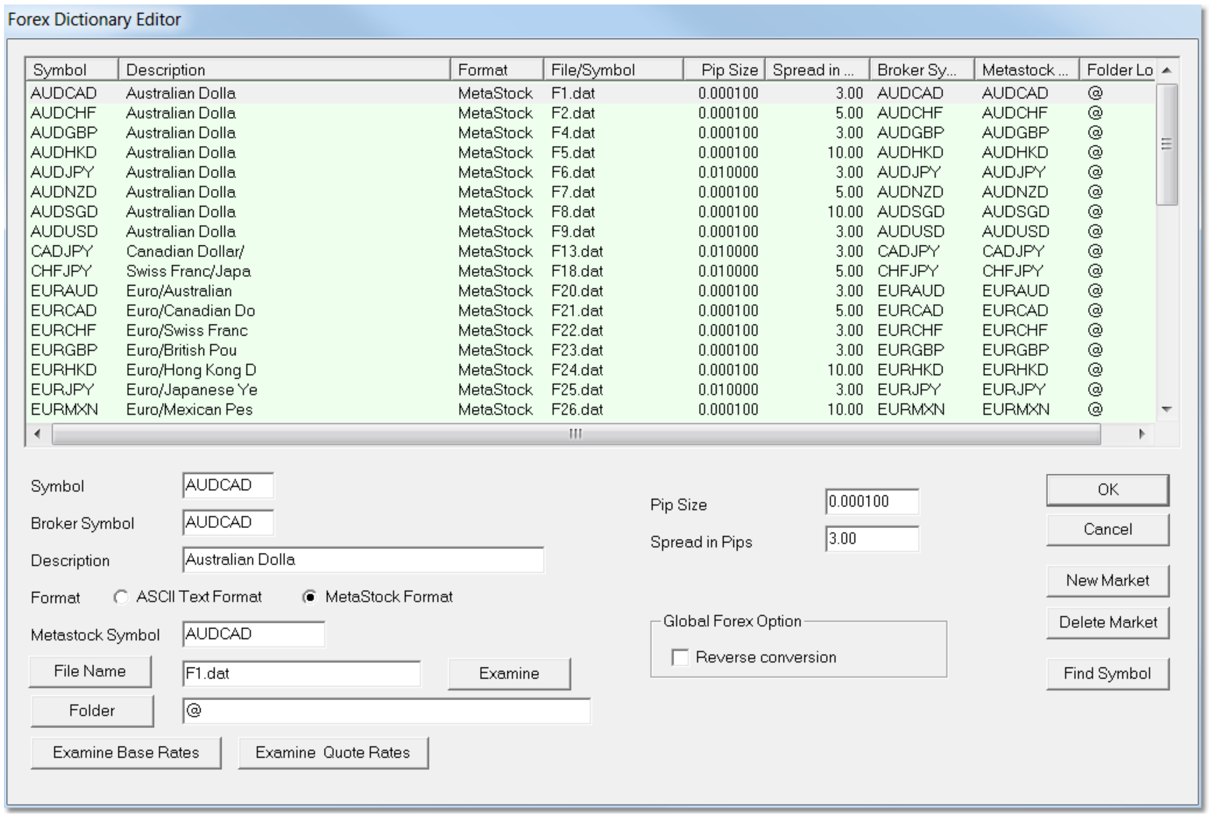
Or you can press the Examine button next to the Exchange dropdown in the Futures Dictionary. The exchanges are the same for futures and stocks, so this will work for both.

The exchange.csv file is located in the Dictionaries folder, as defined in preferences, default is Trading Blox/Data/Dictionaries.

Section 1 – Forex Dictionary Editor

You can access the Forex Dictionary editor by choosing the Edit -> Forex Dictionary menu.

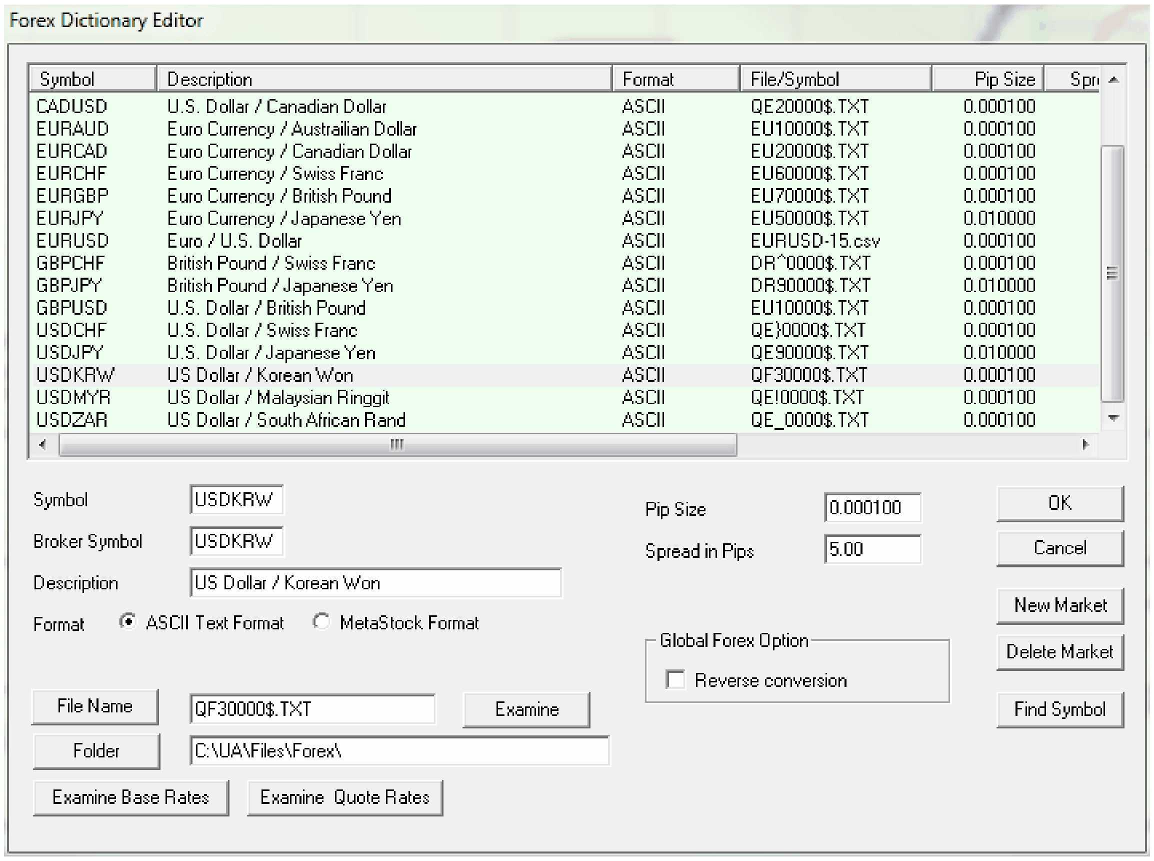
See Data and Folder Preferences to change the default data folder from the sample data to your own data source. When changing your data source, be sure to update the Forex Dictionary and move the interest rates files to the new location.



Symbol	Description	Format	File/Symbol	Pip Size	Spread in ...	Broker Sy...	Metastock ...	Folder Lo
AUDCAD	Australian Dolla	MetaStock	F1.dat	0.000100	3.00	AUDCAD	AUDCAD	@
AUDCHF	Australian Dolla	MetaStock	F2.dat	0.000100	5.00	AUDCHF	AUDCHF	@
AUDGBP	Australian Dolla	MetaStock	F4.dat	0.000100	3.00	AUDGBP	AUDGBP	@
AUDHKD	Australian Dolla	MetaStock	F5.dat	0.000100	10.00	AUDHKD	AUDHKD	@
AUDJPY	Australian Dolla	MetaStock	F6.dat	0.010000	3.00	AUDJPY	AUDJPY	@
AUDNZD	Australian Dolla	MetaStock	F7.dat	0.000100	5.00	AUDNZD	AUDNZD	@
AUDSGD	Australian Dolla	MetaStock	F8.dat	0.000100	10.00	AUDSGD	AUDSGD	@
AUDUSD	Australian Dolla	MetaStock	F9.dat	0.000100	3.00	AUDUSD	AUDUSD	@
CADJPY	Canadian Dollar/	MetaStock	F13.dat	0.010000	3.00	CADJPY	CADJPY	@
CHFJPY	Swiss Franc/Japa	MetaStock	F18.dat	0.010000	5.00	CHFJPY	CHFJPY	@
EURAUD	Euro/Australian	MetaStock	F20.dat	0.000100	3.00	EURAUD	EURAUD	@
EURCAD	Euro/Canadian Do	MetaStock	F21.dat	0.000100	5.00	EURCAD	EURCAD	@
EURCHF	Euro/Swiss Franc	MetaStock	F22.dat	0.000100	3.00	EURCHF	EURCHF	@
EURGBP	Euro/British Pou	MetaStock	F23.dat	0.000100	3.00	EURGBP	EURGBP	@
EURHKD	Euro/Hong Kong D	MetaStock	F24.dat	0.000100	10.00	EURHKD	EURHKD	@
EURJPY	Euro/Japanese Ye	MetaStock	F25.dat	0.010000	3.00	EURJPY	EURJPY	@
EURMXN	Euro/Mexican Pes	MetaStock	F26.dat	0.000100	10.00	EURMXN	EURMXN	@

Symbol: AUDCAD
Broker Symbol: AUDCAD
Description: Australian Dolla
Format: ASCII Text Format [✓ MetaStock Format]
Metastock Symbol: AUDCAD
File Name: F1.dat
Folder: @
Pip Size: 0.000100
Spread in Pips: 3.00
Global Forex Option: Reverse conversion ☐

Here is an example of adding the Korean Won to the Forex Dictionary. The symbol in CSI UA is QF3, thus the file name. The folder location is typically UA\Files\Forex using the default setup. Adding USDKRW is required to trade any KRW Futures or Stocks.
To add a new market, press the New Market button, enter the symbol of USDKRW, press the File Name button and select the data file.



Forex Dictionary Editor

Symbol	Description	Format	File/Symbol	Pip Size	Spr
CADUSD	U.S. Dollar / Canadian Dollar	ASCII	QE20000$.TXT	0.000100	
EURAUD	Euro Currency / Austrailian Dollar	ASCII	EU10000$.TXT	0.000100	
EURCAD	Euro Currency / Canadian Dollar	ASCII	EU20000$.TXT	0.000100	
EURCHF	Euro Currency / Swiss Franc	ASCII	EU60000$.TXT	0.000100	
EURGBP	Euro Currency / British Pound	ASCII	EU70000$.TXT	0.000100	
EURJPY	Euro Currency / Japanese Yen	ASCII	EU50000$.TXT	0.010000	
EURUSD	Euro / U.S. Dollar	ASCII	EURUSD-15.csv	0.000100	
GBPCHF	British Pound / Swiss Franc	ASCII	DR^0000$.TXT	0.000100	
GBPJPY	British Pound / Japanese Yen	ASCII	DR90000$.TXT	0.010000	
GBPUSD	U.S. Dollar / British Pound	ASCII	EU10000$.TXT	0.000100	
USDCHF	U.S. Dollar / Swiss Franc	ASCII	QE}0000$.TXT	0.000100	
USDJPY	U.S. Dollar / Japanese Yen	ASCII	QE90000$.TXT	0.010000	
USDKRW	US Dollar / Korean Won	ASCII	QF30000$.TXT	0.000100	
USDMYR	US Dollar / Malaysian Ringgit	ASCII	QE!0000$.TXT	0.000100	
USDZAR	US Dollar / South African Rand	ASCII	QE_0000$.TXT	0.000100	

Symbol: USDKRW
Broker Symbol: USDKRW
Description: US Dollar / Korean Won
Format: [✓ ASCII Text Format] MetaStock Format
Pip Size: 0.000100
Spread in Pips: 5.00
Global Forex Option: ☐ Reverse conversion
File Name: QF30000$.TXT
Folder: C:\UA\Files\Forex\

Symbol
The symbol for the market. These symbols should be composed of six letter combinations of ISO Standard Currency Codes with the first three letters as the ISO code for the base currency, and the last three as the ISO code for the quote currency. The symbol can include additional characters, such as AUDCAD-5 to represent a 5 minute bar version of the symbol.

Description
The name of the market longhand. When creating a new market Trading Blox will fill in this field automatically if the six letter symbol is constructed of valid ISO codes.

Data Format
The data format: ASCII or MetaStock.

ASCII File
For ASCII text format, the name of the file must be input here.

Metastock Symbol
For MetaStock this should be the symbol, which is used to extract information from the MetaStock MASTER file.

File Name
The name of the file where the data for this market is located. Press the File Name button to locate the file using a browser interface.

Folder
The folder where the file is located. The "@" represents the default location as set in the Forex Data Folder in preferences. Press the Folder button to locate this folder using a browser interface.

Examine
Press this button to open the data file in notepad or excel for editing or viewing.

Examine Base Rates
Press this button to open the base currency interest rates file for editing or viewing. The rates file is located in the Forex Data Folder, and has an ISO_Rates.txt format. Example USD_Rates.txt for the USD rates.

Examine Quote Rates
Press this button to open the quote currency interest rates file for editing or viewing. The rates file is located in the Forex Data Folder, and has an ISO_Rates.txt format.

Pip Size
The minimum trading increment or PIP. Usually 0.01 or 0.0001.

Spread in Pips
The spread used for both entry and exit slippage. This value is designed to support simulation of retail Forex brokerage slippage which is generally a fixed number of pips. The exact value depends on the broker and the market.

Reverse Conversion
This box, if checked, will reverse the currency conversion for use with non-USD denominated futures, stocks, and other currency conversions.

Find Symbol
Used primarily to find metastock file names. In the example above, the metastock symbol is AUDCAD, and the Find Symbol button will determine from the master file that the metastock file name is F1.dat.

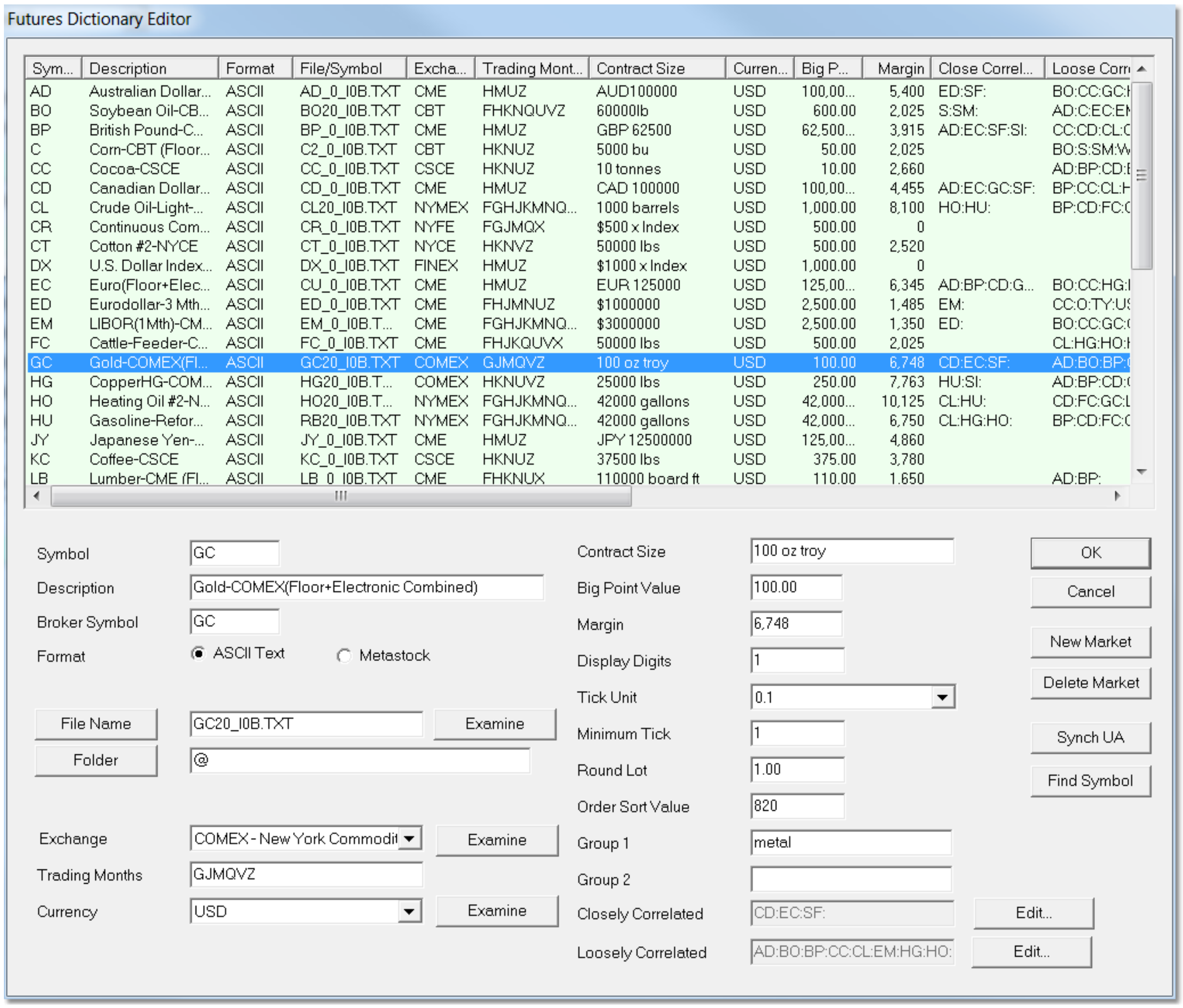
More tutorials:

http://www.tradingblox.com/tradingblox/howto-ua.htm
http://www.tradingblox.com/tradingblox/howto-ua-newmarket.htm
http://www.tradingblox.com/tradingblox/CSIUA/howtouastocks.doc
http://www.tradingblox.com/tradingblox/CSIUA/Trading%20Blox%20CSI%20Forex.doc

Section 2 – Futures Dictionary Editor

This allows you to edit and modify the futures market information. This file is located in the Dictionary folder. Alternately, it can be located in your Futures Data folder as well. Trading Blox will look in the Futures Data folder first, and if the file is not found it will look in the Dictionary folder.

You can access the Futures Dictionary editor choosing the Edit -> Futures Dictionary menu. This will bring up the Futures Dictionary Editor:



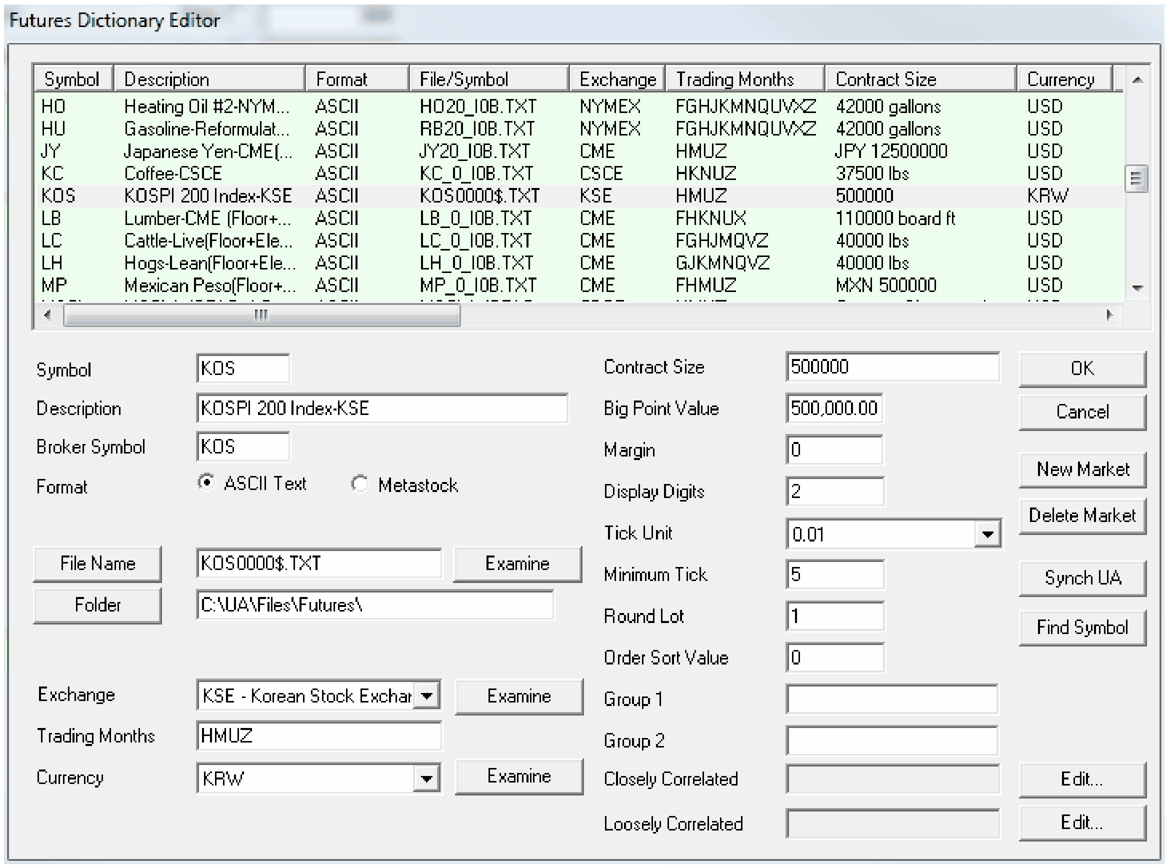
Futures Dictionary Editor

Sym...	Description	Format	File/Symbol	Excha...	Trading Mont...	Contract Size	Curren...	Big P...	Margin	Close Correl...	Loose Corr
AD	Australian Dollar...	ASCII	AD_0_I0B.TXT	CME	HMUZ	AUD100000	USD	100,00...	5,400	ED:SF:	BO:CC:GC:
BO	Soybean Oil-CB...	ASCII	BO20_I0B.TXT	CBT	FHKNQUVZ	60000lb	USD	600.00	2,025	S:SM:	AD:C:EC:E
BP	British Pound-C...	ASCII	BP_0_I0B.TXT	CME	HMUZ	GBP 62500	USD	62,500...	3,915	AD:EC:SF:SI:	CC:CD:CL:
C	Corn-CBT (Floor...	ASCII	C2_0_I0B.TXT	CBT	HKNUZ	5000 bu	USD	50.00	2,025		BO:S:SM:
CC	Cocoa-CSCE	ASCII	CC_0_I0B.TXT	CSCE	HKNUZ	10 tonnes	USD	10.00	2,660		AD:BP:CD:
CD	Canadian Dollar...	ASCII	CD_0_I0B.TXT	CME	HMUZ	CAD 100000	USD	100,00...	4,455	AD:EC:GC:SF:	BP:CC:CL:
CL	Crude Oil-Light-...	ASCII	CL20_I0B.TXT	NYMEX	FGHJKMNQ...	1000 barrels	USD	1,000.00	8,100	HO:HU:	BP:CD:FC:
CR	Continuous Com...	ASCII	CR_0_I0B.TXT	NYFE	FGJMQX	$500 x Index	USD	500.00	0		
CT	Cotton #2-NYCE	ASCII	CT_0_I0B.TXT	NYCE	HKNVZ	50000 lbs	USD	500.00	2,520		
DX	U.S. Dollar Index...	ASCII	DX_0_I0B.TXT	FINEX	HMUZ	$1000 x Index	USD	1,000.00	0		
EC	Euro(Floor+Elec...	ASCII	CU_0_I0B.TXT	CME	HMUZ	EUR 125000	USD	125,00...	6,345	AD:BP:CD:G...	BO:CC:HG:
ED	Eurodollar-3 Mth...	ASCII	ED_0_I0B.TXT	CME	FHJMNUZ	$1000000	USD	2,500.00	1,485	EM:	CC:O:TY:U
EM	LIBOR(1Mth)-CM...	ASCII	EM_0_I0B.T...	CME	FGHJKMNQ...	$3000000	USD	2,500.00	1,350	ED:	BO:CC:GC:
FC	Cattle-Feeder-C...	ASCII	FC_0_I0B.TXT	CME	FHJKQUVX	50000 lbs	USD	500.00	2,025		CL:HG:HO:
GC	Gold-COMEX(Fl...	ASCII	GC20_I0B.TXT	COMEX	GJMQVZ	100 oz troy	USD	100.00	6,748	CD:EC:SF:	AD:BO:BP:
HG	CopperHG-COM...	ASCII	HG20_I0B.T...	COMEX	HKNUVZ	25000 lbs	USD	250.00	7,763	HU:SI:	AD:BP:CD:
HO	Heating Oil #2-N...	ASCII	HO20_I0B.T...	NYMEX	FGHJKMNQ...	42000 gallons	USD	42,000...	10,125	CL:HU:	CD:FC:GC:
HU	Gasoline-Refor...	ASCII	RB20_I0B.TXT	NYMEX	FGHJKMNQ...	42000 gallons	USD	42,000...	6,750	CL:HG:HO:	BP:CD:FC:
JY	Japanese Yen-...	ASCII	JY_0_I0B.TXT	CME	HMUZ	JPY 12500000	USD	125,00...	4,860		
KC	Coffee-CSCE	ASCII	KC_0_I0B.TXT	CSCE	HKNUZ	37500 lbs	USD	375.00	3,780		
LB	Lumber-CME (Fl...	ASCII	LB_0_I0B.TXT	CME	FHKNUX	110000 board ft	USD	110.00	1,650		AD:BP:

Symbol	GC
Description	Gold-COMEX(Floor+Electronic Combined)
Broker Symbol	GC
Format	[✓ ASCII Text] Metastock
File Name	GC20_I0B.TXT
Folder	@
Exchange	COMEX - New York Commodit
Trading Months	GJMQVZ
Currency	USD
Contract Size	100 oz troy
Big Point Value	100.00
Margin	6,748
Display Digits	1
Tick Unit	0.1
Minimum Tick	1
Round Lot	1.00
Order Sort Value	820
Group 1	metal
Group 2	
Closely Correlated	CD:EC:SF:
Loosely Correlated	AD:BO:BP:CC:CL:EM:HG:HO:

Here is an example of adding a Foreign Market, such as KOS. The Kospi is denominated in KRW, so the USDKRW would need to be added to the Forex Dictionary and linked to the appropriate data file to convert the KRW back to system wide base currency.

To add a new market, Press New Market, Enter the symbol KOS, Press Synch UA, then Press the File Name button to select the data file.



Futures Dictionary Editor

Symbol	Description	Format	File/Symbol	Exchange	Trading Months	Contract Size	Currency
HO	Heating Oil #2-NYM...	ASCII	HO20_I0B.TXT	NYMEX	FGHJKMNQUVXZ	42000 gallons	USD
HU	Gasoline-Reformulat...	ASCII	RB20_I0B.TXT	NYMEX	FGHJKMNQUVXZ	42000 gallons	USD
JY	Japanese Yen-CME(...	ASCII	JY20_I0B.TXT	CME	HMUZ	JPY 12500000	USD
KC	Coffee-CSCE	ASCII	KC_0_I0B.TXT	CSCE	HKNUZ	37500 lbs	USD
KOS	KOSPI 200 Index-KSE	ASCII	KOS0000$.TXT	KSE	HMUZ	500000	KRW
LB	Lumber-CME (Floor+...	ASCII	LB_0_I0B.TXT	CME	FHKNUX	110000 board ft	USD
LC	Cattle-Live(Floor+Ele...	ASCII	LC_0_I0B.TXT	CME	FGHJMQVZ	40000 lbs	USD
LH	Hogs-Lean(Floor+Ele...	ASCII	LH_0_I0B.TXT	CME	GJKMNQVZ	40000 lbs	USD
MP	Mexican Peso(Floor+...	ASCII	MP_0_I0B.TXT	CME	FHMUZ	MXN 500000	USD

Symbol	KOS
Description	KOSPI 200 Index-KSE
Broker Symbol	KOS
Format	[✓ ASCII Text] Metastock
File Name	KOS0000$.TXT
Folder	C:\UA\Files\Futures\
Exchange	KSE - Korean Stock Exchar
Trading Months	HMUZ
Currency	KRW
Contract Size	500000
Big Point Value	500,000.00
Margin	0
Display Digits	2
Tick Unit	0.01
Minimum Tick	5
Round Lot	1
Order Sort Value	0
Group 1	
Group 2	
Closely Correlated	
Loosely Correlated	

Symbol
The symbol for the market. The symbol in a futures portfolio must match this symbol.

Synch UA -- This button will take the Symbol entered, and look up the data for this symbol in the CSI UA program on your computer. If you do not have CSI UA on your computer, you cannot use this button. But if you do, it will bring all the information it can from the market database and populate the fields. If you do not like what it has done, press cancel.

Find Symbol -- This button will use the Symbol to try to find the corresponding data file in the Futures Data Folder as defined in preference. It will look for a file starting with the symbol. If the symbol is just one letter, it will append a "_" and look for the file. Example: Symbol "S" will look for files starting with "S_", symbol "AD" will look for files starting with "AD". If it does not find the file, it will return a blank. If it finds more than one file, it will use the first one. Please be sure to check if this is really the file you want, since your symbol and datafile will be linked at this point.

Description
The name of the market longhand.

Broker Symbol
The symbol used by your broker for this market.

Data Format
The data format: ASCII or MetaStock.

Metastock Symbol (if metastock data format)

For MetaStock format data it defines the symbol used within the MetaStock MASTER file since these sometimes do not correspond exactly with the symbol name as used by the exchange.

File Name
This is the actual file name of the data file that will be linked to this symbol. Press the File Name button to locate the file using a browser interface.

Folder Location
This is the actual folder location of the data file. Press on the Folder Location button to locate the folder using a browser interface. The "@" sign is used to represent the default data folder as set in preferences. In this way the Dictionary is transportable to another installation of Trading Blox. If you hard code the folder location using C:/xxx then it will not be transportable, but you can specify any location even outside of the Trading Blox folder or the default data folder as set in preferences.

Examine
Press this button to open the data file in notepad or excel, for editing or viewing.

Exchange
The exchange that the market trades on.

Examine
Press this button to open the exchanges file in notepad or excel, for editing or viewing.

Trading Months
The months that the market trades using standard futures month letters. This is used to determine how many times per year account for contract rolls, when you have the Account for Contract Rolls global parameter set to true. The more months you have here, the more times the system will simulate a roll, and account for commission and slippage. Note that if you have the delivery month in your data, then this is not used at all. But if you don't have the delivery month in your data, and you want to account for contract rolls, a roll we be assumed every x bars where x is 250 divided by the number of months entered here. Good for daily data, but not for intraday or weekly data.

F - January
G - February
H - March
J - April
K - May
M - June
N - July
Q - August
U - September
V - October
X - November
Z - December

Currency
The ISO Currency Code that the contract is denominated in. U.S. markets are denominated in USD, European markets are generally denominated in either USD - U.S. Dollars, GBP - British Pounds or EUR - EEC Euros.

If you select a currency other than the system wide base currency, the system will use the corresponding forex file to convert the prices into the system wide base currency. So you can test futures with different currencies in one test, and the results will all be in the system wide base currency. You should enter the Big Point Value and Margin in the foreign currency, since the system will convert these into the system wide base currency for you.

If your system wide base currency is USD, and you select EUR as the currency, the system will look for the EURUSD forex file and assume it is in the format USD per EUR. If it does not find this file, it will look for USDEUR and assume it is in the format EUR per USD.

If you want to reverse the way the system uses these files, you can check "Reverse Conversion" in the Forex Dictionary.

Examine
Press this button to open the currency file in notepad or excel, for editing or viewing.

Contract Size
The contract size description, e.g. Sugar - 112,000 pounds.

Big Point Value
The value of a 1.0 price movement (see Determining Big Point Value for a detailed explanation). The Big Point Value should be expressed in the currency of the underlying contract.

Margin
The margin used to simulate margin calls while testing.

Display Digits
The number of significant decimal digits displayed for this market.

Tick Unit
The unit of the tick. This is either a decimal like 1.0, 0.1, 0.01, etc. or a fraction like 1/4, 1/8, 1/32 etc.

Minimum Tick
The number of tick units which constitutes a minimum tick. For example, if the minimum tick is 0.025 or twenty five thousands, the tick unit is thousandths, 0.001 and the minimum tick value is 25.

Round Lot
The round lot for trading. If you enter a value of 100 here, Trading Blox will only trade in increments of 100.

Order Sort Value
This value determines the sort order of the markets for order generation. The sample data sets these values to a number corresponding to the market open time using U.S. Eastern Standard Time. This results in order files that are in the same order as the market open. For example, the 820 cited above for AD - Australian Dollars corresponds to the 8:20 AM market opening of the AD on the Chicago Mercantile Exchange.

Group1
This string is the group1 to which the market belongs. It can be any string or number, up to 32 characters. When more than one market have the same group, you can access certain instrument properties in scripting such as group margin and group risk, etc.

Group2
This string is the group2 to which the market belongs. It can be any string or number, up to 32 characters. When more than one market have the same group, you can access certain instrument properties in scripting such as group margin and group risk, etc.

Closely Correlated
The markets which closely correlate to the given market. Clicking on the Edit button will bring up an editor which allows you to select that markets which closely correlate to a particular market.

Loosely Correlated

The markets which loosely correlate to the given market. Clicking on the Edit button will bring up an editor which allows you to select that markets which loosely correlate to a particular market.

More tutorials:

http://www.tradingblox.com/tradingblox/howto-ua.htm
http://www.tradingblox.com/tradingblox/howto-ua-newmarket.htm
http://www.tradingblox.com/tradingblox/CSIUA/howtouastocks.doc
http://www.tradingblox.com/tradingblox/CSIUA/Trading%20Blox%20CSI%20Forex.doc

Section 3 – The Stock Dictionary

The Stock Dictionary *does not have a graphical editor* like Futures Dictionary and Forex Dictionary do because the data format for stocks is simpler and more consistent across different data vendors. Instead, the Stock Dictionary is defined using an ASCII Text file called StockInfo.txt which is located in the Dictionary Folder. Alternately the file can be located in the Stock Data Folder. Trading Blox will look in the Stock Data Folder first, and if the file is not found, it will look in the Dictionary folder.

This file is optional. If you don't have a stockinfo.txt file present in the Stock Data Folder or the Dictionaries Folder Trading Blox will look for all Metastock files and all text files ending in ".txt" or ".csv". It will take these files and create symbols. A file named "IBM.txt" will create a symbol called IBM with a name of IBM and read the data.

Trading Blox will look for all data files in all linked folders, and make them available for trading. It will then look for the Stock Dictionary file and append any additional information. So you can have a large number of stock files, and a small dictionary with extra data. This also means you can add industries and groups to Metastock data.

Important Note: If you have a Stock Dictionary, then only those stocks in the dictionary will be available for testing. If you add new data files to your data folders they will not be available for testing until you have added the stock to the Stock Dictionary. If you are using CSI this process is easy, in that you can use the Create Stock Dictionary button in preferences. If you don't need the description, sector, industry, and round lot, etc then you can delete the Stock Dictionary (stockinfo.txt) file. Trading Blox will then load up all text files for processing.

Format

The format of the stockinfo.txt file is:
```
Symbol,Description,Exchange,Sector/Group1,Industry/Group2,Country,Data
Vendor ID,Broker Symbol,Active Status,Round Lot,Currency
```

If you want to use a header, be sure to indicate the file is version 6 or greater on the first line. If this version information is missing, or less than version 6, Trading Blox will not accept the header.

Example:

Version, 6
Symbol,Description,Exchange,Sector,Industry,Country,Data Vendor ID,Broker Symbol,Active Status,Round Lot,Currency
GV,Goldfield Cp,AMEX,IG,HC,USA,3243,GV,A,1,USD
CGV,CGG Veritas,NYSE,BM,OE,USA,11512,CGV,A,1,USD

For each of the instruments listed, there needs to be a corresponding data file. See Historical Data Setup for more information.

Volume Multiplier

Be sure to set the Volume Multiplier in preferences. So if your stock data reports a volume of 1,000, we can translate that into 100,000. This is the way CSI reports the data. If your data provider uses some other multiplier, or reports the actual data, you would want to use a different setting in preferences. The default is 100.

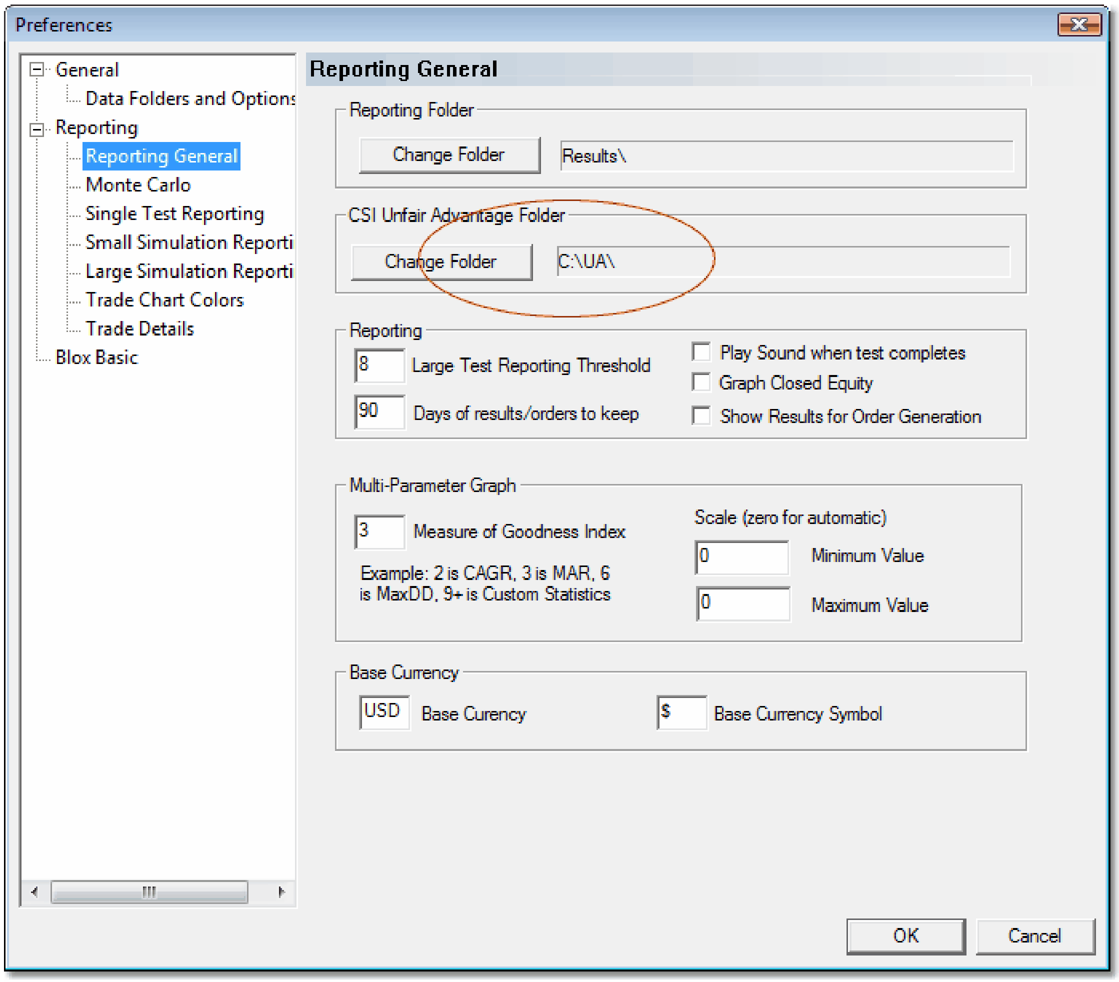
The volume is used by Trading Blox to check Minimum Volume and Maximum Percent Volume Per Trade, both global variables.

Currency Conversion

Trading Blox will look first to the Stock Dictionary for the currency of the stock. If there is no currency it will use the system wide base currency. Be sure to have the corresponding forex pair available in the forex data folder when doing currency conversions.

Create a Stock Dictionary using the CSI UA information

Set the correct folder location of the CSI installation in the Trading Blox Preferences



Reporting General

Reporting Folder: Results\

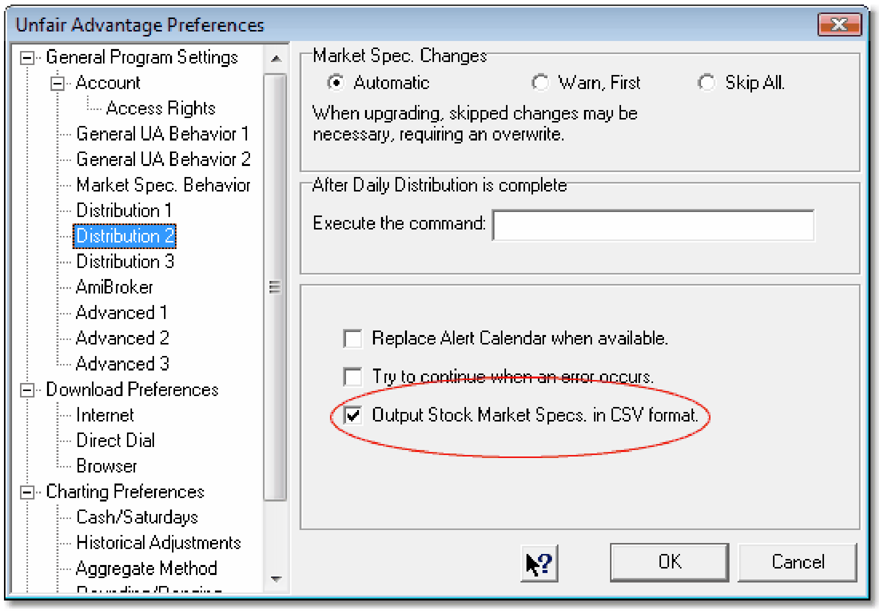
CSI Unfair Advantage Folder: C:\UA\

Reporting
- 8 Large Test Reporting Threshold
- 90 Days of results/orders to keep
- [ ] Play Sound when test completes
- [ ] Graph Closed Equity
- [ ] Show Results for Order Generation

Multi-Parameter Graph
- 3 Measure of Goodness Index
- Example: 2 is CAGR, 3 is MAR, 6 is MaxDD, 9+ is Custom Statistics
- Scale (zero for automatic)
- 0 Minimum Value
- 0 Maximum Value

Base Currency
- USD Base Currency
- $ Base Currency Symbol

Set the CSI UA preferences to download the market data file:


Unfair Advantage Preferences

- General Program Settings
  - Account
    - Access Rights
  - General UA Behavior 1
  - General UA Behavior 2
  - Market Spec. Behavior
  - Distribution 1
  - Distribution 2
  - Distribution 3
  - AmiBroker
  - Advanced 1
  - Advanced 2
  - Advanced 3
- Download Preferences
  - Internet
  - Direct Dial
  - Browser
- Charting Preferences
  - Cash/Saturdays
  - Historical Adjustments
  - Aggregate Method

Market Spec. Changes

[✓ Automatic] Warn, First Skip All.

When upgrading, skipped changes may be necessary, requiring an overwrite.

After Daily Distribution is complete

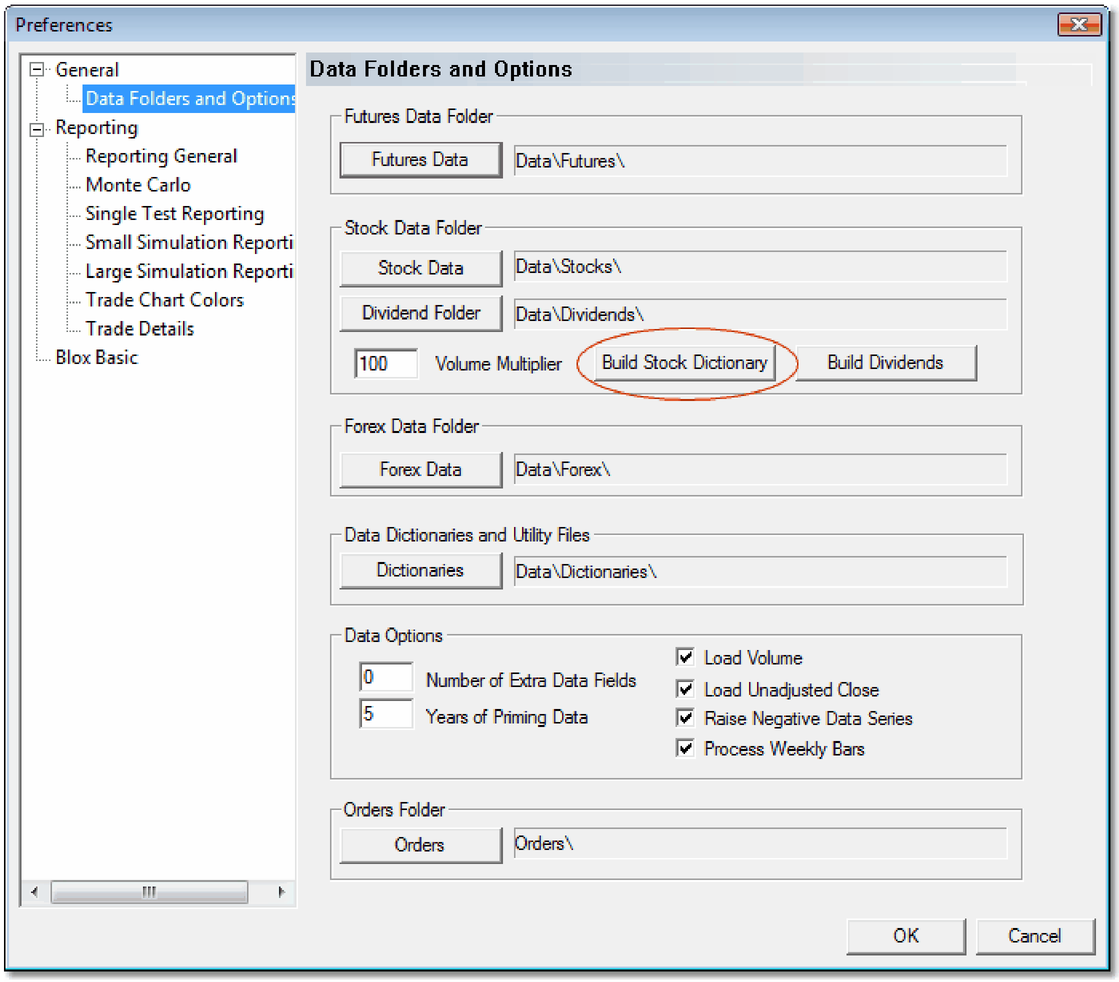
Execute the command:

☐ Replace Alert Calendar when available.

☐ Try to continue when an error occurs.

☑ Output Stock Market Specs. in CSV format.

Then press the "Get Data" button in CSI UA to update your data and download this file.
Now Trading Blox will be able to build a stock dictionary file from the CSI market data using this button in preferences:


Preferences

- General
  - Data Folders and Options
- Reporting
  - Reporting General
  - Monte Carlo
  - Single Test Reporting
  - Small Simulation Reporti
  - Large Simulation Reporti
  - Trade Chart Colors
  - Trade Details
- Blox Basic

Data Folders and Options

Futures Data Folder
Futures Data: Data\Futures\

Stock Data Folder
Stock Data: Data\Stocks\
Dividend Folder: Data\Dividends\
100 Volume Multiplier
Build Stock Dictionary
Build Dividends

Forex Data Folder
Forex Data: Data\Forex\

Data Dictionaries and Utility Files
Dictionaries: Data\Dictionaries\

Data Options
0 Number of Extra Data Fields
5 Years of Priming Data
☑ Load Volume
☑ Load Unadjusted Close
☑ Raise Negative Data Series
☑ Process Weekly Bars

Orders Folder
Orders: Orders\

OK
Cancel

This process will delete your old stock dictionary and build a new one. It will build the file for all data files in your data folder. If you add new data files into your data folder later, you will need to create a new stock dictionary. Only stocks in the stock dictionary will be available for testing. If you add a new data file later, it will not be available for testing until added to the stock dictionary, or the stock dictionary is removed.

More Tutorials:

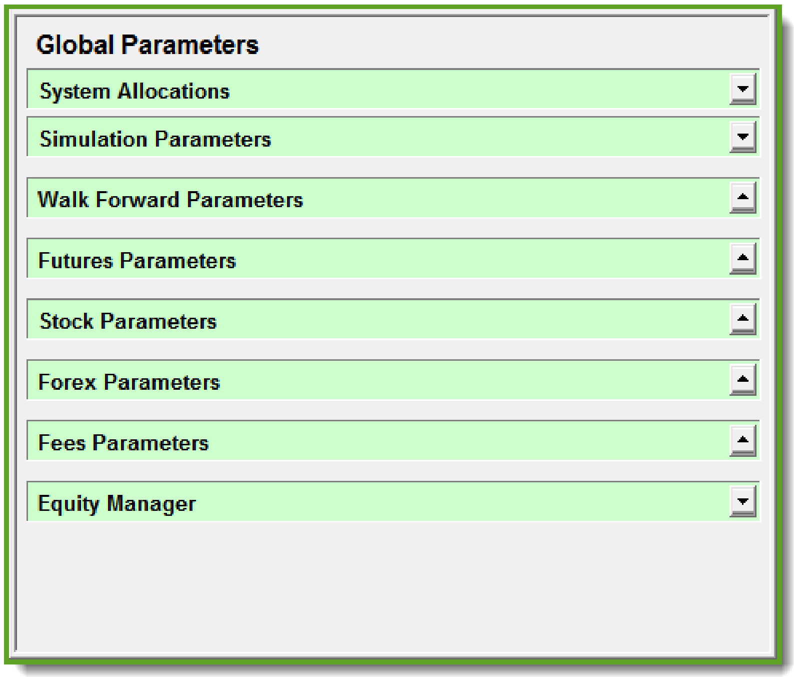
http://www.tradingblox.com/tradingblox/howto-ua.htm
http://www.tradingblox.com/tradingblox/howto-ua-newmarket.htm
http://www.tradingblox.com/tradingblox/CSIUA/howtouastocks.doc
http://www.tradingblox.com/tradingblox/CSIUA/Trading%20Blox%20CSI%20Forex.doc

Part IV

Part 4 – Global Parameters

Global parameters affect every system in the test and include global equity management parameters like: Account Balance, Equity Base, etc. as well as Simulation Parameters like slippage, commission, and interest rate assumptions. This section describes the *Global Parameters.*

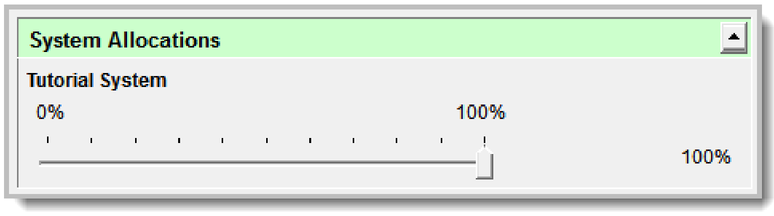
The Global Parameters are further divided into eight sections:

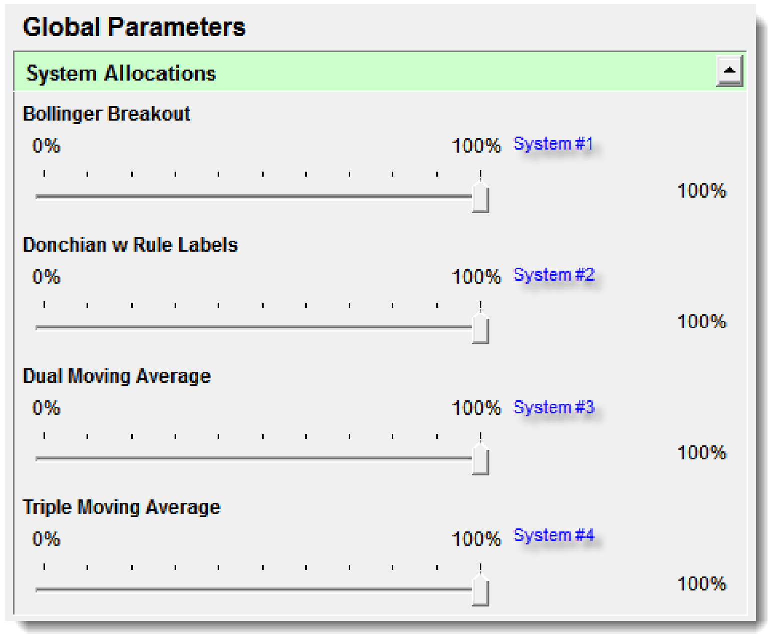


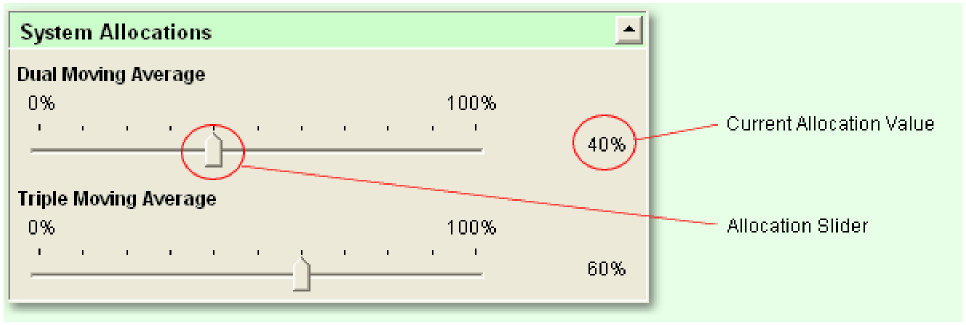
Parameter Setting Sections:	**Description:**
System Allocations	When multiple systems are active, each can be allocated a percentage of the equity.
Simulations	General parameters like commission rates, slippage assumptions, interest rate assumptions, etc.
Walk Forward	
Futures	
Stocks	
Forex	
Fees	
Equity Management	Parameters affecting the equity management including the starting account balance, the leverage, the type of equity used for position sizing calculations known as the *Trading Equity*, etc.

Section 1 – System Allocations

For multiple-system tests, Trading Blox's uses a set of system allocation controls to set percentage-based allocations for each of the active systems.







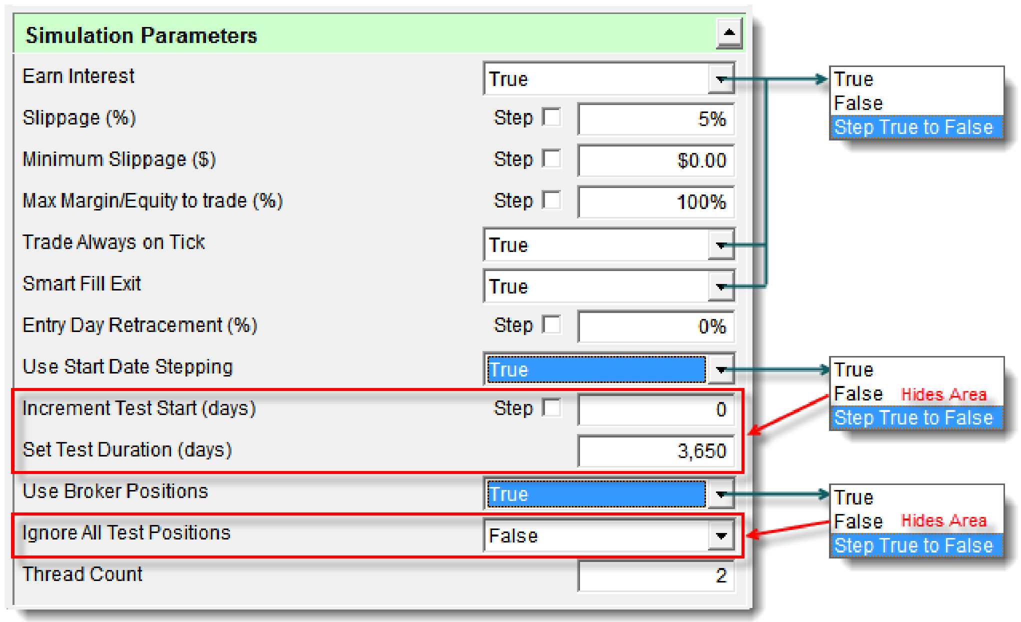
Clicking and dragging the *Allocation Slider* to the right or left will increase or decrease the current allocation values. The Current Allocation Value is displayed to the right of the allocation control for each system.

Trading Blox will allow allocations that total more than 100%. At first glance, it might seem that this would violate common sense and should not be permitted. However, since futures contracts have high leverage most of the time a single trading system uses a small percentage of the available equity for margin. Consider two trading systems each of which uses a maximum of 25% of trading equity for margin. A single account could easily trade both systems at the same time. Trading Blox's permission of allocations totalling more than 100% allows you to test this ability.

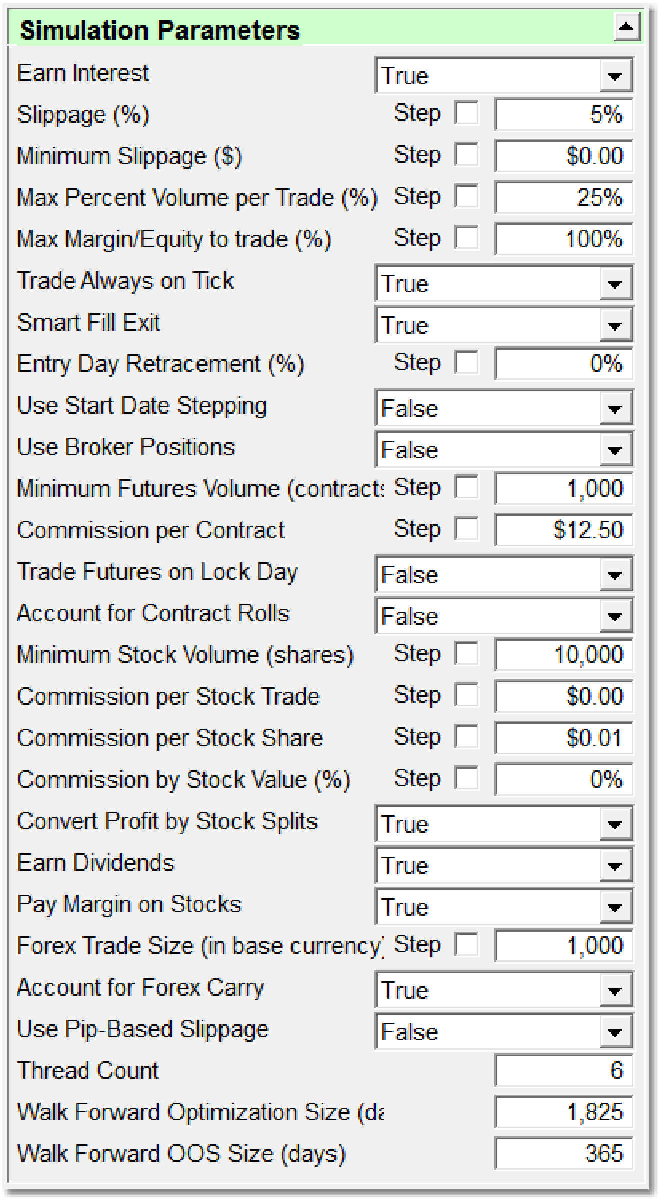
The trading equity used in the Fixed Fractional Money Manager to determine trade size is the test equity times this allocation slider times the leverage adjusted by the drawdown threshold. So each system can trade at 100% of the total test equity, or can trade at a smaller percent.

Section 2 – Simulation Parameters

The following is a comprehensive list of the Trading Blox global *Simulation Parameters*, and an explanation of how each is used. The global *Test Parameter* values can be viewed and changed using the *Global Parameters* tab. To reset the parameters to their default values, the following picture shows them.



Simulation Parameters		
Earn Interest		True
Slippage (%)	Step ☐	5%
Minimum Slippage ($)	Step ☐	$0.00
Max Margin/Equity to trade (%)	Step ☐	100%
Trade Always on Tick		True
Smart Fill Exit		True
Entry Day Retracement (%)	Step ☐	0%
Use Start Date Stepping		True
Increment Test Start (days)	Step ☐	0
Set Test Duration (days)		3,650
Use Broker Positions		True
Ignore All Test Positions		False
Thread Count		2



Earn Interest		True
Slippage (%)	Step ☐	5%
Minimum Slippage ($)	Step ☐	$0.00
Max Percent Volume per Trade (%)	Step ☐	25%
Max Margin/Equity to trade (%)	Step ☐	100%
Trade Always on Tick		True
Smart Fill Exit		True
Entry Day Retracement (%)	Step ☐	0%
Use Start Date Stepping		False
Use Broker Positions		False
Minimum Futures Volume (contract:	Step ☐	1,000
Commission per Contract	Step ☐	$12.50
Trade Futures on Lock Day		False
Account for Contract Rolls		False
Minimum Stock Volume (shares)	Step ☐	10,000
Commission per Stock Trade	Step ☐	$0.00
Commission per Stock Share	Step ☐	$0.01
Commission by Stock Value (%)	Step ☐	0%
Convert Profit by Stock Splits		True
Earn Dividends		True
Pay Margin on Stocks		True
Forex Trade Size (in base currency	Step ☐	1,000
Account for Forex Carry		True
Use Pip-Based Slippage		False
Thread Count		6
Walk Forward Optimization Size (da		1,825
Walk Forward OOS Size (days)		365

Earn Interest
Set to true to enable earning interest on available cash. Set to false to disable earning interest. The rates are set in the currency rate file for your system base currency. If you are using USD as your system wide base currency, then the rates are in the USD_Rates.txt file in the Forex data directory. The Lending Rate is used when earning interest.

Earned Interest is calculated on the cash balance of the account. Money used to purchase stocks is deducted from this cash balance. Money used as margin for futures trading is not deducted from the cash balance since many brokers will allow T-Bills for margin requirements allowing traders to earn interest on the money that is used for margin.

Slippage
The frictional cost of trading has two components: commissions, and slippage (sometimes also known as "skid"). In actual trading, slippage is the difference between a trade's entry or exit order price, and

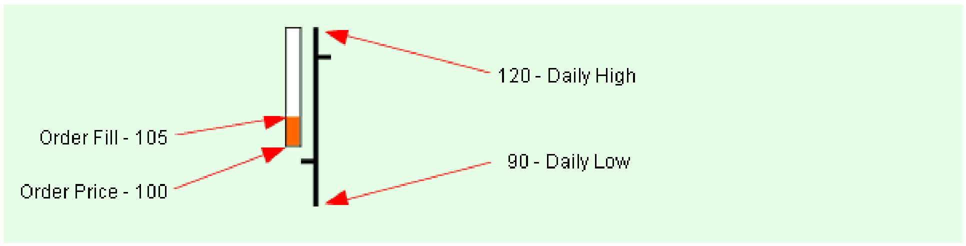
the price at which the trade is actually filled. In order to accurately reflect the conditions of real trading, the impact of slippage must be simulated during back testing.

Since slippage can vary dramatically from trade to trade, depending on market conditions at the time an order is executed, Trading Blox employs a slippage assessment technique that is based on market volatility.

The simulated fill price is obtained by calculating a slippage factor, which is added to (or subtracted from), the theoretical entry price.

For a long entry, the slippage factor is calculated by measuring the range from the theoretical entry price to the day's highest price, and multiplying that amount by the *Slippage Percent*. (For short entries, the slippage factor is calculated by measuring the range from the theoretical entry price to the low). The slippage factor is then added to, or subtracted from the theoretical entry price, to obtain the simulated fill price.

Here's how it works for a buy trade:



Slippage percent	25%
Theoretical buy order price	100
High Price (for the day)	120
Slippage Factor	(120 - 100) x 0.25 = (20 x 0.25) = 5
Simulated fill price	Order Price + Slippage Factor = (100 + 5) = 105

The distance between the high price and the order price is multiplied by the slippage factor. In this example, the difference between the high price and the order price is 20 points. The 20 points are multiplied by the 25% slippage to get an estimated slippage of 5 points. The fill price for the order will be 5 points worse than the stop order price of 100 simulating a fill at 105.

Slippage for sell orders is computed using a similar calculation using the distance between the order price and the low of the day.

In historical back testing, failing to accurately estimate slippage can lead to two types of mistakes: Underestimating frictional costs may lead you to trade a system that produces spectacular hypothetical results, but does not hold up well under real trading. Conversely, overestimating frictional costs may dissuade you from trading an otherwise good system.

It is also worth noting that the more frequently a system trades, the more profound the impact of frictional costs will be.

Minimum Slippage
Applies to Futures only. *Minimum Slippage* is based on a fixed currency value. This parameter works in conjunction with *Slippage Percent* (above).

If set to a non-zero value, this parameter ensures that some slippage cost is assessed against every trade. Trading Blox will impose Minimum Slippage only if the currency value resulting from the Slippage calculation (based on *Slippage Percent*, above) is less than the currency value of slippage as calculated by the *Minimum Slippage* parameter.

Where entry occurs at or near the high or the low of the day, the potential adverse range is practically nonexistent, so the Slippage Percent calculation would be at or near zero. In this case, *Minimum Slippage* can ensure that some slippage is assessed on the trade.

If Slippage Percent is set to zero, then the slippage for all trades will be the value specified by the Minimum Slippage parameter value.

Notes: Can cause fill price to be outside of daily high low range.

Max Percent Market Volume
Max Percent Market Volume determines the maximum trade size based on a percentage of the current volume. For example, if the volume is 200,000 shares and this parameter is set to 2%, this means that the maximum allowed trade size is 2% of 200,000 or 4,000 shares. If an order is placed for 6,000 shares, it will be reduced to 4,000 shares, and an entry will show in the Filtered Trade Log so that you are aware that the order size was reduced. The volume used is the 5-bar exponential moving average of the volume.

Max Margin/Equity to trade
The default is 100%, in that if a requested trade would require more than 100% of available equity in margin or cash, then the trade will be filtered. This parameter allows you to mofidy this behavior. If you enter 50 here, the system will filter trades if the new total margin required would be greater than 50% of the total available equity.

Trade Always on Tick
When this parameter is set to false, the system will trade with maximum precision. So if the system buys Gold on a stop at the moving average, which is 365.44789, then the system will buy at that exact value. Conversely it will exit the trade at an exact value as well. The trade price as listed in the trade details will only show the digits of precision of the instrument, as set in the Futures Dictionary, so the trade price would look like 36.45. But the trade profit would not match that value exactly.

When this parameter is set to true, the system will always trade on the tick. It will round up to the nearest tick for buy orders, and down for sell orders. So in the above example, if you placed an order to buy on stop at 365.44789, the system would place the order to buy at 365.45.

The fill prices would also be on tick, so a slippage of 10% could become more as the order price and then fill price gets pushed to a tick.

Smart Fill Exit (Smart Exit Fills)
When set to true, this function will only fill the exit stop/limit order closest to the open for any particular unit. This can be helpful when using multiple exit blox that place stop/limit orders. If the open is at 10, (and the high is 12 and the low is 7), and you have an exit stop at 11 and a limit order at 8, the stop exit order will be filled.

Entry Day Retracement
This parameter is used to determine if a new position would have been stopped out on the same

day as entry. It can also be used with systems that don't provide a protective exit price with their orders by using a special feature to disable entry-order bar protective order execution when this parameter is set to **"-1"**.

Use Broker Positions
Set to true to insert the broker positions, as entered in the Broker Position Editor, into the simulation. These positions will be entered for both testing and order generation.

Ignore All Test Positions
Set to true to ignore all the test generated positions. Use this when the only positions you want in the test, or order generation, are the broker positions. Use Broker Positions must be true for Ignore All Test Positions to have an effect. Often this is used on the very first day of order generation so that you can start with a clean slate and no open positions. If you want a mix of actual positions and theoretical test positions, then leave this box unchecked.

Use Start Date Stepping Click on topic link for an application example.
Set to true or false to enable or disable the the Increment Test Start option and the Set Test Duration Option.

Increment Test Start (calendar days)
This feature is useful for testing a system's robustness by varying the start date without running multiple tests. This setting will use the initial test start date but will add the number of calendar days to it. For instance, if your original test settings are 2005-01-03 and you enter a value of 3 here, your actual trading will start on 2005-01-06. If you step from 0 to 2, you will get results starting in 2005-01-03, 2005-01-04, 2005-01-05.

Set Test Duration (calendar days)
This setting overrides the original test end date setting. It is useful for testing system robustness by varying the test duration without running multiple tests. If you set this value to 100, every test regardless of start date will be 100 calendar days. Note that the maximum test end date is still the date you entered, so be sure to leave room for both the start date stepping and the test duration so every test is the same length.

Test Start is 1995-01-01 Test End is 2005-01-01
Increment Test Start from 0 to 365 Step 14 (one year step 2 weeks)
Set Test Duration to 3650 (10 years)

The new test end date (the test start plus duration) cannot be beyond the test end date as entered by the user.

Minimum Futures Volume
This parameter applies only to futures. It establishes the minimum daily trading volume, in contracts, required to enter a futures position. It is based on a 5-day exponential moving average of the volume. This value can be viewed by using the instrument.averageVolume property when using the Builder Edition.

Commission per Contract
This parameter indicates the round-trip charge for each futures contract traded. Assuming a *Commission per Contract* of $12.50, buying and then subsequently selling 1,000 contracts of sugar would result in a total transaction cost of ($12.50 per contract) x (1,000 contracts) = $12,500.

Trade Futures on Lock Day
When this parameter is set to false, the simulation will not fill entry or exit orders on days when the high = low, when the trade is in the direction of the lock. The idea is that in futures this could be a lock limit day, and it would be the most conservative assumption to assume you did not get filled.

If you set this parameter to true, you will be filled on these days. This applies to futures only.

Example: With this parameter set to false, and the high equals the low, so it is considered a lock day. If you want to enter long, or exit a short position, and the close of the lock day is less than the close of yesterday, then it will be allowed. But if the close of the lock day is greater than or equal to the close of yesterday, then the fill will be denied.

Account for Contract Rolls
This parameter applies only to futures, and controls whether or not Trading Blox should account for the increased commission and slippage that would have resulted when rolling contracts when a position is held for a long period of time.

If your data includes the Delivery Month of the futures contract being used on any given day in the backadjusted data, then Trading Blox will use this to determine when to roll. It will account for a roll every time the delivery month changes.

If your data does not include the Delivery Month, Trading Blox will estimate when a roll would occur based on the number of trading months listed in the Futures Dictionary. If you actually roll less often than you have months listed, then you should reduce the list to just the roll months, for most accurate results here.

If there are 4 trading months, then Trading Blox will calculate a contract roll every 3 months. If there are 12 trading months, Trading Blox will calculate a contract roll every month. The first simulated roll will occur in 1/2 the normal roll frequency because, on average, the first contract will be entered with 1/2 its trading life left. This process is based on calendar days and works for intraday, daily, weekly, or monthly data.

Each time a simulated roll occurs, Trading Blox accounts for the roll by deducting slippage and commissions for each contract in the position. The Open Equity is moved to Close Equity. If the futures is non-USD denominated, the currency conversion for the roll date will be used to move profit from open equity to closed. So the profit is locked in at the conversion rate of the roll.

Roll Slippage (% of ATR)
This option is available when Account for Contract Rolls is set to true.

The slippage used is the roll slippage percent of the Wilder 20-day ATR (39 day non SMA primed Exponential Moving Average of the True Range). This is an unprimed value, and can be accessed by the instrument.defaultAverageTrueRange property.

Minimum Stock Volume
This parameter applies only to stocks. It establishes the minimum daily trading volume, in shares, required to enter a stock position. It is based on a 5-day exponential moving average of the volume. This value can be viewed by using the instrument.averageVolume property when using the Builder Edition.

Commission per Trade
This parameter indicates the commission charged on a per trade basis. This type of commission charge is used by some brokers for stock brokerage accounts.

Assuming a theoretical *Commission per Trade* of $5.00; Buying 100 shares of XYZ stock would count as one trade, and selling that same 100 shares would be a second trade, for a total transaction cost of $10.00.

Commission per Stock Share
This parameter indicates the round-trip charge for each stock share traded. This can be used in place of or in addition to the *Commission per Trade*. For example, assuming a *Commission per Contract* of $0.02, buying and then selling 1,000 shares of IBM would result in a total transaction cost of ($0.02 per share) x (1,000 contracts) = $20.00. NOTE: Most stock brokerages charge commission for each one-way trade, once for the entry, once for the exit For Trading Blox, use a *Commission per Stock Share* of double the per share commission charged by your broker for one-way trades.

Commission by Stock Value
Uses a percent of the stock value as the commission amount.

Convert Profit by Stock Splits
With stocks, when the data is backadjusted for stock splits, the profits made/lost are respectively smaller as the prices get smaller. Take MSFT as an example. The close on Jan 2, 1987 was $47.75, and the close on Feb 2 was $73.00. But your backadjusted data will probably show around $.17 on Jan 2 and $.25 on Feb 2. So if you bought 100 shares at the close on Jan 2 and sold at the close on Feb 3, you would have made $25.25 per share, or $2,525 total. But many simulations would show a profit of just $.08 per share, or $8 total.

If you set this parameter to true, Trading Blox will determine that the real profit for this trade was $25.25 per share. Your simulation will then be accurate even though the data has been adjusted.

Trading Blox will convert the profit on each trade by the stock split ratio between the trade entry and the trade exit, and it will also convert the volume in the data file back to the real volume amount using the stock split ratio. This volume number is only used for the AverageVolume property, which is used for the Max Percent Volume per Trade and Minimum Stock Volume global parameters. You can also access these properties as instrument.adjustedVolume, and instrument.averageVolume.

The stock split ratio is the unadjusted close / close for any given day, so be sure to include the unadjusted close in the data.

Create two MFST files in CSI, one with no adjustements, and one with stock split adjustments. If you look at the volume on 2005-01-03 for MSFT using the unadjusted data, you will see that the volume presented is 39,545,600. (Note that there is a volume multiplier of 100 in play here as well, so the number you see is actually 395456.) The stock split ratio on that day is 16, so the actual volume traded was 2,471,600. If you are trading a max percent of the day's volume, then you would want to use the adjusted volume number rather than the unadjusted number that is provided in the data file.

Likewise, if you use the example above, you want to make sure the profit on a 100 share trade reflects the actual value of that trade in 'todays' dollars. So Trading Blox converts the profit using the stock split ratio as of the trade entry.

In addition, Trading Blox needs to know when the stock split was, in relation to the dividend distribution.

So if you purchased 100 shares, the stock had a 2x stock split, and then a $.50/share dividend, that is a different profit than if you purchased 100 shares, the stock had a $.50/share dividend, and then it had a 2x stock split.

This is why it's important to keep the stock splits and dividends separate. The stock splits alter the data in a geometric manner, whereas the dividends are simple unadjusted per share amounts. If you try to stuff proportionally adjusted dividend data into a stock split adjusted series, you will get the wrong P&L everytime.

So, set your CSI UA to adjust the stock data by stock splits only, be sure to include the unadjusted close, and call CSI to get the unencrypted dividend data so Trading Blox can build those files for you. CSI will provide the unencrypted dividend data to all Trading Blox customers free of charge.

Pay Dividends
Set to true to account for dividends in the test. The dividend files should be setup in the Dividends folder. The dividend file suffix can be changed in the INI file.

Pay Margin on Stocks
This parameter is for stocks only, and defines whether the system will charge margin for excess cash used when buying stocks. The values used are from the system wide base currency rate files. In the case of USD, the file is the USD_Rates.txt file in the forex data directory. The Borrow Rate is used when accounting for margin.

Margin Interest is charged every day when available cash falls below zero.

Forex Trade Size
The inter-bank forex market trades in units of 100,000 base currency. Many retail forex brokers allow trading units of smaller size. This parameter controls the unit size to allow for simulation of trading through retail forex brokers.

Account for Forex Carry
This determines whether or not to account for Forex Carry charges (see Forex Carry Calculations for details).

Use Pip-Based Slippage
Since many retail Forex brokers have fixed spreads per market and guarantee "zero slippage" fills, Trading Blox provides for this using a fixed per-market slippage which is defined by the spread in pips for each market. The spread for each market should be set to the figures your Forex broker uses or a larger number for more conservative testing. See Forex Dictionary for information on changing the Forex market spreads.

When set to TRUE, the parameter *Use Pip-Based Slippage* overrides the normal Trading Bloxslippage calculations (based on a fixed amount or percentage) for the Forex markets only.

Thread Count
The thread count is the number of simultaneous processes that will be run for data loading and testing. When running a large stepped test simulation, the thread count is the number of tests that will be run simultaneously. Data loading will also be done in parallel. This number is based on the number of cores in your computer, and hyperthreading. A quad core computer will get the fastest speeds for

large tests by setting this number to 8, whereas a six-core hyperthreaded computer could use 12 threads at the same time for max speed. Note that there is some setup time for each thread, so a small test might take longer with many threads enabled.

The thread count can be between 1 and the max number allowed by your license. The turtle edition allows for one thread by default, and the pro and build editions allow 2 by default. To purchase additional threads for faster testing see the Customer Login section of the website.

Thread Count is a global parameter specific to each suite, so suites can have different thread counts depending on what they are doing. Simulation Scoped variables cannot be used in a multi threaded environment because the tests are all being processed in parallel, so the Walk Forward Suite as an example would need the Thread Count set to 1.

If using Simulation Scoped variables consider whether they are really required or not. Exporting data to files, or using the Registry with SetRegistryValue, are additional work arounds. If Simulation Scoped variables are required, then set the Thread Count to 1.

Walk Forward Optimization Size

This sets the number of calendar days used for the Walk Forward optimization process when running a Walk Forward test.

Walk Forward Out of Sample Size

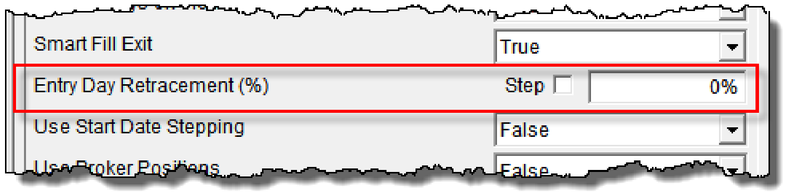
This sets the number of calendar days used for the Walk Forward out of sample process when running a Walk Forward test.

See also the tradingblox.ini file for more settings options.

2.1 Entry Day Retracement

This parameter is used to determine if a new position would have been stopped out on the same day as entry. It can also be used with systems that don't provide a protective exit price with their orders by using a special feature to disable entry-order bar protective order execution when this parameter is set to `"-1"`.

Entry Day Retracement Calculations:
An Entry Day Retracement percent is created by multiplying the price-bar's Range, that is the distance between the high to the low price of the price-bar, to obtain a point estimate.



Smart Fill Exit	True
Entry Day Retracement (%)	Step ☐ 0%
Use Start Date Stepping	False
Use Broker Positions	False

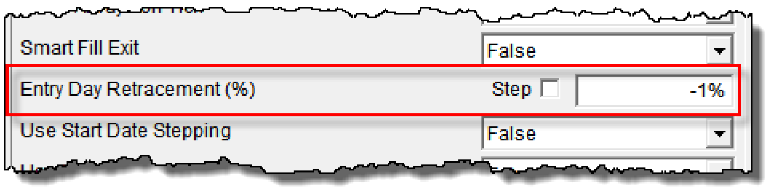
This point-estimate is then added, or subtracted from the order fill price to determine if the percentage applied would execute the protective order price so that a retracement price can be determined. To trigger a protective price the retracement price must be beyond the protective entry-order protective price to enable a same-day exit from the position.

Example:

```
For long entries:
   Exit-Trigger-Price = Order-Fill-Price - ( High - Low ) * Retracement %

For short entries:
   Exit-Trigger-Price = Order-Fill-Price + ( High - Low ) * Retracement %
```

Retracement Percent (%) Rules:

- An Exit-Trigger-Price is never more than the High, or less than the Low prices of the entry-bar. An Exit-Trigger-Price is never more than the close for long entries, or less than the close for short entries.
- A Retracement **%** of zero is the default setting, and that behavior is used as outlined below for an aggressive approach description. A Retracement **%** of **100** will use the extreme of the day for a conservative approach.
- If the position was filled during the day (stop/limit orders), and the close is less than the stop for long entries or more than the stop for short entries, then the position is stopped out
- If the position was filled on the open of the day, and the low is less than the stop for long entries or the high is more than the stop for short entries, then the position is stopped out
- An Entry Day Retracement of less than 0, such as -1, will disable the entry day stop. This can be useful if you want to use stops for risk measurement purposes, but not actually place those stops into the market on the entry day.


Smart Fill Exit — False

Entry Day Retracement (%) — Step ☐ — -1%

Use Start Date Stepping — False

Note:

When Trading Blox triggers a protective exit price order using the Entry Day Retracement process on the bar of entry the reference ID for that order will be zero.

2.2 Reducing Optimization Time

Quicker Optimizations in Trading Blox 4 (Web Link)

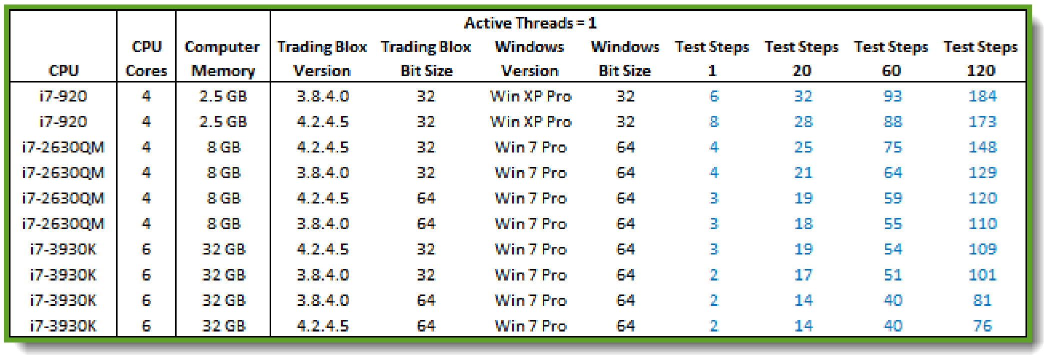
Previous Trading Blox versions executed stepped optimizations one test step at a time. Versions 3.x and earlier executed a stepped optimization in a serial 1-step at a time sequence. Serial step testing requires each test step to complete before the next step begins testing. Processing optimization steps serially adds each step's test time to the amount of time the optimization consumes to complete all test steps.

Version 4 changes how optimizations are executed by putting each test step into a thread. Two threads are available when Trading Blox 4 is first installed, and more threads are available as an option. Each thread can test 1-step at a time, and multiple threads can test other steps in that same test period. With 2-threads always available in version 4 the time it will take to perform an optimization in version 3.x is about twice the amount of time it will take in version 4.x, assuming the same computer specifications and test suite.

Multiple threads can save time because each thread running a step test provides an independent area where calculations can be tested in parallel. By running 2-threads at the same time the amount of time 2-threads requires compared to how a single thread test is nearly have the time of a the single thread time.

To put this in more concrete terms, the following table shows the amount of time that four different optimized test needed to run using 1-thread. All four tests were executed on three different computers. One computer used both the 32-Bit versions of Trading Blox 3.8.4, and Trading Blox 4.2.4.5. Two computers supporting 64-Bit software tested both 32-Bit & 64-Bit version 3 and version 4.

Test duration details in this table represent how many steps were in the four tests and the number of seconds each test required. All tests in this table were limited to 1-thread. Versions prior to version 4.x were only able to use one thread tests, so that test limitation will be the baseline to show how additional threads help to reduce stepped test times.

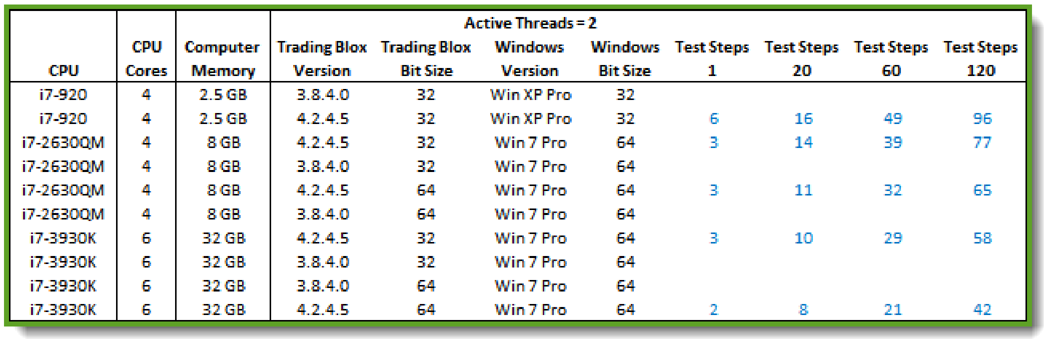


			Active Threads = 1							
CPU	CPU Cores	Computer Memory	Trading Blox Version	Trading Blox Bit Size	Windows Version	Windows Bit Size	Test Steps 1	Test Steps 20	Test Steps 60	Test Steps 120
i7-920	4	2.5 GB	3.8.4.0	32	Win XP Pro	32	6	32	93	184
i7-920	4	2.5 GB	4.2.4.5	32	Win XP Pro	32	8	28	88	173
i7-2630QM	4	8 GB	4.2.4.5	32	Win 7 Pro	64	4	25	75	148
i7-2630QM	4	8 GB	3.8.4.0	32	Win 7 Pro	64	4	21	64	129
i7-2630QM	4	8 GB	4.2.4.5	64	Win 7 Pro	64	3	19	59	120
i7-2630QM	4	8 GB	3.8.4.0	64	Win 7 Pro	64	3	18	55	110
i7-3930K	6	32 GB	4.2.4.5	32	Win 7 Pro	64	3	19	54	109
i7-3930K	6	32 GB	3.8.4.0	32	Win 7 Pro	64	2	17	51	101
i7-3930K	6	32 GB	3.8.4.0	64	Win 7 Pro	64	2	14	40	81
i7-3930K	6	32 GB	4.2.4.5	64	Win 7 Pro	64	2	14	40	76

Trading Blox 3 & 4.x - 1-Thread Active Stepped Optimization Test Time Results

Active Threads:
Being able to do an intensive processing required for good system development quickly is a significant enhancement in our trading platform. It is significant because it allows us to see the results of more ideas in less time, and having a second thread provides us with a significant time reduction from what was possible in version 3.

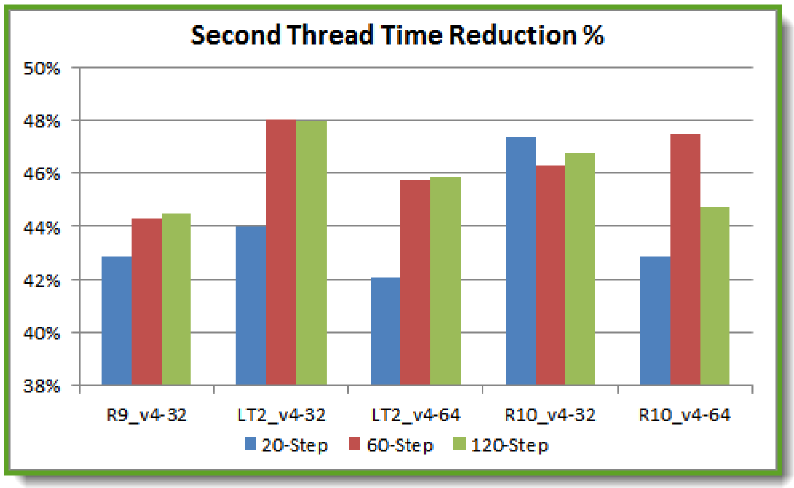
In this next table 2-threads are used to perform the same test as shown above, however only data for version 4 is available because version 3 cannot support a second thread to improve performance, so we need to accept the single thread times are the best older versions will be able to provide.



			Active Threads = 2							
CPU	CPU Cores	Computer Memory	Trading Blox Version	Trading Blox Bit Size	Windows Version	Windows Bit Size	Test Steps 1	Test Steps 20	Test Steps 60	Test Steps 120
i7-920	4	2.5 GB	3.8.4.0	32	Win XP Pro	32				
i7-920	4	2.5 GB	4.2.4.5	32	Win XP Pro	32	6	16	49	96
i7-2630QM	4	8 GB	4.2.4.5	32	Win 7 Pro	64	3	14	39	77
i7-2630QM	4	8 GB	3.8.4.0	32	Win 7 Pro	64				
i7-2630QM	4	8 GB	4.2.4.5	64	Win 7 Pro	64	3	11	32	65
i7-2630QM	4	8 GB	3.8.4.0	64	Win 7 Pro	64				
i7-3930K	6	32 GB	4.2.4.5	32	Win 7 Pro	64	3	10	29	58
i7-3930K	6	32 GB	3.8.4.0	32	Win 7 Pro	64				
i7-3930K	6	32 GB	3.8.4.0	64	Win 7 Pro	64				
i7-3930K	6	32 GB	4.2.4.5	64	Win 7 Pro	64	2	8	21	42

Trading Blox 4.x - Default 2-Thread Stepped Optimization Test Time Results

Time saved versus the time needed for a 2-thread test in Trading Blox 4 versus the time it takes with 1-threads creates a significant time reduction across all three optimizations:

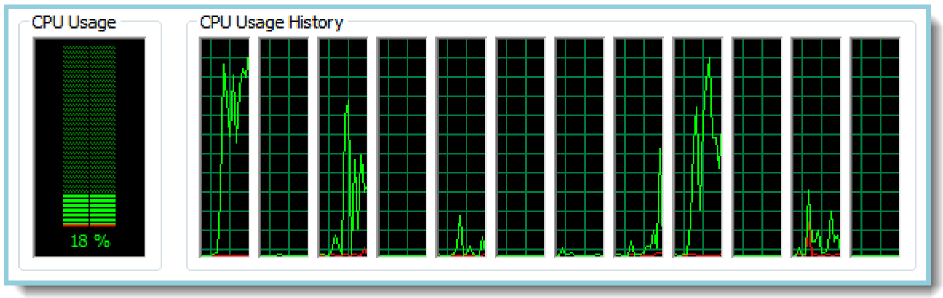


3-Stepped Optimization Time Reduction | 2-Threads versus 1-Thread Testing

A second thread run in parallel for optimization test is a noticeable improvement, and it also asks the question of whether two threads are enough. My answer is no, two threads are not enough. It is no because our test ranged from 20 to 120 steps is a fairly small optimization test, but it is a practical size for careful data collection, and it shows how ineffectively we are using the computer's capacity to get work done.

Computer's capability should be a factor in making the decision about how many threads are enough; 2-threads on multi-core computers will only generate a low utilization. It is low because a single core limits 2-threads to process the entire test. This means all the other cores won't be helping to reduce test times.

This next image shows the six-core computer running our 120-step test. Windows Task Manager's performance tab shows this thread limitation only being able to demonstrate an 18% utilization rate, leaving the other 82% of the computer's capability at idle.

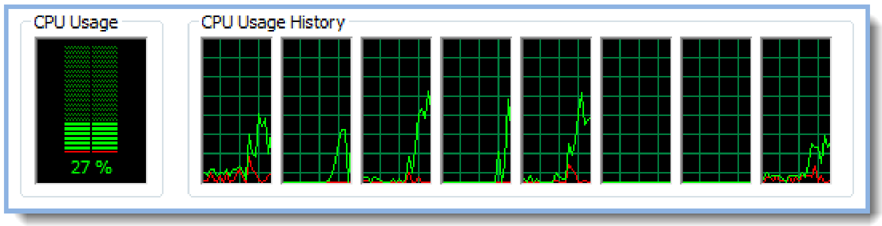


R10, TB 4, 120-Step Test, 2-Thread Setting, 6-Core CPU

For reference Window's Task Manager Performance Tab gives us a good estimation of the CPU Usage based on how much of the computer's design capacity is available.

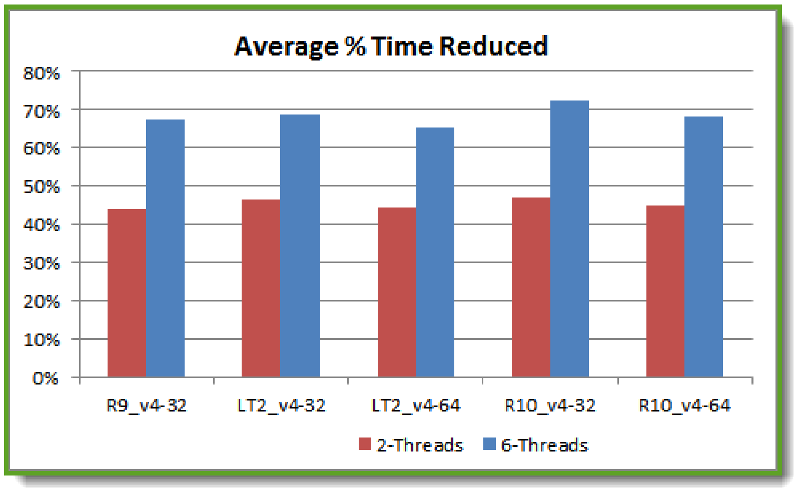
Each in these CPU Usage images the narrow vertical windows represents the number of threads the CPU was designed to handle at the same time. In this case there are 6-physical CPU cores in the chip, with each core capable of managing two threads, for a total of 12-threads available by design.

In this next image of a quad-core CPU the utilization running the same test creates an estimate of 27% utilization leaving 73% of the CPU's ability to sit at idle.



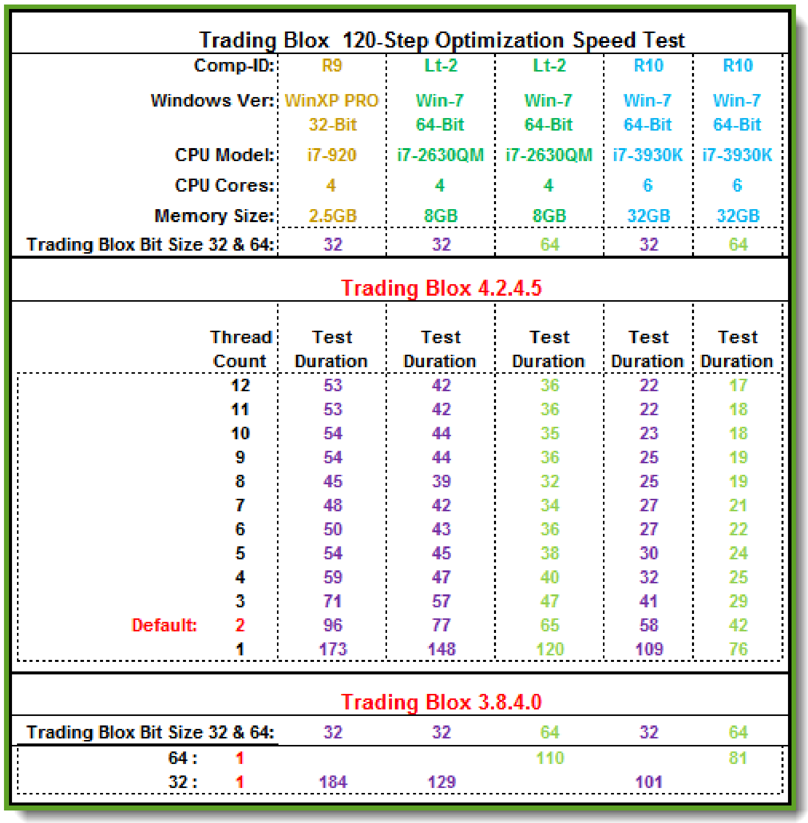
LT2, TB 4, 120-Step Test, 2-Thread Setting, Quad-Core CPU

Being able to support 12-threads, or 8-threads for a quad-core CPU, doesn't mean all the cores should be loaded at 100% because that would interfere with the user's need to do other things while the computer was running optimization test. More threads up to the design limit will improve utilization as long as the operating system and the computer's memory installation are capable of supporting all the threads.



Avg. 3-Stepped Test Time Reduction Percentage | 2-Threads versus 6-Threads Optimizations

In this next thread count test results table image a full range the same our 120-Step test using 1-thread up to 12-threads on each of the computers was executed on all three computers. Test results details for the other steps lengths are available on request.



Trading Blox 120-Step Optimization Speed Test

Comp-ID:		R9	Lt-2	Lt-2	R10	R10
Windows Ver:		WinXP PRO 32-Bit	Win-7 64-Bit	Win-7 64-Bit	Win-7 64-Bit	Win-7 64-Bit
CPU Model:		i7-920	i7-2630QM	i7-2630QM	i7-3930K	i7-3930K
CPU Cores:		4	4	4	6	6
Memory Size:		2.5GB	8GB	8GB	32GB	32GB
Trading Blox Bit Size 32 & 64:		32	32	64	32	64
Trading Blox 4.2.4.5						
	Thread Count	Test Duration	Test Duration	Test Duration	Test Duration	Test Duration
	12	53	42	36	22	17
	11	53	42	36	22	18
	10	54	44	35	23	18
	9	54	44	36	25	19
	8	45	39	32	25	19
	7	48	42	34	27	21
	6	50	43	36	27	22
	5	54	45	38	30	24
	4	59	47	40	32	25
	3	71	57	47	41	29
Default:	2	96	77	65	58	42
	1	173	148	120	109	76
Trading Blox 3.8.4.0						
Trading Blox Bit Size 32 & 64:		32	32	64	32	64
64 :	1			110		81
32 :	1	184	129		101	

Trading Blox 3 & 4 Available Thread Stepped Optimization Test Time Results

This table shows that 4-core computers performance improvement slows significantly when Trading Blox is sets to allow 6 of the possible 8 threads a quad-core CPU will support. On a 6-core computer the number of thread time reduction differences slows when there are 8 to 9 threads operating in memory. If these two thread to cores ratio is indicative, a Dual-Core CPU should easily handle 3-threads, and a single core CPU will be very busy with 2-threads.

Thread counts up to the thread-count design limit can give the best time, but if you try to do something else while a stepped simulation is executing the CPU will need to do some memory or thread swapping operations that will slow everything down. Memory swapping doesn't have a significant time reduction impact on performance, but disk swapping will reduce performance significantly.

Having enough memory to keep the all the available threads in memory throughout the test step testing time is critical to being able to keep the CPU from swapping threads to disk like the results of the R9 computer demonstrates because of Windows XP Pro max working memory size limitation. That memory limitations causes Windows XP Pro results to run slower than its less capable LT2 computer because XP is doing a lot of thread to memory swapping.

Computer Memory Requirements:
Stepped optimizations execute at their fastest rate when the entire optimization test is able to execute in memory without Windows needing to use disk space to support any part of the optimization test.

This means each computer's hardware configuration must have enough memory to handle the loading of all the instrument files into memory, and still have room to allocate dedicated memory to each of the threads active during the stepped simulation. It also means the large simulations using a large portfolio of stocks over a long period of time might not find enough memory to support the planned simulation without having to do a lot of memory to disk space swapping.

When Windows begins to swap memory space to disk so that it can execute other threads before finishing the current thread's processing, the amount of time for the simulation becomes longer than would happen had the computer provided more computer memory for the simulation.

If you find your simulation running slower than expected, check the Windows Task Manager Performance tab to see how much memory is being used. If the Task Manager shows a high percentage of memory is being used chances are Windows is doing a lot of memory to disk swapping that might be prevented if the computer can support a larger capacity of memory.

While there are no simple rules for determining how much memory is needed, most optimization test with Futures data can be easily be executed with 6 GB of memory. Optimization test on large Stock portfolios need more memory, but a capacity of 12 GB should be enough most of the time.

Stepped Optimization test Guidelines:

- Short period test require less memory than long period test.
- Small portfolios require less memory than large portfolios.
- Multiple system optimization tests require more memory than a single system optimization test.

- In process custom data output disk activities require less time and memory than optimization that don't require custom data output operations.
- Small optimization steps create a larger processing need and a longer period of time to complete.
- Active threads that represent between 65 to 75% of CPU supported threads will allow for a high CPU utilization without making other processing needs like email and web browsing stall.

Thread Testing Caution:
Keep in mind how a threads operate in parallel time. When a Stepped optimization is testing the First-Thread (Thread-1), the computer is also testing the Second-Thread (Thread-2) at nearly the same start time. This means if Thread-2, or any other thread test is dependent upon the completion of an earlier thread to get or report information, there will need to be a reference in your source code using the **test.threadIndex** property to ensure that the timing of the data that is need is passed when it is needed.

Here is what the test warning that appears in the main screen's Log Area reports. This warning will always appear when a **Simulation Scoped** variable is used. A Report for each of the Simulation scoped variables is given:

```
Using Simulation Scoped variable <your-variable name1> in multi-threaded test
could have unintended consequences. Use with caution.

Using Simulation Scoped variable <your-variable name2> in multi-threaded test
could have unintended consequences. Use with caution.
```

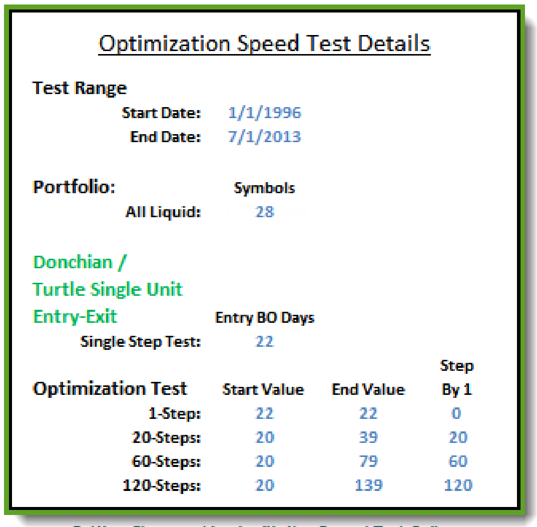
Speed Test Rules: (Web Link)

Trading Blox original optimization “Original Speed Test” was created in January, 2008 to provide a fixed set of data that we could use with our Trading Blox installation. From that test result comparison we would be able to see how our local performance fit into the overall range of results being reported.

When version 3.4.2 was released the "Second Speed Test" date range was increased to include more data, and test result times showed Trading Blox improved.

Now version 4.x has the ability to add multiple threads so it can do more in the same amount of time. With Trading Blox doing things even faster we needed to increase the number of testing steps and lengthen data test range again. Both changes in the new Speed Test requirements improve the resolution of time differences between threads. It also helps us get a better understanding of where the CPU utilization begins to make smaller time improvements as the thread count increases.

This table shows the actual test details used to generate the top posting:



Optimization Speed Test Details

Test Range

Start Date:	1/1/1996		
End Date:	7/1/2013		

Portfolio:	Symbols
All Liquid:	28

Donchian / Turtle Single Unit Entry-Exit	Entry BO Days
Single Step Test:	22

Optimization Test	Start Value	End Value	Step By 1
1-Step:	22	22	0
20-Steps:	20	39	20
60-Steps:	20	79	60
120-Steps:	20	139	120

Setting Changes Used with the Speed Test Suites

Trader's RoundTable Blox Files: Speed Test Modules.zip

Modules in the zip files were exported by Trading Blox 4 using the Suite's Export option. Each Suite will import in Trading Blox 4 using the Suite's Import feature. Installing files into Trading Blox 3.x will require the files to be copied from their respective folders and placed into the same folders in your Trading Blox directory. Trading Blox 3.x should be closed while importing so the start-up logic finds them for display.

To import the SpeedTest's ZIP files contained in the above file, you should remove them from the above file so Trading Blox will find the folders it needs to move the modules.

When you unpack the above Zip file you should see this list of file names:

- **`All Liquid - Orig.set`** - **Futures Portfolio File List**
- **`Export Speed Test Def Sim.zip`** - **Single Step Test**
- **`Export Speed Test Step 020 Sim.zip`** - **20-Step Test**
- **`Export Speed Test Step 060 Sim.zip`** - **60-Step Test**
- **`Export Speed Test Step 120 Sim.zip`** - **120-Step Test**

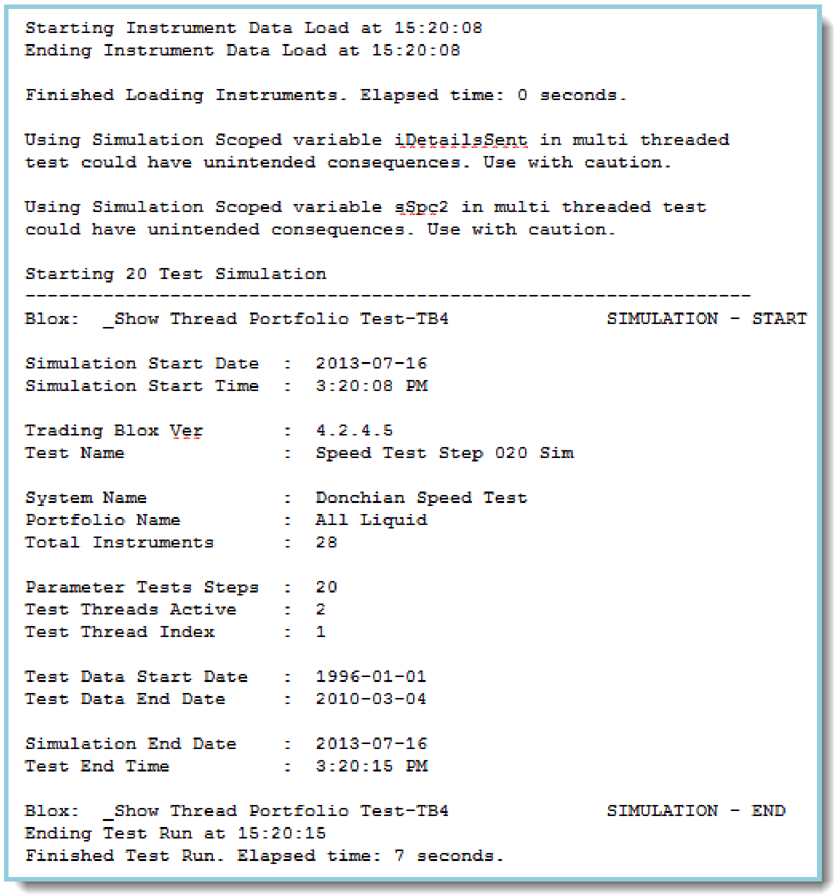
- **`_Show Thread Portfolio Test-TB3.tbx`** - **Log Window Reporting Blox TB 3.x Only**

This last Blox Module listed, "**_Show Thread Portfolio Test-TB3.tbx**", is a modified version of a similar module used with Trading Blox 4 included in the Suite packages shown above. All the modules in the Speed Test Packages will work with Trading Blox 3 except for the Auxiliary module that is used for Log Window Reporting.

For testing in Trading Blox 3, remove the module: "**_Show Thread Portfolio Test-TB4.tbx**" from all the System Lists and replace it with "**_Show Thread Portfolio Test-TB3.tbx**" Auxiliary Blox. Thread properties in the TB4 blox version are not available in TB3 and will cause TB3 to report an error if the Auxiliary module isn't replaced.

Before testing be sure to check the portfolio symbols in the All Liquid - Orig.set so that they show the correct Futures data file will be used.

Testing is best performed with Log Window on Trading Blox's main screen open. Data from each will be generated it will be displayed log window where it can be copied. Log Window output will look like this:
[img] Sample_Test-Output_Log_Window_Example.png [/img]



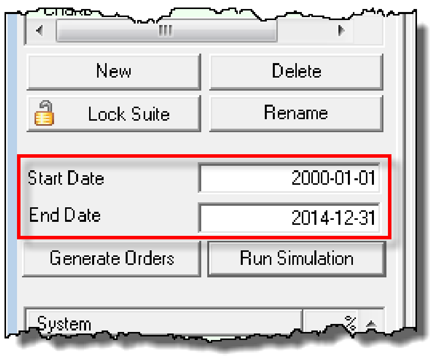
```
Starting Instrument Data Load at 15:20:08
Ending Instrument Data Load at 15:20:08

Finished Loading Instruments. Elapsed time: 0 seconds.

Using Simulation Scoped variable iDetailsSent in multi threaded
test could have unintended consequences. Use with caution.

Using Simulation Scoped variable sSpc2 in multi threaded test
could have unintended consequences. Use with caution.

Starting 20 Test Simulation
-----------------------------------------------------------------
Blox:  _Show Thread Portfolio Test-TB4             SIMULATION - START

Simulation Start Date  :  2013-07-16
Simulation Start Time  :  3:20:08 PM

Trading Blox Ver       :  4.2.4.5
Test Name              :  Speed Test Step 020 Sim

System Name            :  Donchian Speed Test
Portfolio Name         :  All Liquid
Total Instruments      :  28

Parameter Tests Steps  :  20
Test Threads Active    :  2
Test Thread Index      :  1

Test Data Start Date   :  1996-01-01
Test Data End Date     :  2010-03-04

Simulation End Date    :  2013-07-16
Test End Time          :  3:20:15 PM

Blox:  _Show Thread Portfolio Test-TB4             SIMULATION - END
Ending Test Run at 15:20:15
Finished Test Run. Elapsed time: 7 seconds.
```

Sample Speed Test Log Window Output Example

Test result generated on your computer should help you get a relative understanding of the time differences possible on your computer.

2.3 Start Date Stepping

System testing is most often performed over a stationary date range where the test results reflect the entire range of calendar days available starting on the Start-Date and ending on the End-Date. This date range is established on the main screen in these two parameter fields:



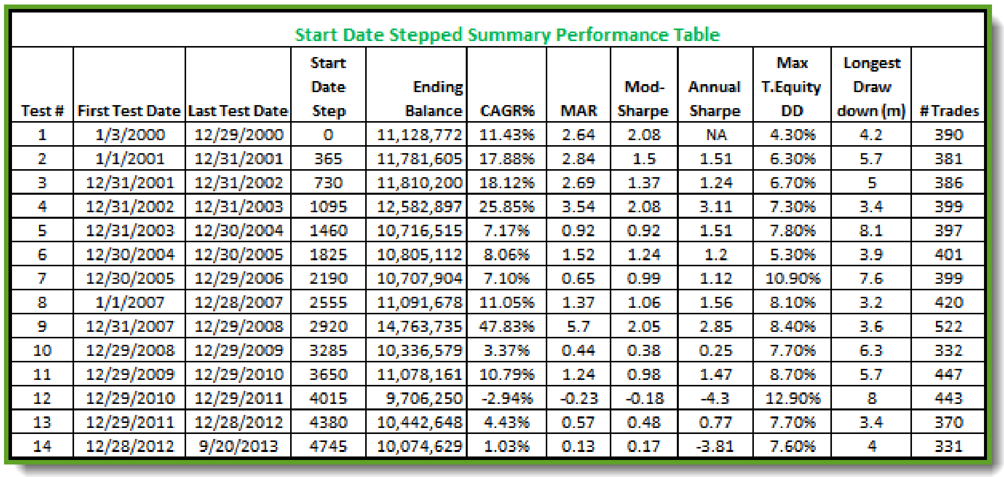
Start Date 2000-01-01
End Date 2014-12-31

Global Parameter Settings:
All standard testing where Start Date Stepping is not enabled, the test will use the date entered on the main screen for all of the instruments in all of the systems contained in the test suite.

Start Date Stepping is designed to provide a series of segmented test between the start and ending dates so as to provide the user with the ability to see a series of results that can be used as a distribution of performance values experienced during each of the segment periods created between the start and end of test dates.

This table shows a series of annual results created with Start Date Stepping so we would be able to see annual test results for all of the years in the test period range. By creating smaller test segments within a larger test period Trading Blox is able to generate individual segment test results.

Segmented test results help us understand the stability or variability of system performance over the entire test range. When the results are fairly uniform and pleasing over the entire test range, it is likely a reflection of a robust and stable system design.

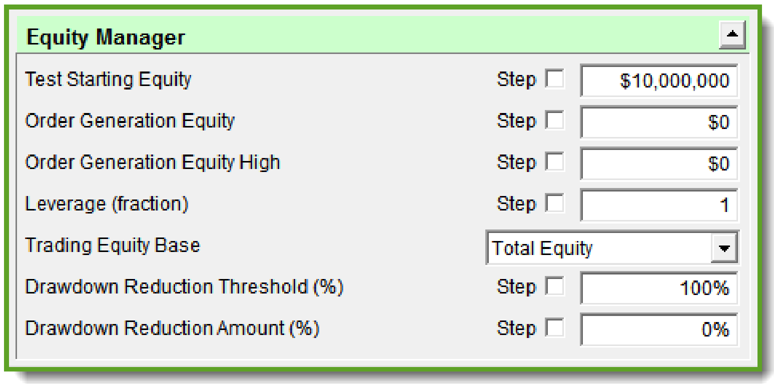


Start Date Stepped Summary Performance Table

Test #	First Test Date	Last Test Date	Start Date Step	Ending Balance	CAGR%	MAR	Mod-Sharpe	Annual Sharpe	Max T.Equity DD	Longest Draw down (m)	#Trades
1	1/3/2000	12/29/2000	0	11,128,772	11.43%	2.64	2.08	NA	4.30%	4.2	390
2	1/1/2001	12/31/2001	365	11,781,605	17.88%	2.84	1.5	1.51	6.30%	5.7	381
3	12/31/2001	12/31/2002	730	11,810,200	18.12%	2.69	1.37	1.24	6.70%	5	386
4	12/31/2002	12/31/2003	1095	12,582,897	25.85%	3.54	2.08	3.11	7.30%	3.4	399
5	12/31/2003	12/30/2004	1460	10,716,515	7.17%	0.92	0.92	1.51	7.80%	8.1	397
6	12/30/2004	12/30/2005	1825	10,805,112	8.06%	1.52	1.24	1.2	5.30%	3.9	401
7	12/30/2005	12/29/2006	2190	10,707,904	7.10%	0.65	0.99	1.12	10.90%	7.6	399
8	1/1/2007	12/28/2007	2555	11,091,678	11.05%	1.37	1.06	1.56	8.10%	3.2	420
9	12/31/2007	12/29/2008	2920	14,763,735	47.83%	5.7	2.05	2.85	8.40%	3.6	522
10	12/29/2008	12/29/2009	3285	10,336,579	3.37%	0.44	0.38	0.25	7.70%	6.3	332
11	12/29/2009	12/29/2010	3650	11,078,161	10.79%	1.24	0.98	1.47	8.70%	5.7	447
12	12/29/2010	12/29/2011	4015	9,706,250	-2.94%	-0.23	-0.18	-4.3	12.90%	8	443
13	12/29/2011	12/28/2012	4380	10,442,648	4.43%	0.57	0.48	0.77	7.70%	3.4	370
14	12/28/2012	9/20/2013	4745	10,074,629	1.03%	0.13	0.17	-3.81	7.60%	4	331

Any size and number of test segments can be created. Our results table is restricted to an annual test period range, but longer or shorter test periods generate more or lest segmented test results.

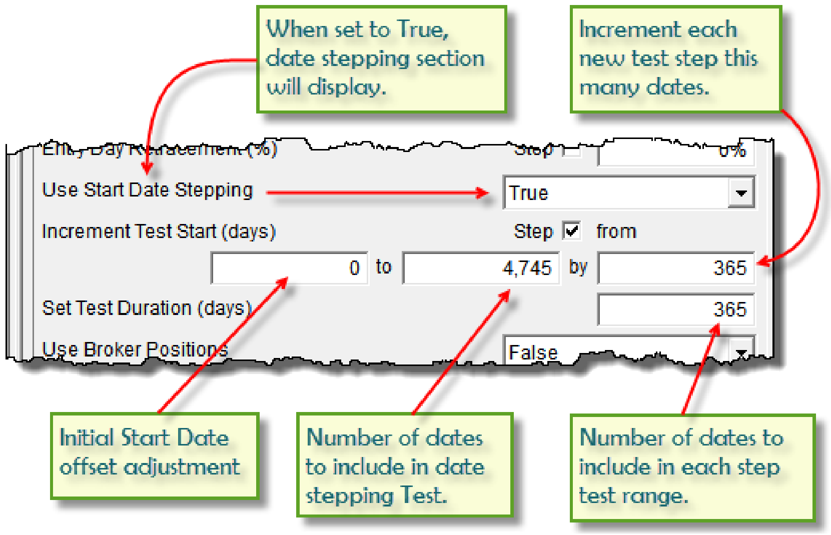
Starting Test Equity:
Start date stepping resets each test segment to the values established in the **Equity Manager** area at the bottom of the **Global Parameter** section. Ending Balance column in the table above shows the ending equity value at the end of each test segment.



Equity Manager

Test Starting Equity	Step ☐	$10,000,000
Order Generation Equity	Step ☐	$0
Order Generation Equity High	Step ☐	$0
Leverage (fraction)	Step ☐	1
Trading Equity Base		Total Equity
Drawdown Reduction Threshold (%)	Step ☐	100%
Drawdown Reduction Amount (%)	Step ☐	0%

Creating Test Period Segments:
Settings to generated the table above used the **Starting Equity** shown in the "**Test Starting Equity**" parameter image, and in the values shown in the stepped fields area shown next:


When set to True, date stepping section will display.
Increment each new test step this many dates.
Initial Start Date offset adjustment
Number of dates to include in date stepping Test.
Number of dates to include in each step test range.

Use Start Date Stepping: True
Increment Test Start (days): Step ☑ from 0 to 4,745 by 365
Set Test Duration (days): 365
Use Broker Positions: False

Start date stepping uses calendar dates.

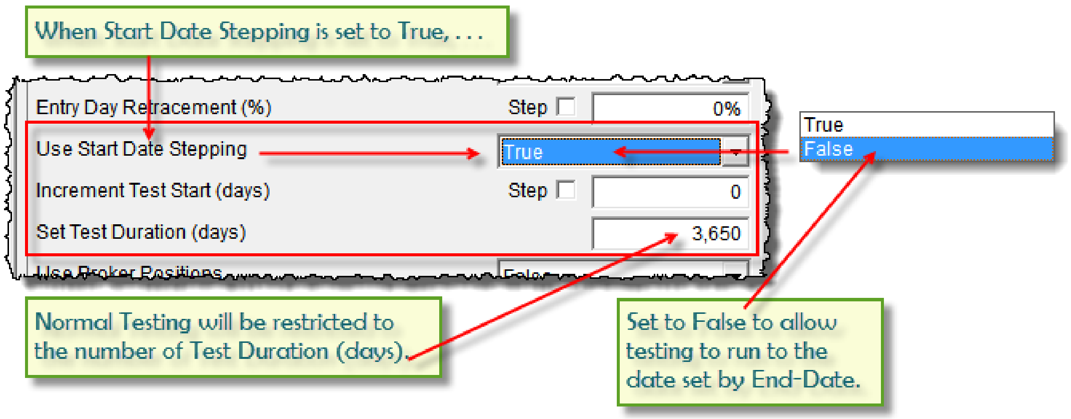
Values shown adjust the starting offset of the period length of each test segment. As each test segments completes the incrementing field value adds "**by**" field value to the start date to define so the next test period moves ahead the fields calendar date count for when the next test segment will begin.

Each test-step will create a new test segment until the number of dates in the "to" parameter have been consumed, or the test End-Date is reached.

Note:
Date stepping does not end creating segment records until the number count of dates have been used. Because Start Date Stepping segments don't end, it is important to know when the last date is reached so the segment records displaying "**+infinity%**" in the generated table are not included. Records where "**+infinity%**" is displayed in any of the reported the values should be ignored so you don't use that information to understand the system's abilities. If this isn't clear, run a test with

a value of calendar dates that are more than what are available so you can see which records are just segment records created to consume the number of dates entered.

Start Date Stepping Warning:
When the **Global Parameter** "**Use Start Date Stepping**" is set to True, the "**Set Test Duraction (days)**" parameter will control how long a normal simulation will execute.

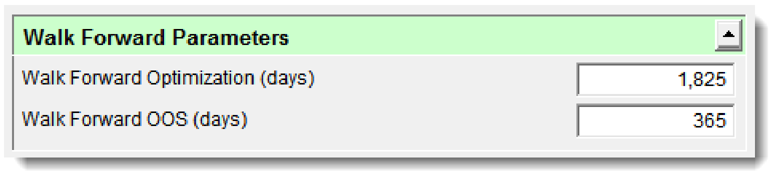

When Start Date Stepping is set to True, . . .
Entry Day Retracement (%)
Step
0%
Use Start Date Stepping
True
True
False
Increment Test Start (days)
Step
0
Set Test Duration (days)
3,650
Use Broker Positions
Normal Testing will be restricted to the number of Test Duration (days).
Set to False to allow testing to run to the date set by End-Date.

This means that if the number of dates entered into the "**Set Test Duration (days)**" field is less than the number of dates available, testing will end when the number of dates between the test **Start-Date** and the number of dates shown in the "**Set Test Duration (days)**" field have been reached the remaining dates available to the End-Date value and included in the instruments in the portfolio, will not be tested.

While testing will stop, and any custom scripted indicators will stop displaying, the built-in indicators created in the **Indicator** section will be displayed.

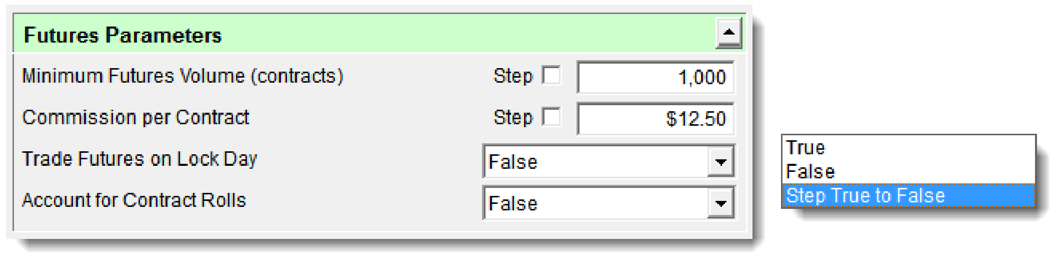
Above images shows that when `3,650` days have been processed by a standard test where the **Test-End** date would have allowed a normal test to run up until the **Test-End** date, script execution will stop executing at "**Set Test Duration (days)**" end of days allowed.

Section 3 – Walk Forward Parameters



Walk Forward Optimization (days)	1,825
Walk Forward OOS (days)	365

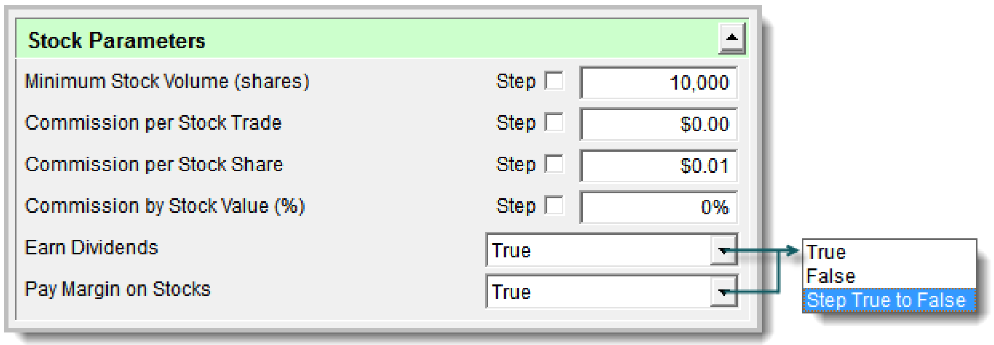
Section 4 – Futures Parameters



Futures Parameters

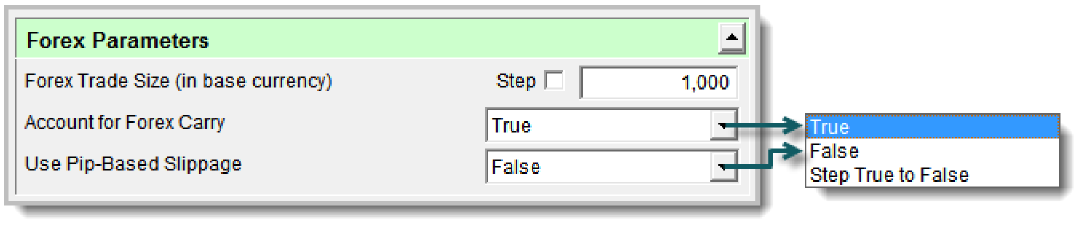
Minimum Futures Volume (contracts)	Step ☐	1,000
Commission per Contract	Step ☐	$12.50
Trade Futures on Lock Day		False
Account for Contract Rolls		False

Section 5 – Stock Parameters



Stock Parameters		
Minimum Stock Volume (shares)	Step ☐	10,000
Commission per Stock Trade	Step ☐	$0.00
Commission per Stock Share	Step ☐	$0.01
Commission by Stock Value (%)	Step ☐	0%
Earn Dividends		True
Pay Margin on Stocks		True

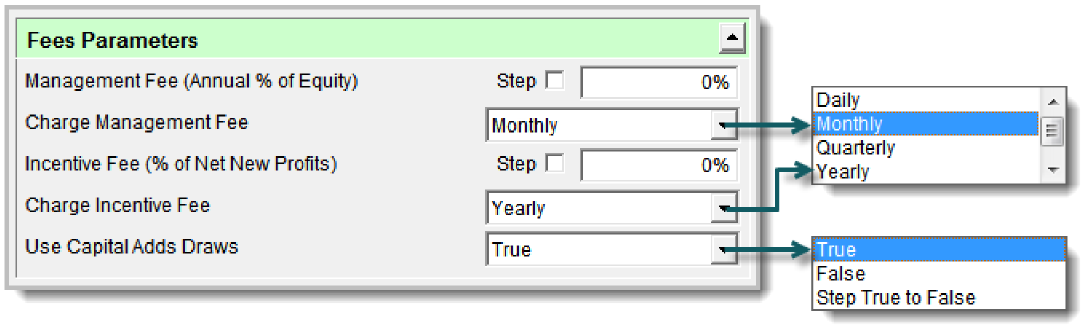
Section 6 – Forex Parameters



Forex Parameters

Forex Trade Size (in base currency)	Step ☐	1,000
Account for Forex Carry		True
Use Pip-Based Slippage		False

Section 7 – Fees Parameters

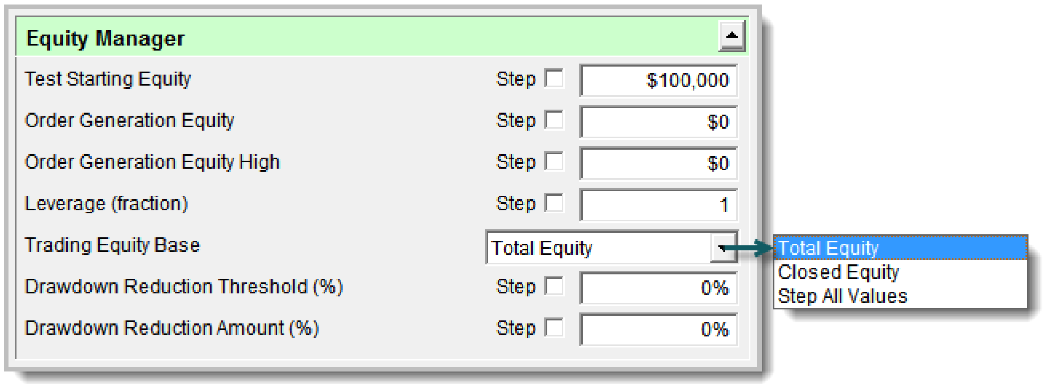


Fees Parameters

Management Fee (Annual % of Equity)	Step ☐	0%
Charge Management Fee		Monthly
Incentive Fee (% of Net New Profits)	Step ☐	0%
Charge Incentive Fee		Yearly
Use Capital Adds Draws		True

Section 8 – Equity Management Parameters

The following is a comprehensive list of the Trading Blox global Equity Management parameters, and an explanation of how each is used. The Test Parameter values can be viewed and changed at the bottom of the *Global Parameters* tab.


Equity Manager

Test Starting Equity	Step ☐	$100,000
Order Generation Equity	Step ☐	$0
Order Generation Equity High	Step ☐	$0
Leverage (fraction)	Step ☐	1
Trading Equity Base		Total Equity
Drawdown Reduction Threshold (%)	Step ☐	0%
Drawdown Reduction Amount (%)	Step ☐	0%

Test Starting Equity
This parameter sets the currency value of the account balance at the start of the simulation (at this point *Closed Equity* = *Total Equity*). It is an initial value only. Trading Blox internally recalculates the various running account balances daily, for the duration of the simulation.

When generating orders, Trading Blox uses the *Starting Balance* to calculate the number of shares or contracts for each order. In order to make sure that the order sizes are accurate, the Starting Balance value should be updated daily using actual trading account balances. The *Starting Balance* for orders should be set to the value corresponding to the *Trading Equity Base* parameter (see below). For *Total Equity* use the actual current daily account values.

There are two special additional Options which affect order generation. These can be used in conjunction or separately from the Broker Positions.

- **Order Generation Equity -** When generating orders, the system first runs a test based on the Global Equity Settings. Let's say you start with $100,000 and end with $150,000 over the course of the test. However, to generate orders your actual account may have a different amount. Perhaps you added money to your account, or made a withdrawal. If you want your orders to reflect your account balance, enter it here.

  The Closed Equity is set to this amount minus any Open Equity in actual positions entered. So your total equity will equal the order generation equity when generating orders. Set to zero to disable this property and use the test simulation equity.

  NOTE: this is only used for Order Generation. It controls the amount of equity available for trading on the order generation date.

- **Order Generation Equity High -** Works in concert with *Starting Balance* (see previous). NOTE: This parameter was designed predominantly for use when generating orders, and is of consequence only if you are using the *Drawdown Reduction Threshold* (DRT) and *Drawdown Reduction Amount* (DRA) parameters (see Equity Management Parameters).

  If you are generating orders with DRT & DRA, Account High works in concert with Account Balance to replicate the current actual drawdown in an actual trading account. In order to mimic the way Trading Blox sizes positions during back testing using DRT and DRA, the value that should be entered in Account High is the highest equity value achieved to date. For instance, if the highest equity value achieved to date in a real trading account is $500,000, and the Account Balance is $400,000, then Trading Blox recognizes that the account is in a 20% drawdown, and passes this information along for use by DRT and DRA.

Set to zero to disable this property and use the test simulation high. Order Generation High is disabled if Order Generation Equity is disabled.

Leverage
While this parameter was designed primarily to allow simulations of stock margin accounts, it can also be used to adjust the leverage of futures trading. Since the leverage available in a futures account is already quite large, this parameter should generally be set to a value of one or less for futures testing.

Lending institutions may lend funds that can be used to leverage your investments. If a lending institution agrees to lend you some amount for every amount of existing account equity, then in Trading Blox the *Trading Equity* (the total money available for trading) would be twice the amount of equity corresponding to the *Equity Base* parameter. Assuming *Equity Base* is set to "Closed Equity", this corresponds to a Leverage value of 2, with the actual trading equity equal to twice the *Closed Equity* at any point in time. If no money was available from lending institutions, then the trading equity would equal *Closed Equity*, and the value of *Leverage* would be 1.

Trading Equity = Leverage x Account Equity

For example, if *Leverage* is increased from a value of 1 to a value of 2, and position sizing parameters remain unchanged, then orders will be sized twice as large.

Leverage can also be set to a fractional value. Setting Leverage to a value of 0.50, for instance, will simulate allocating 50% of your account to cash.

Trading Equity Base
The Trading Equity Base parameter can have one of two separate values:

Closed Equity	Reflects changes in equity from closed out trades only
Total Equity	Reflects *Closed Equity* plus the total *Open Equity* balance

The value of the *Trading Equity Base* defines the available *Trading Equity*. *Trading Equity* is the equity figure used for all money management position sizing calculations and for determining the maximum amount of equity that can be used for margin or purchasing stocks.

You can step the *Trading Equity Base* parameter by selecting the *Step All Values* item.

This will result in two separate runs, one each for *Total Equity* and *Closed Equity*.

Drawdown Reduction Threshold
This parameter and the one that follows allow simulation of the Turtle risk-management rules. They work in concert to reduce the likelihood of dramatic losses by cushioning equity drawdowns. While these parameters and the associated risk management algorithms were designed to support the Turtle System they can also be used with other systems

Drawdown Reduction Threshold (DRT) defines the drawdown point at which the amount of money available for trading is reduced. For example, if DRT is set to 10%, then when a 10% drawdown is incurred, trading equity will be reduced by the percentage specified for the following parameter, *Drawdown Reduction Amount* (DRA).

If you wish to eliminate the effect of DRA entirely, then set the value of DRT to 100% and the DRA to 0%.

The drawdown utilized by DRT is always measured from the most recent *Closed Equity* peak, and is calculated based on the *Closed Equity*.

Drawdown Reduction Amount
Works in concert with *Drawdown Reduction Threshold* (DRT), above. *Drawdown Reduction Amount* (DRA) specifies the percentage by which *Closed Equity* is reduced in the event of a drawdown of the magnitude specified by the DRT parameter.

This parameter pair only affects the sizing of (and the capital available for) new trades. It does not modulate the size of existing positions.

The table below is based on the default Turtles settings of 10% for DRT and 20% for DRA, and shows how DRT and DRA work together to reduce the equity available for investment during periods of drawdown.

As is shown, Available Equity = Actual Closed Equity * (1 - DRA). In this example, the most recent *Closed Equity* high (the peak from which drawdown is measured), is $1,000,000. In Trading Blox 2.0.10, DRA is calculated such that you will never get a zero or negative equity amount. In this way you can test a 90% DRA if you wish.

Actual Closed Equity	Actual DD	DRA	(1-DRA)	Trading Equity
1,000,000	0%	0%	100%	1,000,000
900,000	10%	20%	80%	720,000
800,000	20%	36%	64%	512,000
700,000	30%	49%	51%	357,000
600,000	40%	59%	41%	246,000
500,000	50%	67%	33%	165,000

When actual closed equity drops by 10% (to $900,000), the *Drawdown Reduction Threshold* is violated, and DRA kicks in, reducing the equity available for investment to $720,000 (actual closed equity less 20%). Now, any new trades-instead of being sized to the actual closed equity figure of $900,000-will be sized based on the available equity of $720,000.

DRT and DRA operate in a step-wise fashion: While equity fluctuates below $900,000 but above $800,000 positions will continue to be sized based on the available equity of $720,000.

The effect of DRT and DRA is cumulative (exponential not geometric): If the actual equity falls an additional 10%, to $800,000 (a 20% drawdown as measured from the previous equity peak of $1,000,000), then available equity is reduced further to $512,000 or 64% of actual equity. The process continues as the drawdown steepens.

DRT and DRA work symmetrically as the account begins to climb out of the drawdown, and have no further effect between the time the actual account equity exceeds the original *Drawdown Reduction Threshold* of $900,000 and a new equity high is achieved. If the current drawdown should re-commence before a new equity high occurs, then DRA will kick in again at $900,000.

Notes:

- Whenever a new *Closed Equity* high is achieved, DRT and DRA sit idle, waiting for the next 10% drawdown (or other user-specified DRT value) to occur.
- When running historical simulation tests, the starting *Account Balance* is treated as an equity high when the simulation commences, unless the value of Account High has been set even higher (see Starting Balance and Account High, next).
- DRA is active only during drawdowns whose magnitude exceeds the percentage specified by DRT. So if DRT is set to a large value, it is possible that its threshold will never be violated. If DRT is set to 100%, or if DRA is set to 0%, then the effect of this parameter pair is completely disabled.
- If the values of DRT and DRA remain static, while *Percent Risk per Trade* is increased (i.e. as the percentage of equity risked on each trade is increased and drawdowns are magnified), then DRT will be triggered more frequently, and DRA's effect on performance will be more profound.
- During very large drawdowns (particularly those that occur near the start of a simulation), it is possible that available account equity-having been dramatically reduced by the effect of DRA-will be insufficient to allow new trades to be put on, particularly for futures. At this point, trading will cease.
- The drawdown utilized by DRT is always measured from the most recent *Closed Equity* peak, and is calculated based on the *Closed Equity* high.

Part V

Part 5 – Preferences

Trading Blox's Preferences dialog let you customize you data environment and Trading Blox's reporting. To activate the Preferences Editor, select Edit Preferences from the Edit menu:

The left side of the Preference Editor is the Section Selector. Clicking on an entry in the Section Selector brings the preferences pane for that section into the right side of the dialog which is the Preferences Section Editor.

The Preferences Editor includes five sections:

Data and Order Folders
Sets the folders for data and the destination folder for orders.

Reporting General
Overall settings for reporting.

Monte Carlo
Setting for the Monte Carlo Simulations

Single Run Reporting
Sets the reports included for tests that have only one parameter combination.

Small Test Reporting
Sets the reports included for tests that have only a few parameter combinations. The threshold for determining the difference between a small test and a large test is set in the Reporting General section.

Large Test Reporting
Sets the reports included for tests that have many parameter combinations.

Trade Chart Colors
Sets the colors for the elements of the trade charts.

Trade Details
Setting for which trade detail items are shown on the chart.

Blox Basic Preferences
Settings and preferences for Blox Basic

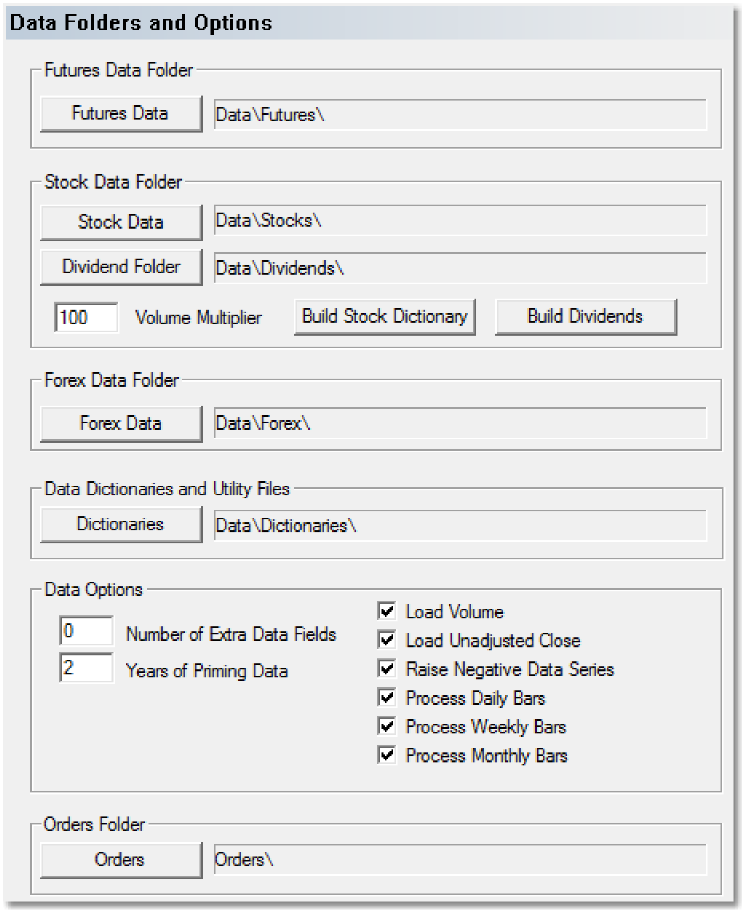
Section 1 – Data and Folder Preferences

Trading Blox determines what data to use for testing based on the data folder settings. These settings determine which Windows folders contain the data for Futures, Stocks, and Forex instruments.

Before you change these settings, please read the section on Historical Data Setup. There must be a dictionary file in the new data folder - otherwise Trading Blox will not read the new data.

If you do not have a Futures Dictionary (futuresinfo.txt file) in the Futures Data folder or Dictionaries folder, you will get warnings on startup, but you can go to the Futures Dictionary Editor and create a new one. Just add the markets for which you have data. Trading Blox will first look in the Futures Data folder for the Futures Dictionary, and then look in the Dictionaries folder. It is not recommended to have a Futures Dictionary in both locations.

If you do not have a Stock Dictionary (stockinfo.txt or stockinfo.csv file), it will look for at MASTER file. If it finds the MASTER file, it will assume Metastock data and read the data. If it does not find a MASTER file, it will assume ASCII Text files and look for all the .txt or .csv files in the folder. It will assemble the Symbols from the file names, ie if the file name is IBM.txt then the symbol will be IBM.


Data Folders and Options

Futures Data Folder
Futures Data: Data\Futures\

Stock Data Folder
Stock Data: Data\Stocks\
Dividend Folder: Data\Dividends\
100 Volume Multiplier

Forex Data Folder
Forex Data: Data\Forex\

Data Dictionaries and Utility Files
Dictionaries: Data\Dictionaries\

Data Options
0 Number of Extra Data Fields
2 Years of Priming Data
✓ Load Volume
✓ Load Unadjusted Close
✓ Raise Negative Data Series
✓ Process Daily Bars
✓ Process Weekly Bars
✓ Process Monthly Bars

Orders Folder
Orders: Orders\

Futures Data Folder

The default location of the folder used for storing the futures data. Within this folder can be other sub folders, or short cuts to other external folders. In addition, the Futures Dictionary has a folder location for each market that is actually used to find the market data. This default folder is where the Find

Symbol button goes to locate the data file, but that data can really be anywhere.

When the Futures Data folder is changed, a message box will indicate whether their exists a Futures Dictionary in the new location and/or the Dictionaries folder. If a Futures Dictionary does exist it will be used. The message box will ask if the Folder Locations for all markets should be reset. This process will locate all the markets using the default folder and replace the current folder location for each market with this new location. This is useful if the all the futures dictionary information is the same, but the folder location has changed. This is not recommended if custom or edited folder locations are used, as they will be replaced with the default location based on the new Futures Data folder location.

Stock Data Folder

The location of the folder used for storing the stock data.

Dividend Folder

The location for storing the dividend files. The Dividend file is named using the symbol and the dividend file suffix, default is "_Div.csv" but this can be changed in the TradingBlox.ini file.

Volume Multiplier

The multiplier for the volume. If your data reports 1,000 instead of 100,000 use a multiplier of 100. The default is 100.

Build Stock Dictionary

Uses the CSI market data file to build the stock dictionary. See the Stock Dictionary for more detailed information.

Build Dividends

Uses the CSI dividend data to build the dividend files for all stocks in your Stock Dictionary. The stock dividend data from CSI is encrypted by default. To use this feature you must contact CSI and make arrangements to have this data unencrypted for your subscription account. This is an additional cost service from them. Alternately you can build your own dividend files manually from another source for each stock and place in the Dividend Folder. The format is Date,Dividend and you can see example files in the sample data that ships with the product.

Forex Data Folder

The location of the folder used for storing the forex data.

Orders Folder

Change the location of files made by generating orders.

Extra Data Fields

Enter the number of "Extra Data Fields" you have in your data. If you have none, set this to 0 to save memory. You can have between 0 and 8 extra data fields in the ascii text data format of D,0,H,L,C,V,OI,DM,UC,E1,E2,E3,E4,E5,E6,E7,E8 for futures and D,0,H,L,C,V,UC,E1,E2,E3,E4,E5,E6,E7,E8 for stocks and forex.

Years of Priming Data

Enter the number of years of priming data required. The default is 5. This is the number of years of data that will be loaded, if available, prior to the user entered start date of the test. This data is generally used to prime indicators prior to the test start so the test can start right on the start date.Recommended to keep this at 5 years unless you are trading lots of stocks and need the memory. To conserve memory, set this as low as your indicator priming requires, or zero if you don't mind your trading starting later than your test start date.

Load Volume

Check to load the volume in your data file into memory. Uncheck this option if you don't use volume for your testing, and you want to save memory on a large stock data load. Be sure to turn off any volume filters if you don't load the volume. Recommended to keep this checked.

Load Unadjusted Close

Check to load and use the unadjusted close data. Uncheck this to save memory if you don't use this data. Recommended to keep this checked.

Raise Negative Data Serie

If a data series has negative values, this option will raise the whole series up to positive territory. Uncheck this option if the data is all positive, as it will save data load time. Recommended to keep this checked.

Process Daily Bars

This option when checked will create daily bars from intraday bars. Can be used in scripting using the instrument.dayOpen[], instrument.dayHigh[], instrument.dayLow[], and instrument.dayClose[] properties. The instrument.dayIndex property counts the number of historic days available at any time, and should be checked before doing a lookback on this series.

Process Weekly Bars

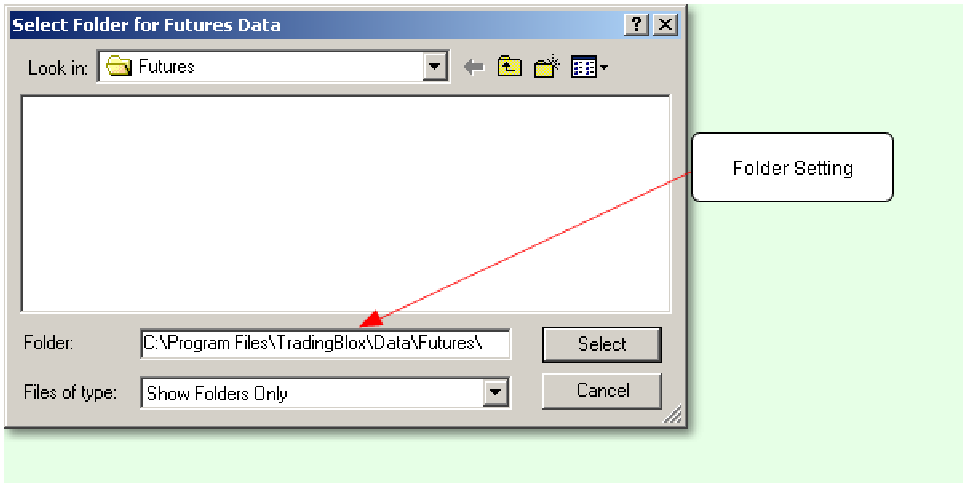
This option when checked will create weekly bars from the daily bars. Can be used in scripting using the instrument.weekOpen[], instrument.weekHigh[], instrument.weekLow[], and instrument.weekClose[] properties. The instrument.weekIndex property counts the number of historic weeks available at any time, and should be checked before doing a lookback on this series.

Process Monthly Bars

This option when checked will create monthly bars from the daily bars. Can be used in scripting using the instrument.monthOpen[], instrument.monthHigh[], instrument.monthLow[], and instrument.monthClose[] properties. The instrument.monthIndex property counts the number of historic months available at any time, and should be checked before doing a lookback on this series.

Note that to conserve memory when doing super large tests, the process daily, weekly, and monthly checkboxes should be unchecked if the test does not require the use of these properties.

Pressing a "Change Folder" button will bring up a window to select the location. To select a folder you must open and go "into" it, not just click and highlight it. You can verify the folder that will be selected using the Folder Setting section of the dialog.


Select Folder for Futures Data

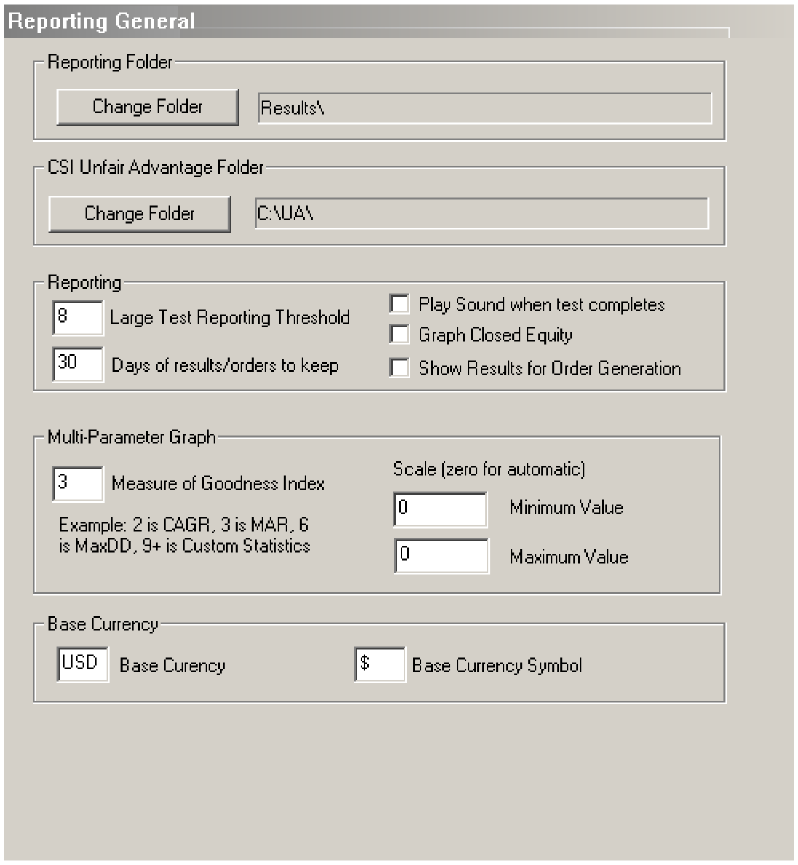
Look in: Futures

Folder: C:\Program Files\TradingBlox\Data\Futures\

Files of type: Show Folders Only

Section 2 – Reporting General

The Reporting General section of the Preferences dialog sets general reporting preferences:


Reporting General

Reporting Folder

Results\

CSI Unfair Advantage Folder

C:\UA\

Reporting
8 Large Test Reporting Threshold
30 Days of results/orders to keep
☐ Play Sound when test completes
☐ Graph Closed Equity
☐ Show Results for Order Generation

Multi-Parameter Graph
3 Measure of Goodness Index
Example: 2 is CAGR, 3 is MAR, 6 is MaxDD, 9+ is Custom Statistics
Scale (zero for automatic)
0 Minimum Value
0 Maximum Value

Base Currency
USD Base Curency
$ Base Currency Symbol

Reporting Folder
Determines the destination folder for the HTML reports that are generated by Trading Blox when you run historical tests.

CSI Unfair Advantage Folder
This is the folder in which CSI UA is located if you have it installed. This program is optional, but the location is used when building the Stock Dictionary on the Data Folders and Options Preferences page.

Large Test Reporting Threshold
Sets the number of distinct parameter combinations (or runs) which determines the cutoff between a Small Test and a Large Test (see the next section Report Content Specification for more details). In the above example, tests with 8 or less distinct parameter combinations will use the settings for Small Tests, test with 9 or more distinct parameter combinations will use the Large Test settings.

Days of results and orders to keep
Set the number of days of historical results and orders to keep in the Results and Orders folder. Defaults to 90 days. If you set this value to 30, Trading Blox will delete any results files and order files that are more than 30 days old. This will be done on startup every time.

Play Sound when test completes
Turns the test completion sound on or off.

Graph Closed Equity
When checked, the closed equity line will be graphed along with the open equity in the summary results graph. If left unchecked, the closed equity will not be graphed.

Show Results for Order Generation
If this box is checked, the test results for the test run prior to order generation will be displayed. If this is unchecked, just the order generation report will be displayed. Default is unchecked.

Measure of Goodness Index
This is used for the Multi-Parameter chart and the Parameter Stepping Graphs. This can be a number between 1 and 8 to use built in statistics, or a number greater than 8 for custom statistics. The numbers represent the left to right index of the Summary Statistics for a test as follows:

1 -- End Balance
2 -- CAGR
3 -- MAR
4 -- Sharpe
5 -- Annual Sharpe
6 -- Max Total Equity Drawdown
7 -- Longest Drawdown
8 -- Number of trades
9+ -- Custom statistics

Scaling
Enter the minimum and maximum value for the multi-parameter graph. Enter zero to have the chart auto scale.

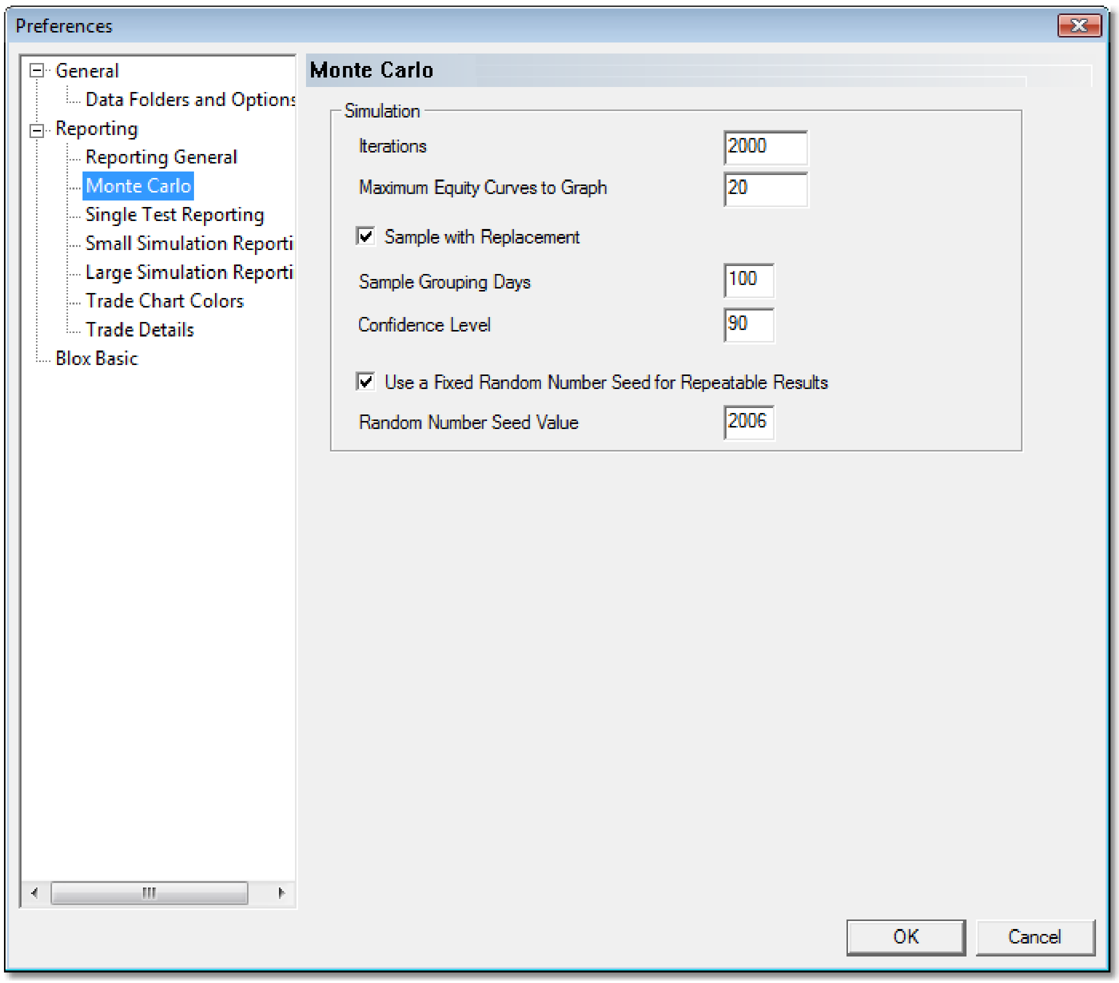
Base Currency
Enter the system wide base currency to use. The default is USD.This is the currency that all market results will be converted back into. You must have a forex pair in the Forex Data Folder to convert back from any foreign futures or stocks, back into the system wide base currency.

Base Currency Symbol
Enter the symbol of the system wide base currency to use.

Section 3 – Monte Carlo

Trading Blox supports Monte Carlo Simulations. The Monte Carlo preferences section controls the overall simulation.

There are seven items which control the Monte Carlo simulation and output:



Monte Carlo

Simulation

Iterations	2000
Maximum Equity Curves to Graph	20
✓ Sample with Replacement	
Sample Grouping Days	100
Confidence Level	90
✓ Use a Fixed Random Number Seed for Repeatable Results	
Random Number Seed Value	2006

Iterations
This controls the total number of separate alternate equity curves which are used to generate the reports. Bigger is better but bigger takes longer. We recommend leaving these set to 2000 for a 10 year test.

Maximum Equity Curves to Graph
This controls how many equity curves out of the total number of iterations will be graphed on the Monte Carlo Equity Curve Graph. You can set this to the same as iterations but this might result in it being impossible to see any distinct equity curves since they will all write over each other.

Sample with Replacement
This controls how the alternate equity curves are generated. If you check this it means that we will randomly select a portion of the equity curve and the same portion might be selected more than once. This setting will result in differing end points for the equity curve.

If this setting is not checked it will result in a simple reordering of the equity curve since each section of the initial equity curve will only be selected once and only once.

Sample Grouping Days
This controls how many consecutive days at a time are selected from the initial equity curve. A setting greater than one will preserve some of the serial correlation in the initial equity curve.

Confidence Level
This setting controls where the confidence lines are placed the graphs. It is a percentage which can range from 1 to 100. It also controls the value for the Confidence Level report. For example, if you set this to 85, we will report the MAR, CAGR%, Sharpe Ratio, R-Squared value, maximum drawdowns and length at the 85th percentile. In other words that value where 85% of the simulations had worse performance. This is fairly obvious from the graphs and is easier to understand when you look at it than by reading this.

Use a Fixed Random Number Seed for Repeatable Results
If this is checked then the value of the next field will be used to seed the random number generator. Effectively, this means that the sequence will repeat each time. If unchecked then a new random seed will start the random number sequence which means that each sequence will be different.

You can use this to determine if the number of iterations is sufficient. For smaller numbers of iterations, less than a few thousand, you will get different graphs and confidence level values if you have a different set of random numbers. If this happens it is a sign that the number of iterations is insufficient.

Random Number Seed Value
The value to use to start the random number sequence. Not used if the above is not checked.

Section 4 – Report Content Specification

Generally, traders are interested in more detail when they run tests that have only one combination of parameters (i.e. none of the parameters are stepped). When traders run tests with many different combinations as part of an optimization, they are generally concerned with summary information like CAGR%, MAR Ratio, and a few other summary test values. Trading Blox let's you specify which reports get included for different size tests:

The three sizes are:

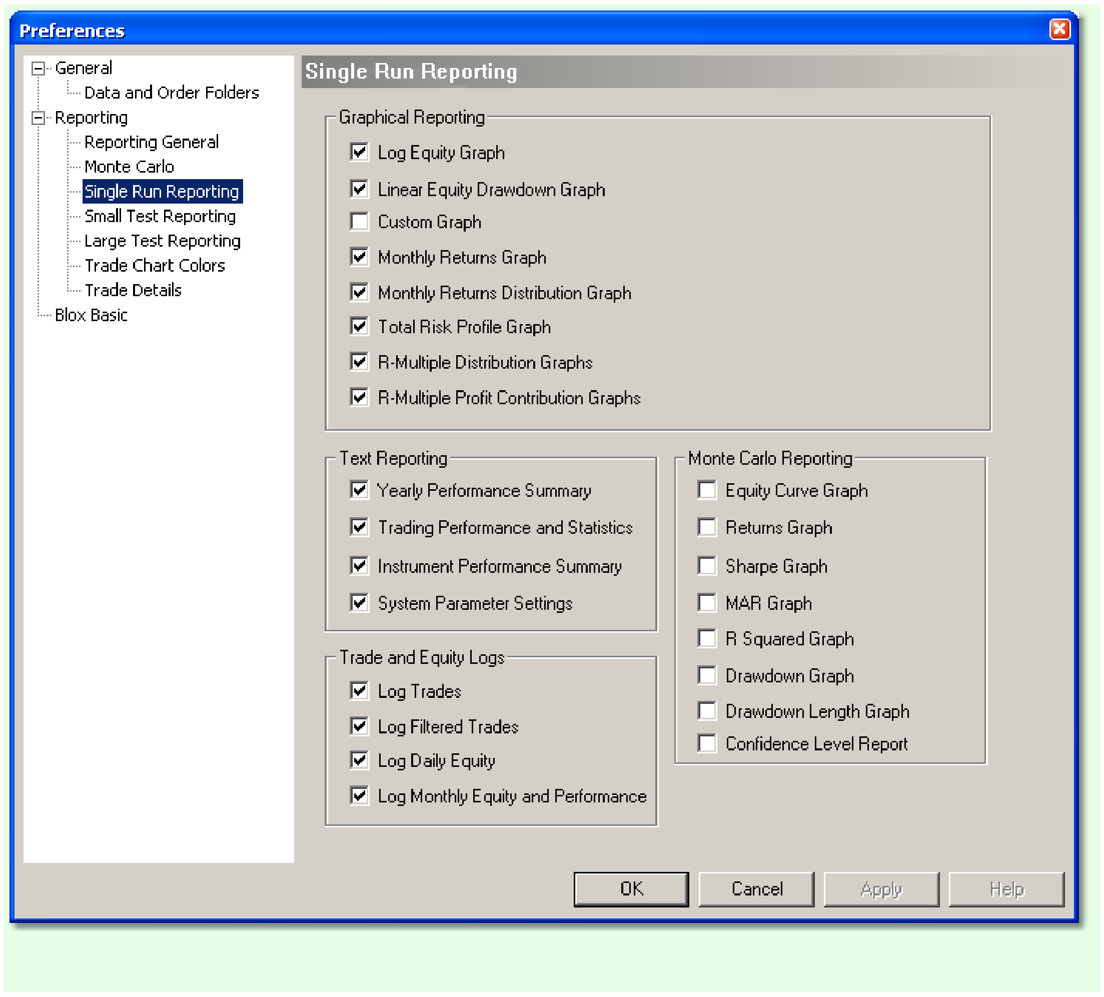
- Single Run - Tests that have only one parameter combination and do not step.
- Small Test - Tests that have a small number of parameter steps
- Large Test - Tests that have many parameter steps.

The Reporting General preferences page lets you set the number of parameter steps which determine the cutoff between a small test and a large test.

There are four sections for each type of report:

Graphical Reporting
Text Reporting
Trade and Equity Logs
Monte Carlo Simulation Reports

Single Run Reporting
The following reports are available for Single Run Reporting:



Single Run Reporting

Graphical Reporting
- [✓] Log Equity Graph
- [✓] Linear Equity Drawdown Graph
- [ ] Custom Graph
- [✓] Monthly Returns Graph
- [✓] Monthly Returns Distribution Graph
- [✓] Total Risk Profile Graph
- [✓] R-Multiple Distribution Graphs
- [✓] R-Multiple Profit Contribution Graphs

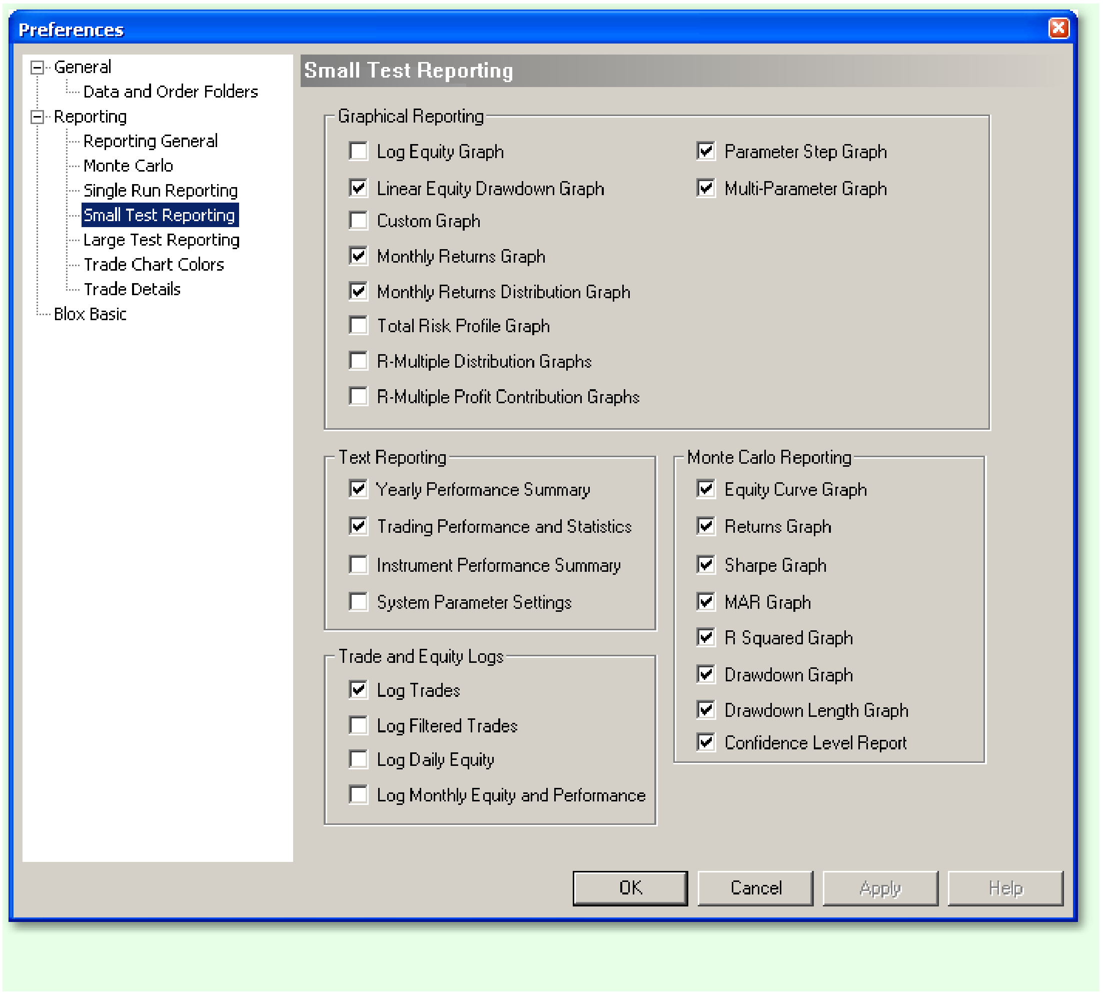
Text Reporting
- [✓] Yearly Performance Summary
- [✓] Trading Performance and Statistics
- [✓] Instrument Performance Summary
- [✓] System Parameter Settings

Trade and Equity Logs
- [✓] Log Trades
- [✓] Log Filtered Trades
- [✓] Log Daily Equity
- [✓] Log Monthly Equity and Performance

Monte Carlo Reporting
- [ ] Equity Curve Graph
- [ ] Returns Graph
- [ ] Sharpe Graph
- [ ] MAR Graph
- [ ] R Squared Graph
- [ ] Drawdown Graph
- [ ] Drawdown Length Graph
- [ ] Confidence Level Report

Small Test Reporting

The following reports are available for Small Test Reporting:



Small Test Reporting

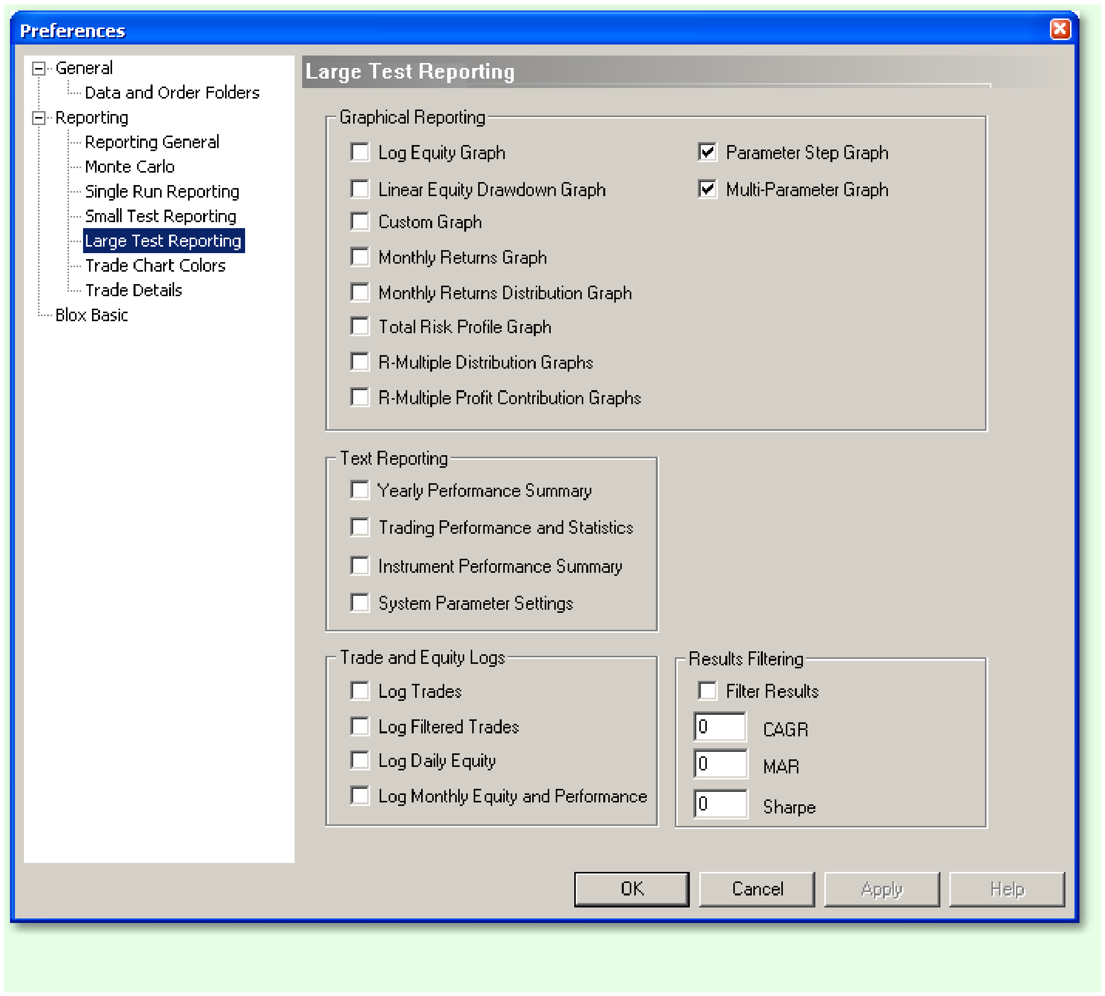
Graphical Reporting
- [ ] Log Equity Graph
- [✓] Linear Equity Drawdown Graph
- [ ] Custom Graph
- [✓] Monthly Returns Graph
- [✓] Monthly Returns Distribution Graph
- [ ] Total Risk Profile Graph
- [ ] R-Multiple Distribution Graphs
- [ ] R-Multiple Profit Contribution Graphs
- [✓] Parameter Step Graph
- [✓] Multi-Parameter Graph

Text Reporting
- [✓] Yearly Performance Summary
- [✓] Trading Performance and Statistics
- [ ] Instrument Performance Summary
- [ ] System Parameter Settings

Trade and Equity Logs
- [✓] Log Trades
- [ ] Log Filtered Trades
- [ ] Log Daily Equity
- [ ] Log Monthly Equity and Performance

Monte Carlo Reporting
- [✓] Equity Curve Graph
- [✓] Returns Graph
- [✓] Sharpe Graph
- [✓] MAR Graph
- [✓] R Squared Graph
- [✓] Drawdown Graph
- [✓] Drawdown Length Graph
- [✓] Confidence Level Report

Large Test Reporting
The following reports are available for Large Test Reporting:



Large Test Reporting

Graphical Reporting

- [ ] Log Equity Graph
- [ ] Linear Equity Drawdown Graph
- [ ] Custom Graph
- [ ] Monthly Returns Graph
- [ ] Monthly Returns Distribution Graph
- [ ] Total Risk Profile Graph
- [ ] R-Multiple Distribution Graphs
- [ ] R-Multiple Profit Contribution Graphs
- [✓] Parameter Step Graph
- [✓] Multi-Parameter Graph

Text Reporting

- [ ] Yearly Performance Summary
- [ ] Trading Performance and Statistics
- [ ] Instrument Performance Summary
- [ ] System Parameter Settings

Trade and Equity Logs

- [ ] Log Trades
- [ ] Log Filtered Trades
- [ ] Log Daily Equity
- [ ] Log Monthly Equity and Performance

Results Filtering

- [ ] Filter Results
- 0 CAGR
- 0 MAR
- 0 Sharpe

Normally large tests have all reporting off to increase the speed of testing. Additionally, Trading Blox may run out of memory on a very large test if "Log Trades" is checked.

If "Log Trades" is set to off only the summary results of the test will be shown. Trade details will not be shown. It is recommended that the user test large parameter runs with these defaults and then use smaller tests to narrow down the best results.

Another feature of Large Test Reporting is the ability to filter the results by CARG, MAR, and Sharpe values. For instance, if you have a test of 40,000 different combinations, you can filter the results to only display runs with a CARG of 20% or more. This is handy because otherwise you would be looking at 40,000 different test results.

4.1 Graphical Reporting

The graph-based reports include:

Log Equity Graph
Graph of *Total Equity* and *Closed Equity* on a logarithmic scale.

Linear Equity Drawdown Graph

Graph of *Total Equity* and *Closed Equity* on a linear scale with drawdowns.

Custom Graphs
This option shows the Custom graphs, that could be created by Blox Scripting. These graphs could be any data as defined by the system.

Monthly Returns Graph
Graph of the Monthly Returns.

Monthly Returns Distribution Graph
Graph showing the distribution of the Monthly Returns.

Total Risk Profile Graph
Graph of the total risk to *Total Equity* and *Closed Equity*.

R-Multiple™ Distribution Graph
Graph showing the distribution of trades by R-Multiple™.

R-Multiple™ Contribution Graph
Graph showing the profit contribution of trades by R-Multiple™.

4.2 Text Reporting

The text-based reports include:

Yearly Performance Summary
A summary of the trading performance by year.

Trading Performance and Statistics
General performance statistics.

Instrument Performance Summary
Summary performance statistics broken down by instrument.

System Parameter Settings
Listing of the settings for every parameter of each active system.

4.3 Trade and Equity Logs

The Logs include:

Log Trades
This will log all trades in the Trade Log, and save trade details to the chart. For large tests, you can save memory and increase speed by keeping this box unchecked. When the system does not have to save trade details for every run, you can run much larger tests with available memory.

Log Filtered Trades
This option when checked logs the reason for all filtered or rejected trades into the Filtered Trade Log. It will indicate if a trade was filtered because of equity, volume, portfolio manager, risk manager, lock limit day, or open through the stop.

Log Daily Equity
This will log daily equity in the Daily Equity Log.

Log Monthly Equity and Performance

This will log monthly performance in the Monthly Equity Log.

Print Output
The file to which the PRINT statement goes when using the PRINT statement in Blox Basic Scripting. See the Print Output File.

4.4 Monte Carlo Reporting

Trading Blox supports Monte Carlo simulations and reporting of various types. These can be activated for Single Run and Small Tests. They are not active for Large Tests: The Monte Carlo Simulation reports include:

Equity Curve Graph
A log-scale of the different equity graphs overlaid on the test equity curve.

Returns Graph
A curve of the distribution frequency and accumulated percentiles of the various returns. This graph won't be useful unless you check Sample with Replacement as the returns will be the same without that checked.

Sharpe Graph
Same as above but with CAGR% for each alternate equity curve.

MAR Graph
Same as above but with MAR for each alternate equity curve.

R Squared Graph
Same as above but with R-Squared for each alternate equity curve.

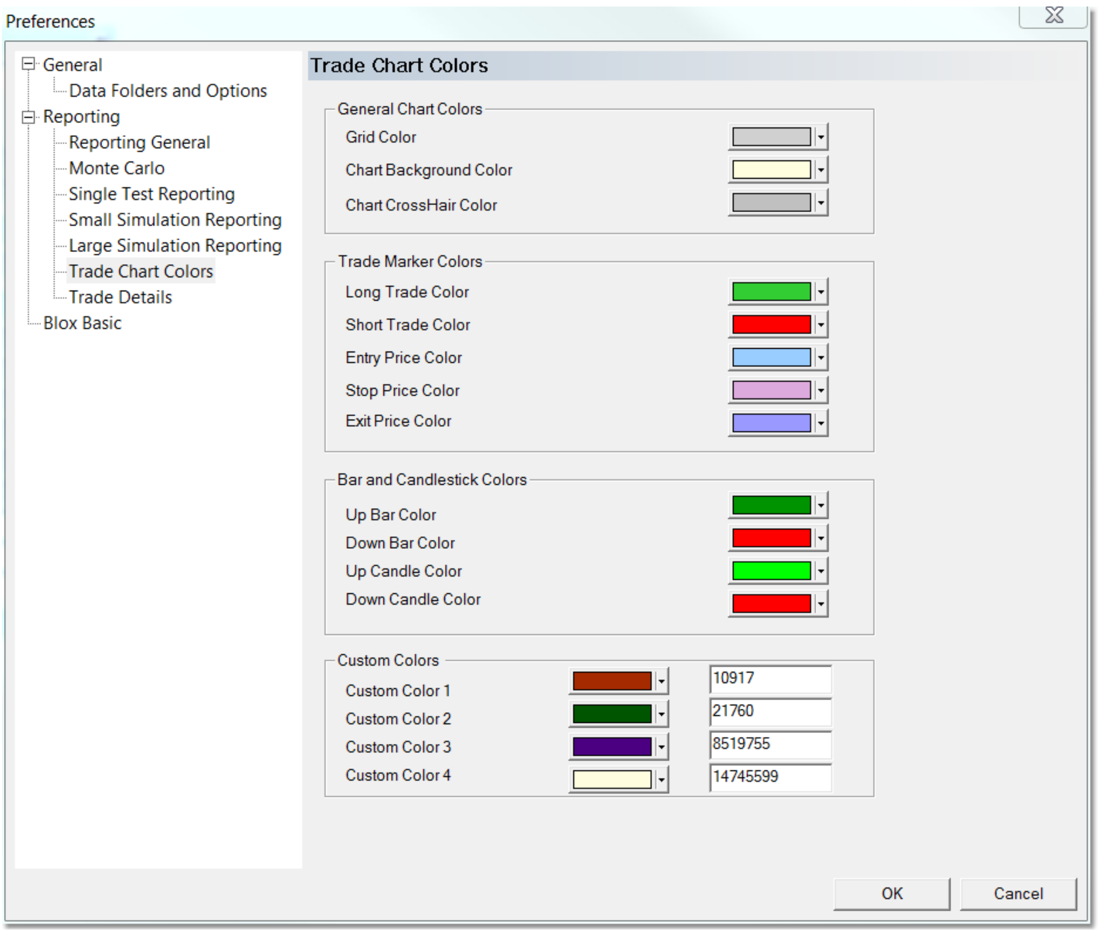
Drawdown Graph
Same as above but with the maximum drawdown, second largest, and third largest for each alternate equity curve.

Drawdown Length Graph
Same as above but with the longest drawdown, second longest, and third longest drawdown for each alternate equity curve.

Confidence Level Report
A text-based percentile report which will be run as part of the normal performance settings. It shows the values of the performance statistics at the percentiles specified by the confidence level above. NOTE: You must also have Trading Performance and Statistics checked for this report to work.

Section 5 – Trade Chart Colors

Trading Blox lets you specify the colors for most of the elements on the trade chart using the Trade Chart Colors section of the Preferences Editor.



Trade Chart Colors

General Chart Colors

Grid Color	⬜
Chart Background Color	⬜
Chart CrossHair Color	⬜

Trade Marker Colors

Long Trade Color	🟩
Short Trade Color	🟥
Entry Price Color	🟦
Stop Price Color	🟪
Exit Price Color	🟦

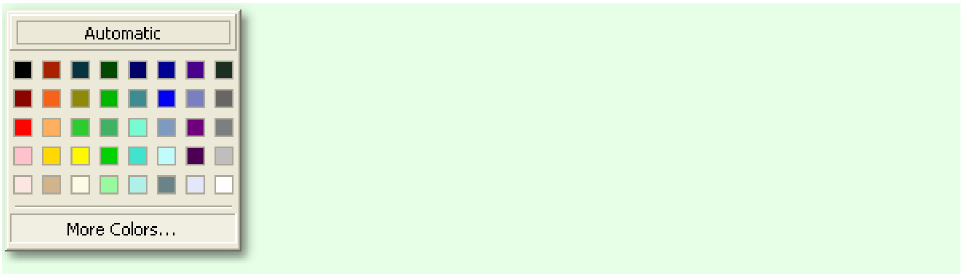
Bar and Candlestick Colors

Up Bar Color	🟩
Down Bar Color	🟥
Up Candle Color	🟩
Down Candle Color	🟥

Custom Colors

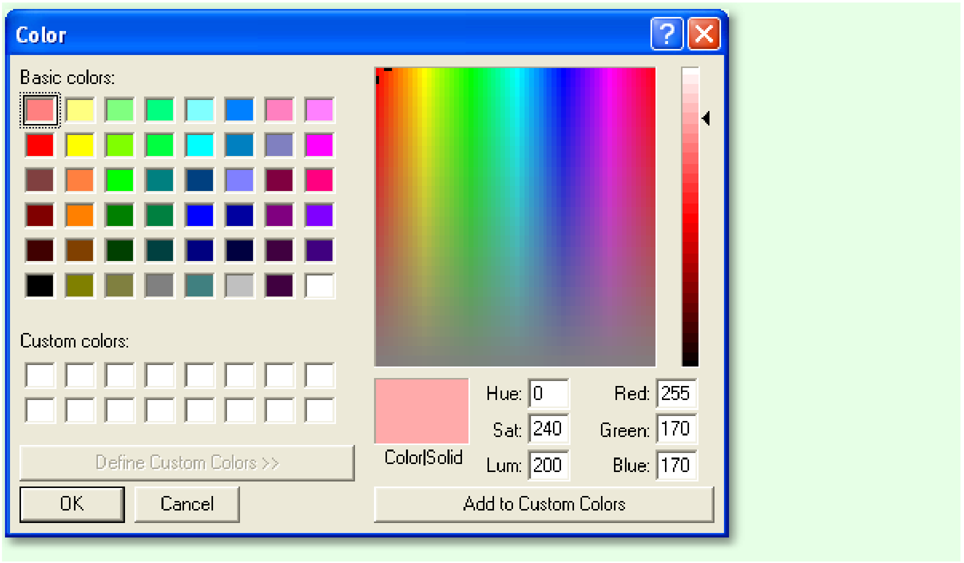
Custom Color 1	🟫	10917
Custom Color 2	🟩	21760
Custom Color 3	🟪	8519755
Custom Color 4	⬜	14745599

To select a new color, click on the color button. This will bring up a color selector window:



Selecting *Automatic* will set the item's color to the default color that was set when you received Trading Blox.

If one of the built in colors is not to your liking then you can press the *More Colors* button which will bring up the color picker:



Hue: 0
Sat: 240
Lum: 200
Red: 255
Green: 170
Blue: 170

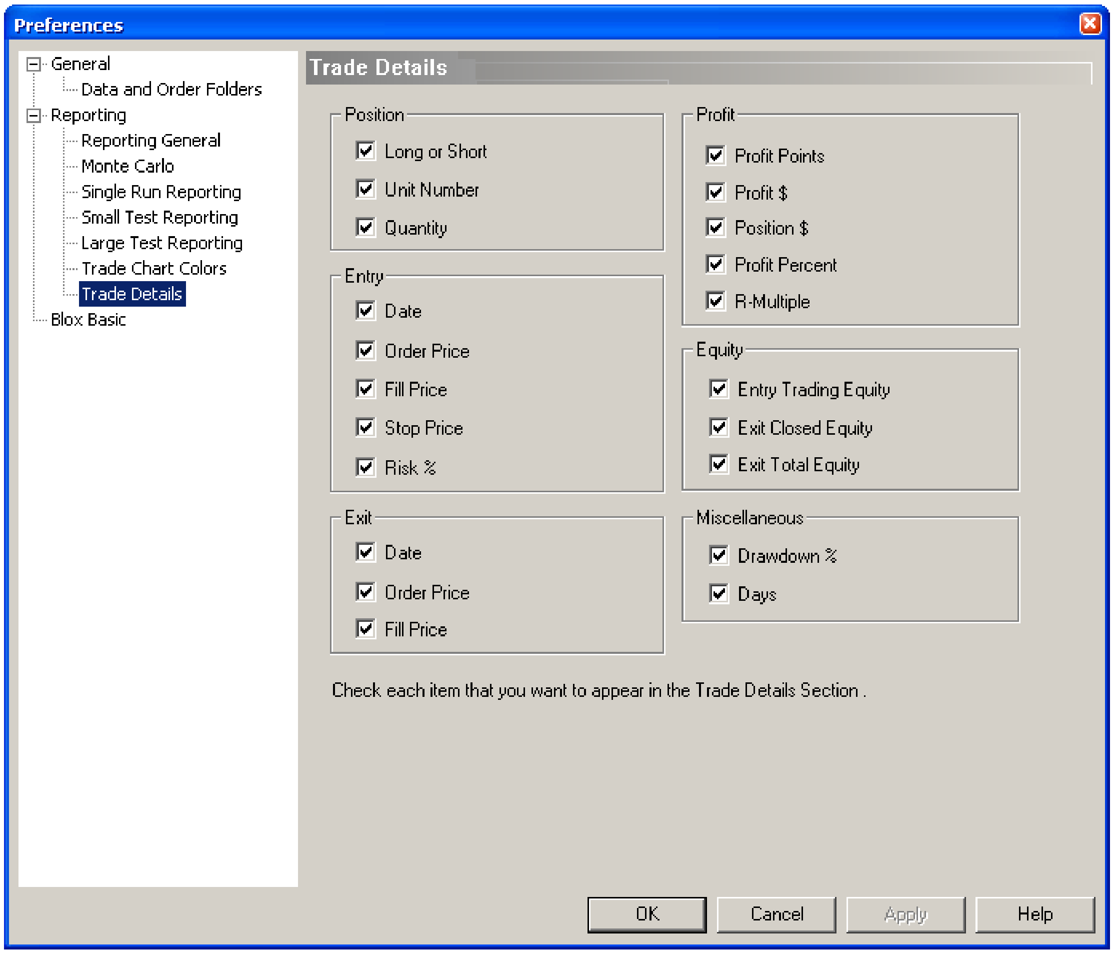
You can select a color by clicking on the rainbow color or by entering the Red, Green, and Blue values directly. These values range from 0 to 255. You can also change the brightness by selecting a point on the Brightness bar on the right hand side.

Note that the cross hair color will be reversed from the chosen color if the background is dark. The color will be as selected for a light background.

The Custom Colors can be used in Blox Basic to dynamically change the color of bars and indicators.

Section 6 – Trade Details

You can set the details of the trade that will be displayed in the trade detail area when viewing historical trades on the graph (in the lower right of the screen.)



Trade Details

Position
- [✓] Long or Short
- [✓] Unit Number
- [✓] Quantity

Entry
- [✓] Date
- [✓] Order Price
- [✓] Fill Price
- [✓] Stop Price
- [✓] Risk %

Exit
- [✓] Date
- [✓] Order Price
- [✓] Fill Price

Profit
- [✓] Profit Points
- [✓] Profit $
- [✓] Position $
- [✓] Profit Percent
- [✓] R-Multiple

Equity
- [✓] Entry Trading Equity
- [✓] Exit Closed Equity
- [✓] Exit Total Equity

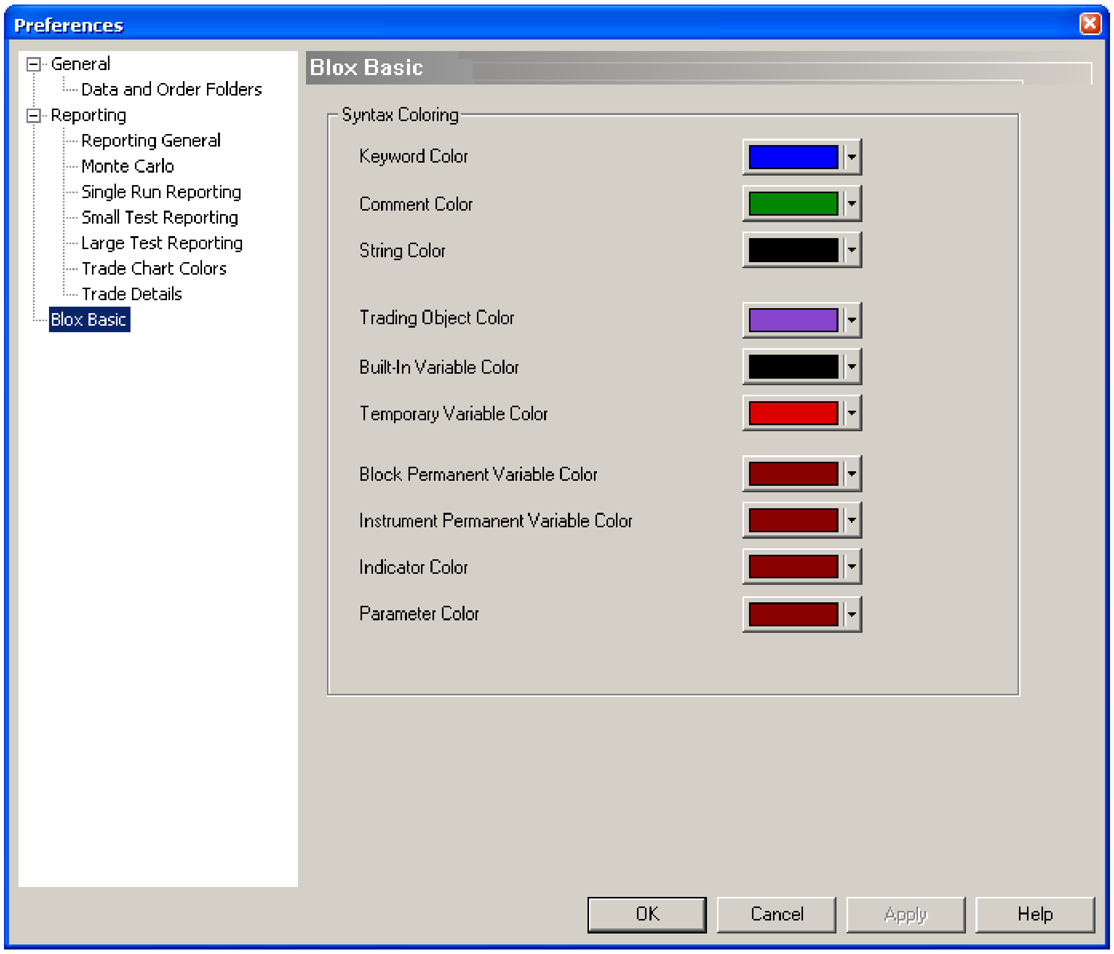
Miscellaneous
- [✓] Drawdown %
- [✓] Days

Check each item that you want to appear in the Trade Details Section .

Section 7 – Blox Basic Preferences

If you have Trading Blox Builder, you can access these preference for the Blox Editor.
Trading Blox Turtle and Trading Blox Professional do not edit Blox, so they do not have this option.

You can set the colors for Blox Basic:



Blox Basic

Syntax Coloring

Keyword Color	🟦
Comment Color	🟩
String Color	⬛
Trading Object Color	🟪
Built-In Variable Color	⬛
Temporary Variable Color	🟥
Block Permanent Variable Color	🟥
Instrument Permanent Variable Color	🟥
Indicator Color	🟥
Parameter Color	🟥

Part VI

Part 6 – Historical Testing

Trading Blox has the most sophisticated historical simulation engine available. This engine lets you accurately test trading systems while accounting for all of the costs and issues that are part of real trading.

The distinguishing features of Trading Blox's *Test Simulation Engine* include:

Multiple Simulation Suites
Trading Blox supports a concept called suites. A suite lets you keep track of different tests at the same time. Each suite is a self-contained simulation specification that includes all the parameters for a complete test including which systems are part of the simulation, their equity allocation, the simulation parameters for each system, as well as the values for global parameters like commission rates, slippage assumptions, the starting balance, etc.

Multi-Market Simulations
Trading Blox simulations include all the markets in a test portfolio to accurately reflect the effect of simultaneous trades in different markets.

Multi-System Simulations
Trading Blox simulations can include many systems each trading simultaneously using an allocation from the common equity pool.

Multi-Portfolio Simulations
Each system can trade a separate and distinct portfolio of futures, stocks or forex markets.

Stepped-Parameter Simulations
Any of the simulation or system-level parameters can be varied during one simulation to show an accurate reflection of the effect of changes of that parameter on a single tests results. See Parameter Stepping for more details of this important feature.

Robust Accounting
Trading Blox keeps track of statistics and simulation results so you can better understand your systems.

True Dynamic Money Management
Unlike other products, Trading Blox's money management is not a post-simulation process applied to a list of fixed trades. Trading Blox's money management runs dynamically during the trade simulations so your trading system can use the value of the current equity and risk to control system entries, exits, and stops for increased control.

Contract-Roll Accounting
Trade simulations using back-adjusted data do not normally account for the effect of increased commissions and slippage due to rolling futures contracts over to new contract months when a position is held for more than a few days. Trading Blox's Contract Roll Accounting allows you to accurately simulate the costs of contract rolling in the simulation.

Volatility-Based Slippage
While Trading Blox supports the common fixed slippage used by other testing packages, it also supports the simulation of slippage using a more accurate volatility-based slippage formula that takes into account the changes in market volatility on the slippage.

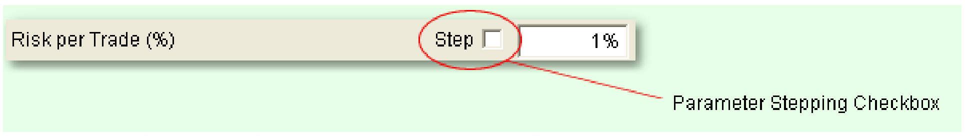
Section 1 – Parameter Stepping

Altering the input value of even a single rule or test condition will often cause changes that ripple throughout the system, directly affecting its performance characteristics, often in dramatic and unforeseen ways. To assist users in gaining a clearer understanding of this important concept, Trading Blox incorporates an extremely powerful feature called *Parameter Stepping*. For any parameter, *Parameter Stepping* allows a range of values to be stepped through and tested one increment at a time - all within the course of a single test.

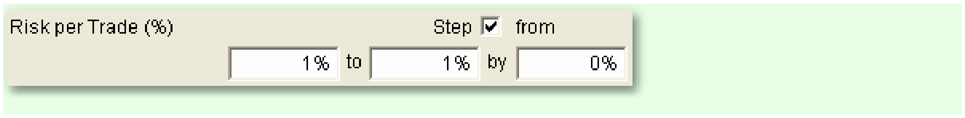
This document and the Trading Blox interface refers to the simulation of a unique combination of parameter input values as a *Test*. Thus, every simulation session (activated by depressing the Run Simulation button), will contain at least one *Test*, or in the case of *Parameter Stepping,* many *Tests*.

For example, if you want to see how changing the value of the Risk per Trade parameter affects overall profitability, you could conduct a number of separate (single-run) test sessions, changing the value of the Risk per Trade each time. Using Parameter Stepping, you could execute multiple parameter runs within a single test session, as follows:

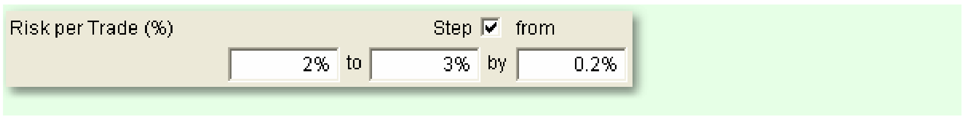
First check the parameter stepping check box:



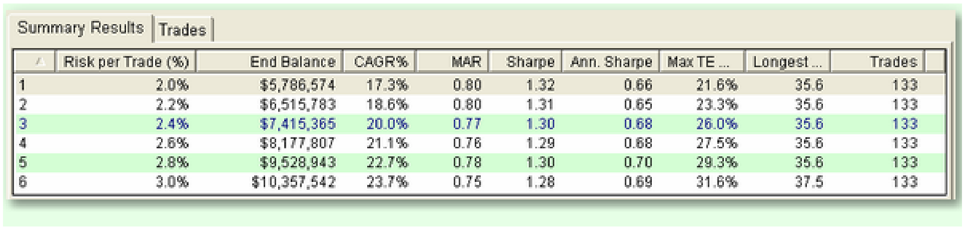
This will expand the single edit box to three separate edit boxes:



You can change the values in the edit boxes to control the stepping range and increment. The following shows the values that will result in six separate stepped parameter runs ranging from 2% to 3% in step increments of 0.2%, with values of 2%, 2.2%, 2.4%, 2.6%, 2.8% and 3%:

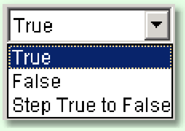


After running the test, the Summary Results for this test will contain a Run Summary List which shows the major performance measures for each of the values of the parameter. This list can be sorted by clicking any of the column headings with the mouse:



	Risk per Trade (%)	End Balance	CAGR%	MAR	Sharpe	Ann. Sharpe	Max TE ...	Longest ...	Trades
1	2.0%	$5,786,574	17.3%	0.80	1.32	0.66	21.6%	35.6	133
2	2.2%	$6,515,783	18.6%	0.80	1.31	0.65	23.3%	35.6	133
3	2.4%	$7,415,365	20.0%	0.77	1.30	0.68	26.0%	35.6	133
4	2.6%	$8,177,807	21.1%	0.76	1.29	0.68	27.5%	35.6	133
5	2.8%	$9,528,943	22.7%	0.78	1.30	0.70	29.3%	35.6	133
6	3.0%	$10,357,542	23.7%	0.75	1.28	0.69	31.6%	37.5	133

Non-numeric parameters that use pop-up menus for True and False, or other values, can also be stepped by selecting the "Step True to False" or other stepping item

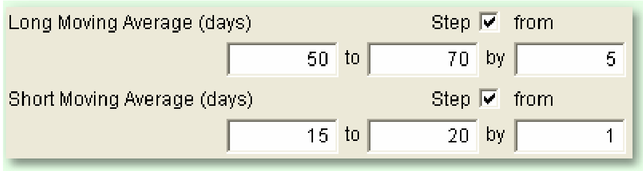


This will result in two tests, one with the value set to True and one with it set to False.

For Integer type parameters only, there is an option to step by percent or fixed value. So in the following example, you could step from 50 to 70 by 5% or by 5. Use the percent sign in this case to indicate a percent step. Note that this is different than the above example which is stepping by a fixed value but the value happens to be a percent.

It is also possible to step through a range of values for multiple parameters, all within a single simulation. Keep in mind, though, that the total number of unique *Tests* within a Parameter Stepping simulation is equal to the number of steps in each range of values, multiplied together. Because of this multiplicative effect, automated Parameter Stepping simulations can grow quite large.

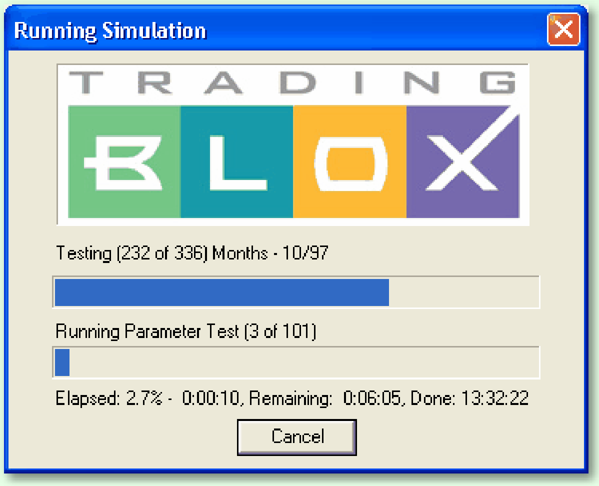
Consider the following combination:


Long Moving Average (days) Step ☑ from 50 to 70 by 5

Short Moving Average (days) Step ☑ from 15 to 20 by 1

The Long Moving Average has five steps (50, 55, 60, 65 and 70), and the Short Moving Average has six (15, 16, 17, 18, 19, and 20). Based on these instructions, the total number of runs that Trading Blox will automatically execute is 5 x 6, or 30. If a third condition with only five (3) steps is added, then the number of runs will equal 5 x 6 x 5, or 150.

Once a simulation has begun, Trading Blox shows a progress dialog. This dialog has two progress bars which indicate how much of the simulation has completed:



Testing (232 of 336) Months - 10/97

Running Parameter Test (3 of 101)

Elapsed: 2.7% - 0:00:10, Remaining: 0:06:05, Done: 13:32:22

It also shows the percentage of the simulation and elapsed time, the time remaining until the simulation is complete and the time when the simulation is estimated to be done.

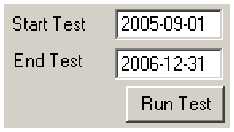
Trading Blox first loads the price data into memory before applying the trading rules. Price data is loaded only once at the beginning of the simulation, even when the simulation will test hundreds of parameter combinations. The symbols for each of the futures, stock or forex market data which are loaded are shown above the top progress bar as the price data is read.

After the data is read, the top progress bar shows the month being tested as Trading Blox applies the trading rules to month after month of historical data for each unique combination of parameter values. The bottom progress bar displays which test (out of the total number of tests), is currently being tested. In the above example, the progress bar shows that the simulation is currently processing test number 3 out of the 101 that will be tested as part of the simulation.

For simulations where there is no stepping and only a single test, the bottom progress bar will move only twice - once when the data has been read and once after the test is complete.

Section 2 – Specifying the Test Dates

The dates for a historical simulation test are specified in the Test Control Area:


Start Test 2005-09-01
End Test 2006-12-31

The date for the start date and end date of the time period for the historical simulation can be specified by typing the start and end dates in the following formats. These formats can also be used for data files and when entering actual positions in the Order Generation Preferences.

YYYY-MM-DD
YYYYMMDD
YYMMDD
MM/DD/YY
MM/DD/YYYY

The above examples shows dates for a simulation that runs from September 1, 2005 through December 31, 2006.

The test will start on the first weekday equal to or after the *Test from* edit box through the last day of available data, or the test end date specified in the *to* edit box, whichever is first. It is possible to test multiple start and end days in the same test by stepping through values in the global parameters.

If your test start date is near the beginning of, or before, the start of data for some instruments, those instruments will start trading when their indicators have primed. The priming period is determined by adding the max prime required for indicators and the max prime required for lookback parameters.

Section 3 – Order Filter

In certain circumstances, Trading Blox will filter an order without placing with the broker object even though it might have been filled based on the order type and price data of the day.

Filtering Criteria:

Orders will be filtered during "**Order Processing**" according to the following criteria:

1. Volume: Checks the 5-day EMA of the volume versus the global settings. There is a max percent volume per trade, and a minimum volume required. If you set both of these globals to zero, they will be disabled.

2. Equity: Check that there is enough equity for the trade. For futures this is determined by adding up the margin required for the new trade and checking existing margin requirements vs. available equity. For stocks this is determined by adding up the full price of the stock. Available equity is determined by either the test equity or the Order Generation Equity if non-zero and generating orders. The Max Margin/Equity parameter is used here. If the margin is greater than the max amount times the available equity, then the trade is filtered. Set to zero to disable any margin checking. Note that if many orders are placed on the same day, they could all be filled even though it will use more equity than available.

3. Risk Manager: If you have a Risk Manager block, Trading Blox executes the canAddUnit script. If the canAddUnit script Rejects the order, then it will be filtered out.

4. Money Manager: If the unitSize script Rejects the order, then it will be filtered out.

If a trade is filtered due to any of the above, there will be an indication in the Filtered Trade Log file which is located in the Results folder if you set the Log Filtered Trades option to true. You can access all results files from the File/Results menu. These types of filters will be noted with an "Order Processing" type, and the date will be the Instrument Date.

Orders can also be filtered during the fill process, and these will be noted with a "Fill Processing" type and the date will be the Test Date.

The reasons for a "**Fill Processing**" reject include:

1) No trading data -- an order was placed for a date without trading data.
2) Lock Limit Day -- with futures by default the simulation will not buy on a up lock limit day or sell on a down lock limit day. Lock limit is defined as H=L. This feature can be turned off in global preferences.
3) Position is locked by Broker Positions -- if you have a broker position for this market and system, then new entry or exit orders cannot be filled.
4) Open past stop price -- The order cannot fill if the open price is already past the protective stop price.

Custom Portfolios

Part VII

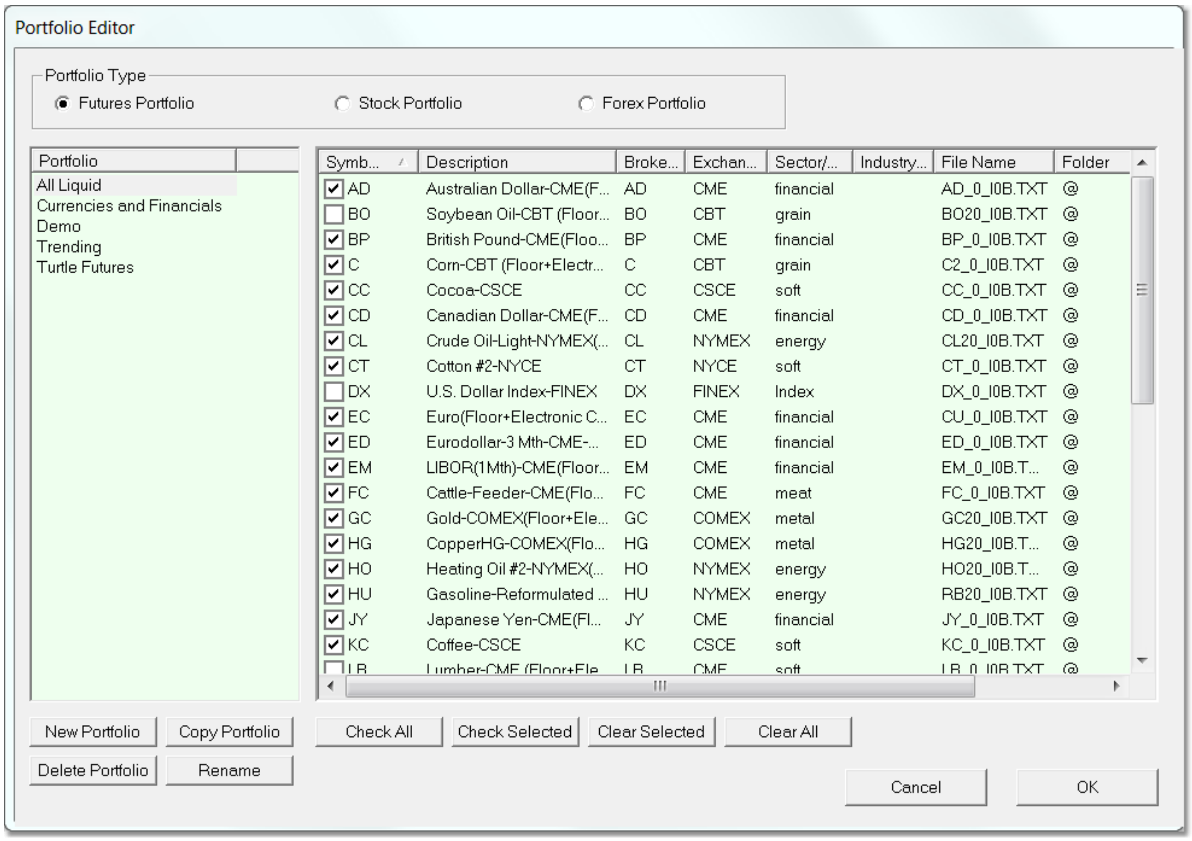
Part 7 – Custom Portfolios

Trading Blox allows users to create customized portfolios of futures, stocks, or forex that can be historically tested, or used to generate daily entry orders. Each system has a portfolio manager section where you can define the portfolio which that system will trade. This allows you to test multiple systems at once - each with a different custom portfolio.

To create new portfolios, change the markets in an existing portfolio, or delete portfolios use the Portfolio Editor.

Section 1 – Portfolio Editor

To activate the Portfolio Editor, select Portfolios menu item from the Edit menu, or use F2.



Portfolio Editor

Portfolio Type: [● Futures Portfolio] ○ Stock Portfolio ○ Forex Portfolio

Portfolio
All Liquid
Currencies and Financials
Demo
Trending
Turtle Futures

	Symb...	Description	Broke...	Exchan...	Sector/...	Industry...	File Name	Folder
✓	AD	Australian Dollar-CME(F...	AD	CME	financial		AD_0_I0B.TXT	@
☐	BO	Soybean Oil-CBT (Floor...	BO	CBT	grain		BO20_I0B.TXT	@
✓	BP	British Pound-CME(Floo...	BP	CME	financial		BP_0_I0B.TXT	@
✓	C	Corn-CBT (Floor+Electr...	C	CBT	grain		C2_0_I0B.TXT	@
✓	CC	Cocoa-CSCE	CC	CSCE	soft		CC_0_I0B.TXT	@
✓	CD	Canadian Dollar-CME(F...	CD	CME	financial		CD_0_I0B.TXT	@
✓	CL	Crude Oil-Light-NYMEX(...	CL	NYMEX	energy		CL20_I0B.TXT	@
✓	CT	Cotton #2-NYCE	CT	NYCE	soft		CT_0_I0B.TXT	@
☐	DX	U.S. Dollar Index-FINEX	DX	FINEX	Index		DX_0_I0B.TXT	@
✓	EC	Euro(Floor+Electronic C...	EC	CME	financial		CU_0_I0B.TXT	@
✓	ED	Eurodollar-3 Mth-CME-...	ED	CME	financial		ED_0_I0B.TXT	@
✓	EM	LIBOR(1Mth)-CME(Floor...	EM	CME	financial		EM_0_I0B.T...	@
✓	FC	Cattle-Feeder-CME(Flo...	FC	CME	meat		FC_0_I0B.TXT	@
✓	GC	Gold-COMEX(Floor+Ele...	GC	COMEX	metal		GC20_I0B.TXT	@
✓	HG	CopperHG-COMEX(Flo...	HG	COMEX	metal		HG20_I0B.T...	@
✓	HO	Heating Oil #2-NYMEX(...	HO	NYMEX	energy		HO20_I0B.T...	@
✓	HU	Gasoline-Reformulated ...	HU	NYMEX	energy		RB20_I0B.TXT	@
✓	JY	Japanese Yen-CME(Fl...	JY	CME	financial		JY_0_I0B.TXT	@
✓	KC	Coffee-CSCE	KC	CSCE	soft		KC_0_I0B.TXT	@
☐	LB	Lumber-CME (Floor+Ele	LB	CME	soft		LB_0_I0B.TXT	@

The top portion of the Portfolio Editor selects the portfolio type: Future, Stock, or Forex. Clicking on a radio button will cause Trading Blox to load all the Portfolios of that type into the Portfolio List on the left side of the Portfolio Editor.

Selecting a Portfolio from the Portfolio List will update the Market Selection Area on the right side of the Portfolio Editor. Each market included in a given Portfolio will have it's corresponding Market Selection Checkbox checked. To add a market to a new Portfolio, click in the Market Selection Checkbox. Uncheck the box to remove a market from a Portfolio.

Where does the list of available instruments come from?
The only instruments that will be available to select into a portfolio are those in the current data folder selected or those that were already in the portfolio. To setup different portfolios for different data folders, you can choose the data folder, then come here to setup the portfolio. The portfolio will not be affected when you change to another data folder.

New Portfolio
Creates a new Portfolio with no markets selected.

Delete Portfolio
Removes a Portfolio.

Copy Portfolio
Copies the portfolio.

Rename
Changes the name of the selected Portfolio.

Check All
Checks all the available markets.

Check Selected
Checks the highlighted (selected) markets.

Clear All
Clears the checks from all the available markets..

OK
Saves and exits.

Cancel
Discards changes and exits.

Section 2 – Sample Portfolios

Trading Blox comes with four sample portfolios for the U.S. futures markets, one sample stock portfolio of U.S. equities for the *NASDAQ 100*, and two sample forex portfolios. The futures portfolios are *Trending*, *Currencies and Financials*, *Turtle Futures*, and *Demo*.

The following table shows which futures markets are included in each of the four different futures portfolios.

Instrument	Trending	Currency & Financials	Turtle Futures	Demo
British Pound	☒	☒	☒	☒
Canadian Dollar		☒	☒	☒
Cocoa			☒	
Coffee	☒		☒	
Corn	☒		☒	☒
Cotton				☒
Crude Oil	☒		☒	☒
DM/Euro	☒	☒	☒	☒
Eurodollar	☒	☒	☒	☒
Gold			☒	
Heating Oil			☒	☒
HG Copper			☒	
Japanese Yen	☒	☒	☒	☒
Mexican Peso	☒	☒	☒	☒
Natural Gas			☒	
Palladium				☒
Pork Bellies				☒
Silver			☒	
Soybeans			☒	
Soybean Oil			☒	☒
Soybean Meal			☒	
Sugar	☒		☒	
Swiss Franc		☒	☒	
10 year T-Note		☒		☒
Unleaded Gas			☒	☒
U.S. T-Bonds		☒	☒	
Wheat			☒	

Log Files

Part VIII

Part 8 – Log Files

Trading Blox also creates several detailed logs when a test is run. Output is stored in a separate folder called "Results".

You can quickly open this folder from the File Menu using the Results submenu. You can also navigate directly to the Results folder and open each of the output files. These files are described in detail later in this section. Be sure to close them or Save As before running another test or Trading Blox will complain.

Depending on the preferences that you set for reporting, data may not be logged to these files. See the section on Report Content Specifications.

Trade Log
A complete list of each trade in text format.

Filtered Trade Log
A complete list of each filtered trade.

Daily Equity Log
A daily log of the major equity values for each trading day.

Monthly Log
Shows the equity for each month.

Print Output File
A file in which all PRINT statements go when using the PRINT statement in Blox Basic Scripting.

Section 1 – Trade Log

The Trade Log contains an entry for every single trade in a test. The file is kept in the Results folder along with the other result files. The name of the file is "Trade Log.csv". An entry is placed in the trade log each time a trade closes. No open trades are logged here. The trades are logged in order of their completion. In order that it will always reflect the most recently executed test, the Trade Log is overwritten each time a new test is run.

Note that if "Log Trades" is disabled in Preferences (see Report Content Specification) nothing will be logged in this file. For large tests with thousands of parameter combinations, disabling this option will result in faster tests.

The Trade Log includes the following columns for any given trade:

Log Column Name:	Description:
Test	The test number when doing multiple parameter runs
System Number	System number based on position in System List
System Name	The system name
Symbol	Future/Stock symbol
Unit	Unit Number
Position	Position: Long or Short
Entry Date	Date the entry was logged
Entry Time	Time the entry was logged
Entry Order	Entry order price (price which initiated entry)
Stop Price	Predefined stop price
Risk %	Measurement of risk in currency, based on the entry and stop
Quantity	Number of shares or contracts
Entry Fill	The "actual" entry price after accounting for slippage
Entry DPP	The Dollars Per Point (actually system wide currency per point) used on the day of entry.
Entry Stock Split Ratio	The Stock split ratio on the day of entry.
Exit Date	Date the trade was closed
Exit Time	Time the trade was closed
Exit Order	Exit order price (price which initiated exit)
Exit Fill	The "actual" exit price after accounting for slippage
Exit DPP	The Dollars Per Point (actually system wide currency per point) used on the day of exit.
Dividends	The dividends accrued on this trade, if the dividend files are available.
Commission	The total commission for this trade.
Profit	Listed in system wide base currency
Profit Percent	Profit as a percentage of Base Equity. The base equity used is from the end of the day prior to the trade entry.

Closed Equity	New account balance after adding this trade's profit
Total Equity	*Total Equity* at time of exit
Draw down	Drawdown
Days in Trade	Number of days between entry and exit
R Multiple	The R Multiple of the trade
Custom Value	The custom value set for the trade in scripting
Rule Label	The Rule Label as set in scripting

Section 2 – FilteredTradeLog

The log also contains entries for each trade that was Filtered, for whatever reason. This is extremely useful if you find yourself asking, "Why didn't my system trade on this day when I expected it to?" There is a preference to have filtered trades included in this output or not. Be sure to turn this on if you want to see the filtered trades. Lines for trades not taken could look like the following:

- For system 8, symbol ADBE on 1995-03-31, trade attempted but Money Manager set a unitSize of zero
- For BIIB on 1995-03-31, the volume of 3666 is less than required 10000
- For FLEX on 1995-11-20, cannot add 99300.00 additional margin to 978255.01 existing with only 998343.31 equity.

There are many reasons why a trade may not be taken, and you will see the reason in this file.

Log Column Name:	**Description:**
Test Number	The test number when running multiple parameter runs
Type of filter	Either Order Processing or Fill Processing
System Name	System Name
Symbol	Instrument symbol
Position	Either Long or Short
Quantity	The potential quantity for the order
Date	The date this order was rejected. Note that this is the instrument date for Order Processing types and the test date for the Fill Processing types.
Message	The reason the order was rejected.

See the Order Filter for additional information.

Section 3 – Daily Equity Log

The Daily Equity Log contains an entry for every single day of the test showing summary equity information as well as a limited summary of the open positions for each day. You can use the Equity Log to determine if a trade that appears like it should have been taken wasn't taken because Purchase Equity or Margin Equity was too high. The file is named "Daily Equity Log.csv" and is stored in the Results folder.

An entry is placed in the Equity Log for each trading day in a given historical test. This daily entry includes data on all open positions, for each day of the simulation. In order that it will always reflect the most recently executed test, the Equity Log is overwritten each time a new test is run.

The Equity Log includes the following columns:

Log Column Name:	Desription:
Test Number	The test number
Date	Includes trading dates only
Total Equity	*Total Equity* is *Closed Equity* plus *Open Equity*
Closed Equity	*Closed Equity at month end*
Drawdown	Drawdown in Total Equity
Open Equity	*Open Equity* is the value of open positions marked to market
Cash	*Closed Equity* minus Purchase Equity
Purchase Equity	Equity used for Stock purchases
Margin Equity	Equity used as Margin for Futures purchases
Total Slippage	Total Slippage to date
Total Dividends	Dividends earned test to date.
Total Earned Interest	Interest earned test to date.
Total Margin	Margin paid test to date.
Total Commissions	Commissions test to date.
Total Carry	Forex carry cost test to date.

Section 4 – Monthly Equity Log

The Monthly Equity Log contains an entry for each month that is tested, providing a month-by-month view into the results. It is named "Monthly Equity Log.csv" and is stored in the Results folder. The log entry is placed at the end of the month. This log is overwritten each time a new test is run such that it will always reflect only the most recently executed historical simulation test.

Monthly Equity Log includes the following columns:

Log Column Name:	**Description:**
Test	Test number
Year	Year of the month entry
Month	Month of the entry
Days	Number of days in the month
Closed Balance	*Closed Equity* Balance at month end
Total Equity	*Total Equity* at month end
Total Equity Gain	*Total Equity* gain from the previous month in system wide base currency
Gain%	Percentage *Total Equity* gain from the previous month
C Risk %	Closed Equity Risk
T Risk %	Total Equity Risk
CT DD%	Closed Equity Drawdown
TE DD%	Total Equity Drawdown
#Trades	Number of Trades for this month

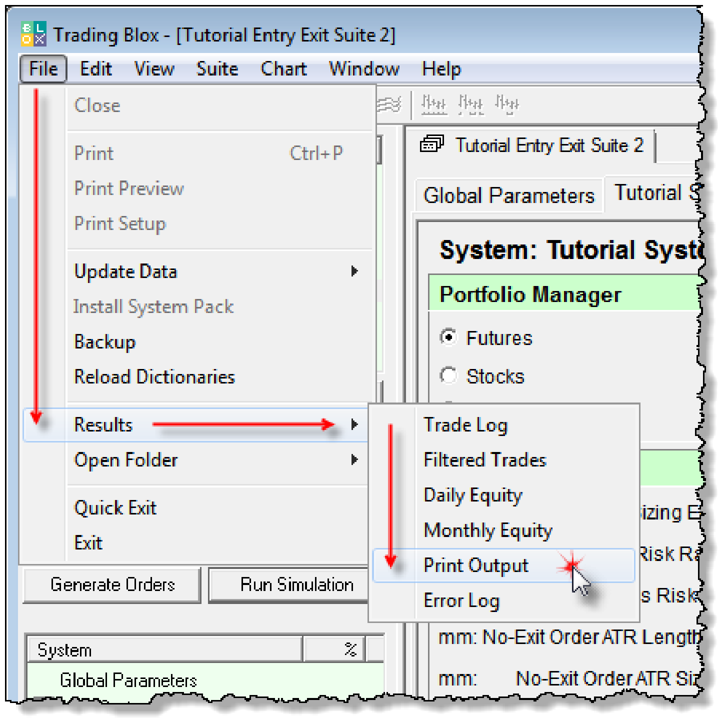
Section 5 – PrintOutput.csv

PrintOutput.csv is a text file that is re-written every time a test is run.

This file is used to record the output from all PRINT statements (only applicable for Trading Blox Builder.

Most often the data contained in this file is comma delimited for easy use in other programs like Excel. It is also easily opened with the Windows Notepad.

Print Output.csv file is stored in the Trading Blox Results folder, and it can be accessed by click on the main screen's File menu, and then clicking on the Results sub-menu under the File menu, and then on the next sub-menu where PrintOutput appears.



Part IX

Part 9 – Test Results

The Summary Test Results are shown after a test completes and can be found on the Summary Results tab of the main Results Window (see the Test Results user interface overview for more information). A copy of the Summary Test Results is also stored in the Results folder. Each time a test is run, a separate Summary Test Results file (and folder containing images) with a unique file name is created. Results are never overwritten, so that a permanent record of each test is maintained.

Trading Blox names each Summary Test Results file according to the unique date and time at which the test was executed. The naming convention is as follows:

<Suite Name> yyyy-mm-dd_hh_mm_ss.htm

For example, a file with the name "Test 2003-04-01_14_53_01.html" was run on April 1, 2003, and completed at 14:53:01 (or 2:53 PM).

You can move these files out of this folder, or send them to others. However, you should also move the corresponding folder to the same place because it contains the images for the reports. The test .htm file is named according to the suite. The corresponding folder is named "Test" with the same timestamp as the .htm file. For the above example, the folder would be named "Test 2003-04-01_14_53_01.html".

If you run many tests with many different parameter combinations Trading Blox will generate many results files. If you find your hard drive space is getting low, you can delete older unused results files and their associated directories.

The reports in Trading Blox include:

Stepped Parameter Performance
To quickly see the effects of parameter changes on system results

Equity Graphs
Log-scale and linear-scale equity graphs to see system performance over time

Monthly Return Graphs
Shows the monthly returns as well as the distribution of monthly returns by Percentage.

Total Risk Profile
Shows the amount of risk or portfolio heat carried over time

R-Multiple™ Graphs
Shows the distribution of trades by R-Multiple™ as well as the profit contribution of the trades by R-Multiple™.

Yearly Performance Summary
Shows the performance of the system broken down by year

Trading Performance Statistics
CAGR%, MAR ratios, etc. Trading Blox has the most extensive trading system performance statistics available

Instrument Performance Summary
A breakdown of the performance by instrument

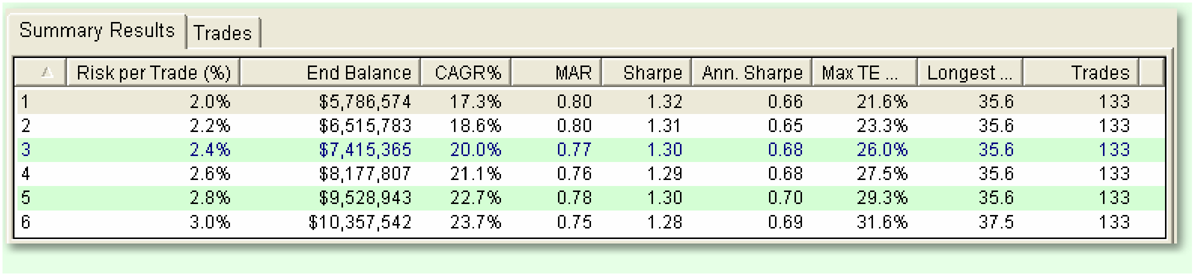
System Parameter Settings
Shows the exact parameters used to generate the results so you can reproduce results or compare against other results

Trade List with Charts
A complete set of each trade drawn on a chart for quick perusal of each trade responsible for the results

You can specify which of these reports is included in the Summary Test Results using the Preferences for reporting (see: Report Content Specification for details).

Section 1 – Stepped Parameter Performance

The Stepped Parameter Summary Performance List is displayed at the top of the results list:



	Risk per Trade (%)	End Balance	CAGR%	MAR	Sharpe	Ann. Sharpe	Max TE ...	Longest ...	Trades
1	2.0%	$5,786,574	17.3%	0.80	1.32	0.66	21.6%	35.6	133
2	2.2%	$6,515,783	18.6%	0.80	1.31	0.65	23.3%	35.6	133
3	2.4%	$7,415,365	20.0%	0.77	1.30	0.68	26.0%	35.6	133
4	2.6%	$8,177,807	21.1%	0.76	1.29	0.68	27.5%	35.6	133
5	2.8%	$9,528,943	22.7%	0.78	1.30	0.70	29.3%	35.6	133
6	3.0%	$10,357,542	23.7%	0.75	1.28	0.69	31.6%	37.5	133

Stepped Parameter Summary Performance List

Run
A sequential count that keeps track of the number of distinct parameter combinations being tested.

End Balance
The account balance at the completion of each run.

CAGR%
The compounded annual growth rate (geometric mean return).

MAR
A ratio obtained by dividing the CAGR% by Max Total Equity Drawdown. The higher the MAR ratio, the better the risk-adjusted performance.

Sharpe
This version of the Sharpe Ratio divides annualized return by annualized standard deviation of returns, using monthly data points. Excludes the risk-free-rate in the numerator.

Ann. Sharpe
This version of the Sharpe Ratio divides annual return by standard deviation of annual returns, using actual annual calendar data points (The Sharpe Ratio is the classic measure of return versus. risk. Divides excess return by standard deviation to determine reward per unit of risk. The higher the Sharpe ratio, the better the risk-adjusted performance.)

Max TE DD
Max Total Equity Drawdown - a one-time event that reflects the largest equity retracement-relative to a previous equity high-in the entire simulation. Expressed as a percentage, this statistic is measured peak-to-valley, using Total Equity.

Longest TE DD
Longest Total Equity Drawdown - a one time event that reflects the longest duration drawdown in Total Equity over the life of the simulation. It is measured from previous equity peak to new equity peak, and expressed in months.

# Trades
Number of round-trip trades placed over the course of the entire simulation.

Custom Statistics
If your system includes custom statistics they will show up here.

Clicking on any row in the Stepped Parameter Summary Performance List will bring the detailed reports for that particular parameter run to the Detailed Test Results area of the Test Results Window.

Section 2 – Multi-parameter Surface Chart

When you run two or more stepped parameters in the same test, the results will be shown in a contoured surface chart. All combinations of the steps will be graphed on the x and y axis. The average selected Goodness measure will be shown as various colors.

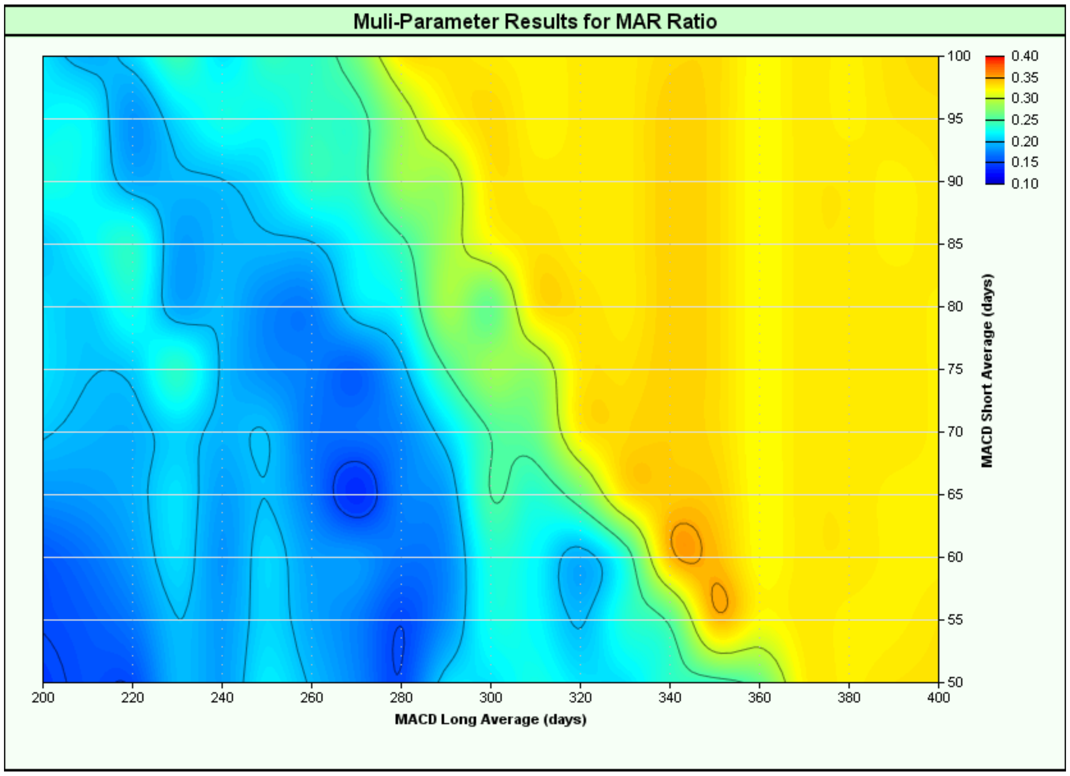
You can select which goodness measure to use in Preferences, and you can set the scaling to be automatic or manual.

If you run a stepped test with more than two parameters, the goodness measure will be **averaged** for all results at a given x and y axis.

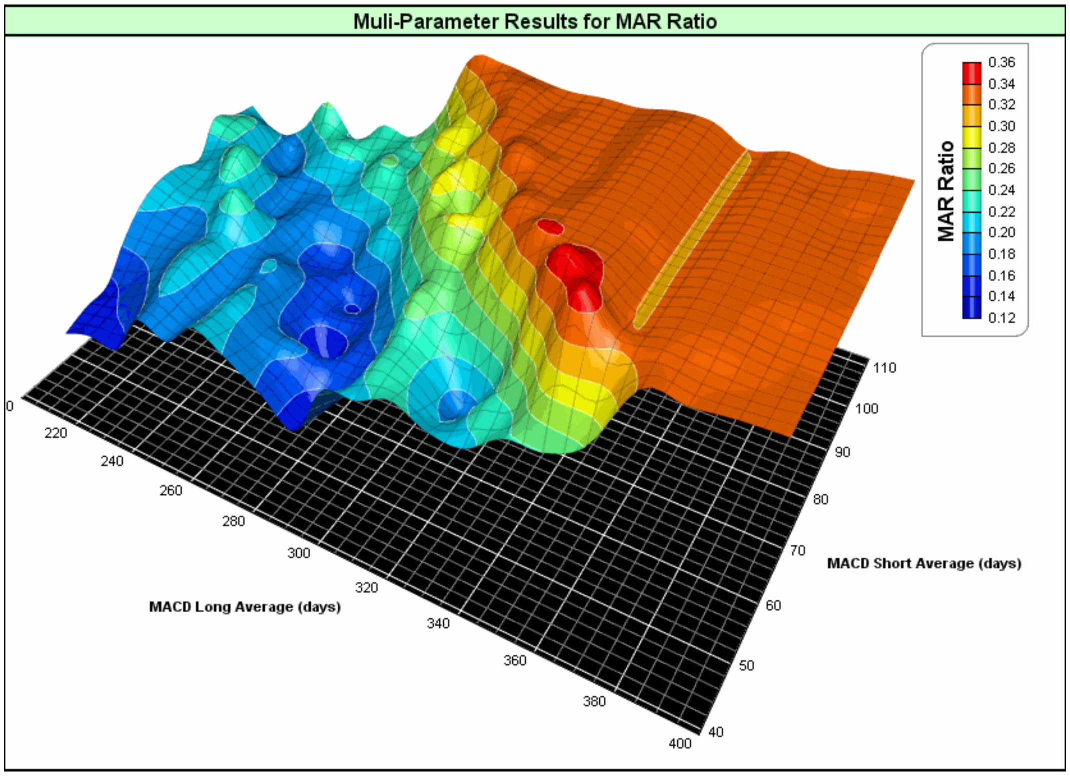
Multiple charts can be produced simultaneously in one simulation by use the `test.SetGoodnessToChart()` function in Blox Basic.

Each chart can be turned on or off in the TradingBlox.ini file by setting the Multi Parameter 1 and Multi Parameter 2 parameters to true or false. This is useful if you prefer one chart to the other, and want to turn one off.

```
   Multi Parameter Chart 1 = FALSE
   Multi Parameter Chart 2 = TRUE
```

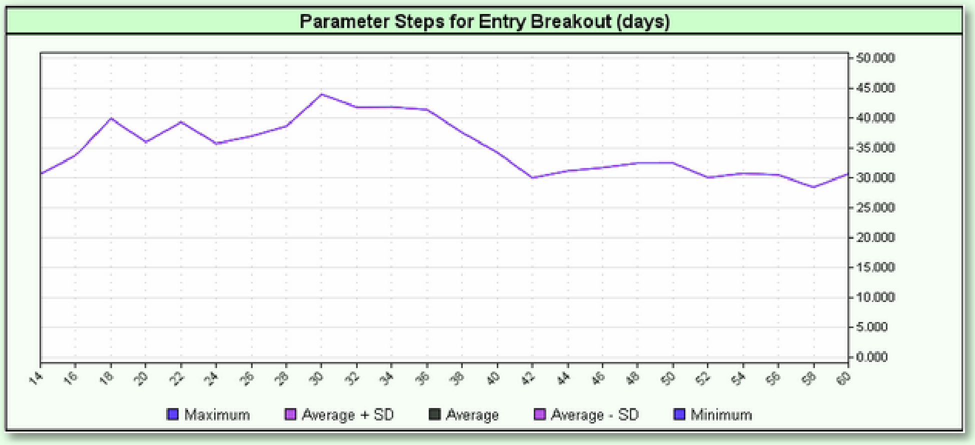


In addition, the Blox Builder Edition provides a 3D look at multi parameter chart:


Muli-Parameter Results for MAR Ratio
MAR Ratio
0.36
0.34
0.32
0.30
0.28
0.26
0.24
0.22
0.20
0.18
0.16
0.14
0.12
MACD Long Average (days)
0
220
240
260
280
300
320
340
360
380
400
MACD Short Average (days)
110
100
90
80
70
60
50
40

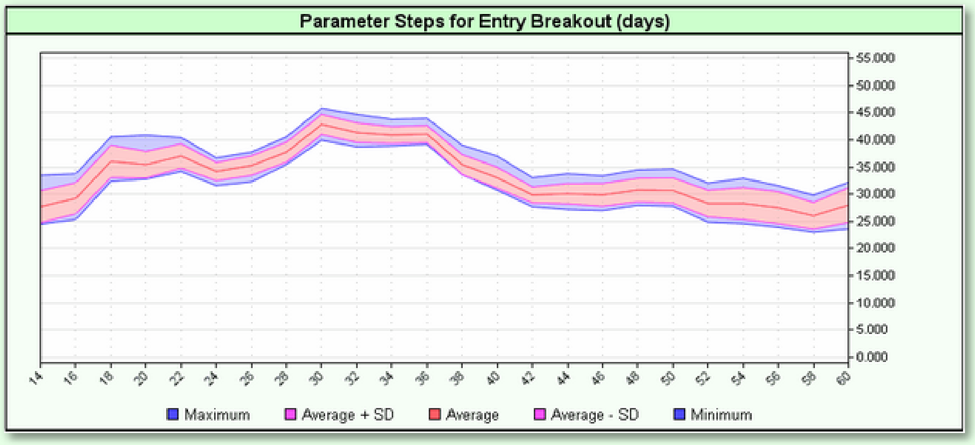
Section 3 – Parameter Stepping Graph

One of the major goals of any system testing effort is the determination of the optimal set of parameters to trade. The Parameter Stepping Graph allows one to see how the values of a given summary measure varies as the values for a particular parameter varies.



For a single parameter test run, the Parameter Stepping Graph shows a line indicating the values of the measure for each step in the test. For example, the above graph for a test stepping the Donchian System's Entry Breakout parameter shows an almost 45%CAGR% value at 30 days for the Entry Breakout dropping to 30% as the Entry Breakout moves higher to 42.

For tests with more than one parameter, the Parameter Stepping Graph shows a band of values for each parameter step.



The above graph shows another test that also stepped the Exit Breakout from 6 to 14. This means that for each of the steps on the Entry Breakout there are nine different steps for the Exit Breakout. The graph plots the average, minimum, maximum and a band that is one standard deviation above and below the average for each value.

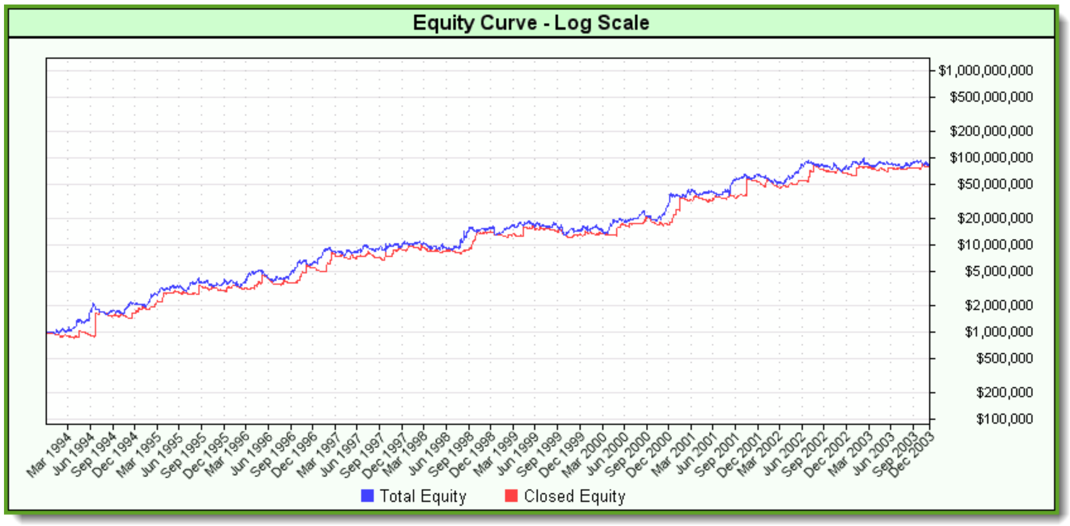
The above graph shows that for an Entry Breakout of 30, the MAR varies from just below 40% CAGR to just above 45% CAGR%.

A narrow band around the average indicates that the other parameters affected the results in a relatively minor way. A wide band indicates that the other parameters affected the test in a large way.The wider the band, the greater the influence the other parameters had on the test measure.

The preferences item Measure of Goodness Index from the Reporting General preference page controls the measure used for the Parameter Stepping Graphs. In the above examples, this value was set to CAGR%.

Section 4 – Log Equity Graph

Following the Summary Test Results, a series of graphs is displayed.



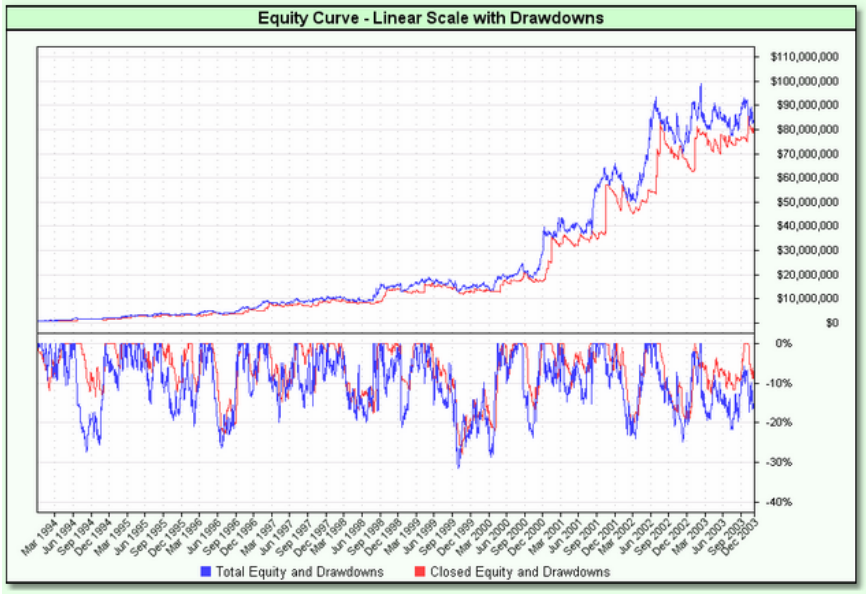
The Log Scale Equity Graph displays the *Total Equity* and *Closed Equity* on a logarithmic scale (with time on the horizontal axis), providing a view into how the overall account value changes over time, as well as how *Total Equity* and *Closed Equity* vary in relation to one another over the same time period.

Total Equity is plotted in blue; *Closed Equity* is plotted in red. In the world of finance, red is often used to represent a loss. Here, it is used only to distinguish between the *Total Equity* and *Closed Equity*, and has no other connotation.

On a logarithmically-scaled Equity chart, a given distance (on the vertical axis) always represents an equal percentage change. For instance, on the chart below, the distance from $100K to $1M is the same as the distance from $1M to $10M. They both represent the same percentage change (in this case 1,000%).

Section 5 – Linear Equity Drawdown Graph

The Linear Equity Graph shows the equity over time measured on a linear scale with the drawdowns over the same period plotted below that graph.



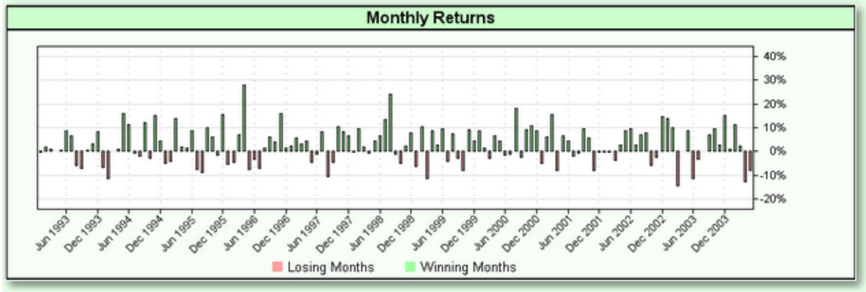
On the Linear Equity Graph, both *Total Equity* and *Closed Equity* are plotted over time on the top half of the graph. This graph has a vertical scale that is linear, or arithmetic, as opposed to logarithmic scale of the Log Scale Equity Graph.

On a linear-scaled Equity chart, a given distance (on the vertical axis) always represents the same absolute change in system wide base currency. For instance, on the chart above, the distance between $10,000,000 - $20,000,000 is the same as the distance between $20,000,000 - $30,000,000. While linear scale Equity Curve graphs tend to lose resolution in the earlier years of an extended study (as can be seen above), most of us are accustomed to looking at linear representations, and they do have their uses.

In the bottom half of the graph, *Total Equity* and its Drawdown are plotted in blue; *Closed Equity* and its Drawdown are plotted in red. Drawdown is a retracement from a previous Equity peak, and is expressed here in percentage terms. This type of graph is also sometimes referred to as an Underwater Equity curve.

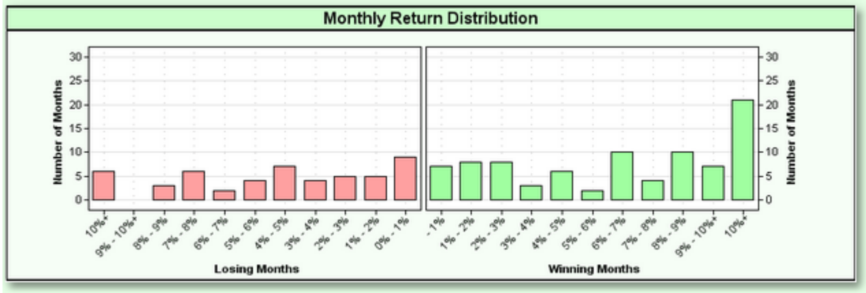
Section 6 – Monthly Returns Graph

The Monthly Returns graph shows the returns for each month of the test. The graph is automatically resized so that you can quickly see each month's individual performance.



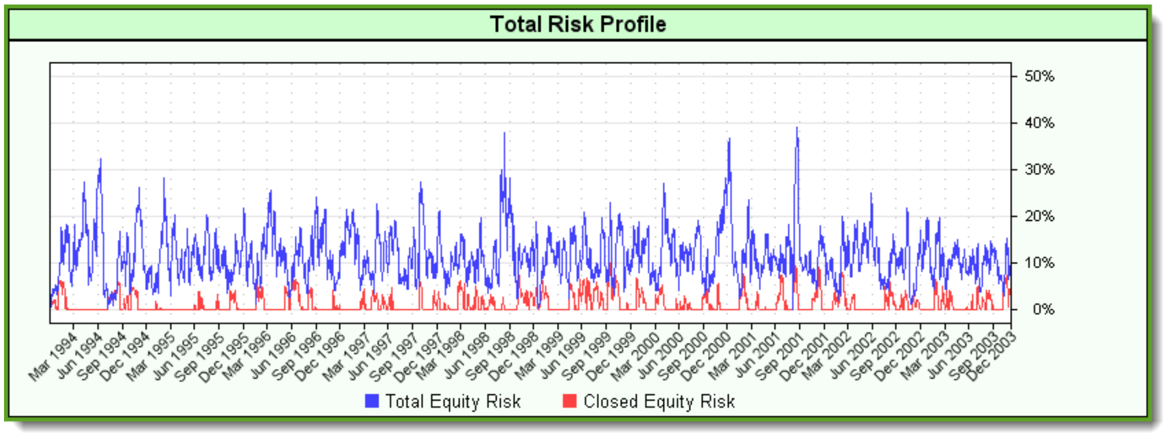
Section 7 – Monthly Returns Distribution Graph

The Monthly Return Distribution graph shows the frequency distribution of the returns. Each bar shows the number of times that the monthly returns fall within a specified range. In the above example, you can see that there were 10 months where the returns were between 8% and 9%.



Section 8 – Total Risk Profile Graph

Total Risk examines the relationship between *Total Equity* and *Closed Equity* to Open Risk (respectively). The Total Risk Profile graph displays Total Equity Risk and Closed Trade Risk plotted over time.



The following definitions will help illustrate the difference between *Total Equity* and *Closed Equity* risk.

Closed Equity
Starting Balance plus the cumulative profit (or loss) from all closed-out trades. (Sometimes referred to as Closed Trade Equity.)

Open Equity
Total profit (or loss) of all open positions

Total Equity
Closed Equity + Open Equity

Open Risk
The (currency) distance from the current price to the closest stop, for all open positions. It is based on the assumption that all open positions will be exited at their current designated stops (though some or all of these positions may prove profitable in the future).

Closed Risk
The (currency) distance from the entry price to the closest stop (either entry or trailing), for all open positions. If the trailing stop is profitable then the closed risk on a position is zero.

Locked-In Profits
Open Equity - Open Risk. (The profit on a trade in progress becomes "locked-in" when the trailing stop moves favorably past the entry price.)

Total Equity Risk
Open Risk / Total Equity

Closed Equity Risk
Closed Risk / Closed Equity

Please consider the following example of *Total Equity Risk*:

Total Equity	$150,000
Closed Equity	$100,000
Open Equity	$50,000
Open Risk	$40,000
Locked-in Profits	$10,000
Closed Risk	$0
Total Equity Risk	Open Risk / Total Equity = $40,000 / $150,000 = 26.67%

Closed Equity Risk, while similar to *Total Equity Risk*, is impacted only by locked-in losses. (Thus, it is 0% in the example above.) The plot of *Closed Equity Risk* will only assume a positive value when *Closed Risk* is positive. *Closed Risk* is positive when the *Locked-In Profits* are less than zero. If there are *Locked-In Profits* then the *Closed Risk* is zero and the *Closed Equity Risk* is zero.

Now consider how *Closed Equity Risk* differs:

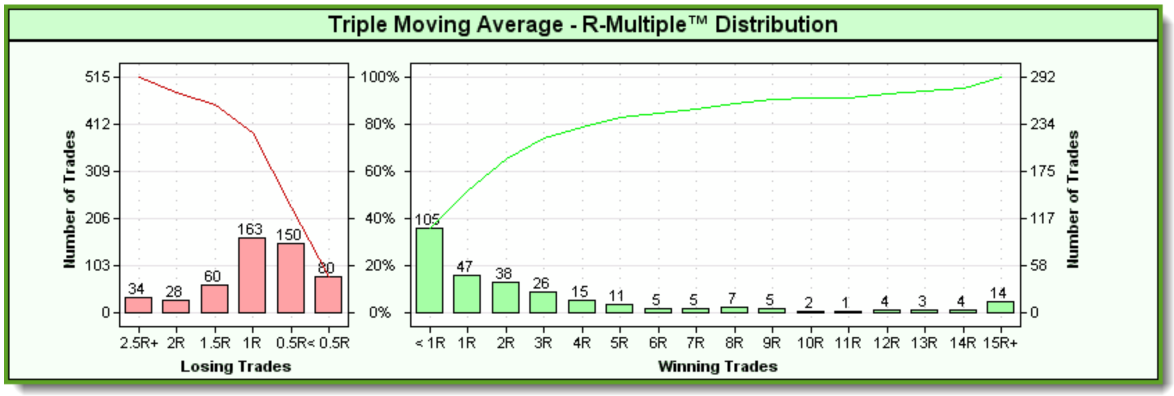
Total Equity	$150,000

Closed Equity	$100,000
Open Equity	$50,000
Open Risk	$60,000
Locked-in Profits	-$10,000
Closed Risk	$10,000
Closed Equity Risk	Closed Risk / Closed Equity = $10,000 / $100,000 = 10.00%

Section 9 – R-Multiple Distribution Graph

The concept of an R-Multiple™ was pioneered in 1993 by trader Chuck Branscomb, who explains that this technique is "The most important way to look at systems." The idea came about as a way to equate all markets and get away from looking at expectation in currency terms." R-Multiple™s were popularized by Dr. Van Tharp in his book, "*Trade Your Way to Financial Freedom*".

An R-Multiple™ is simply the profit or loss for a given trade divided by the entry risk. The entry risk is defined as the difference between the entry price and the stop price at the time of the entry.



The R-Multiple™ Distribution graph accounts for every closed trade in the simulation, both winning (green) and losing (red) trades. For the test reflected in the graph above, there were a total of 807 trades. Of 292 winning trades, 105 fell in the range between 0R < 1R. Continuing to the right, 47 winning trades had R-Multiple™s between 1R < 2R, and 38 trades fell in the bin of trades in the range 2R < 3R, etc. Note that 14 winning trades had R-Multiple™s of 15R or greater.

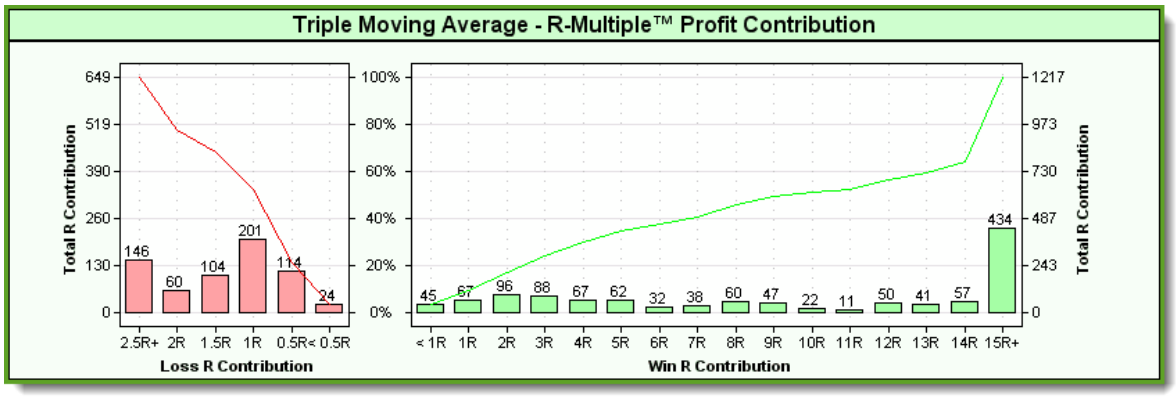
The histogram of Losing Trades (above, left) shows that the vast majority of losers were of a magnitude of -1.5R or less, with 393 trades (out of a total of 515 losing trades) less than -1.5R.

The thin lines above the bars of the histogram are cumulative plots. Note that two scales apply to the vertical axes: on the outside is the Number of Trades, and on the inside is the Cumulative Percentage. For Winning Trades, the starting point of the cumulative plot (<1R on the horizontal axis) contains 105 trades. But the next data point (the 1R < 2R bin on the horizontal axis) grows to 152 trades (105+47) on the right vertical axis, and encompasses slightly more than 50% of all winning trades, according to the left vertical axis.

As can be clearly seen, the Losing Trades were relatively well contained: The cumulative plot line shows at a glance that roughly 90% of losers were of a magnitude of -1.5R or less.

Section 10 – R-Multiple Contribution Graph

The R-Multiple™ Profit Contribution graph in the series shows Profit Contribution as a function of R-Multiple™s. Like the R-Multiple™ Distribution graph, it accounts for every closed trade in the simulation, both winning (green) and losing (red) trades.



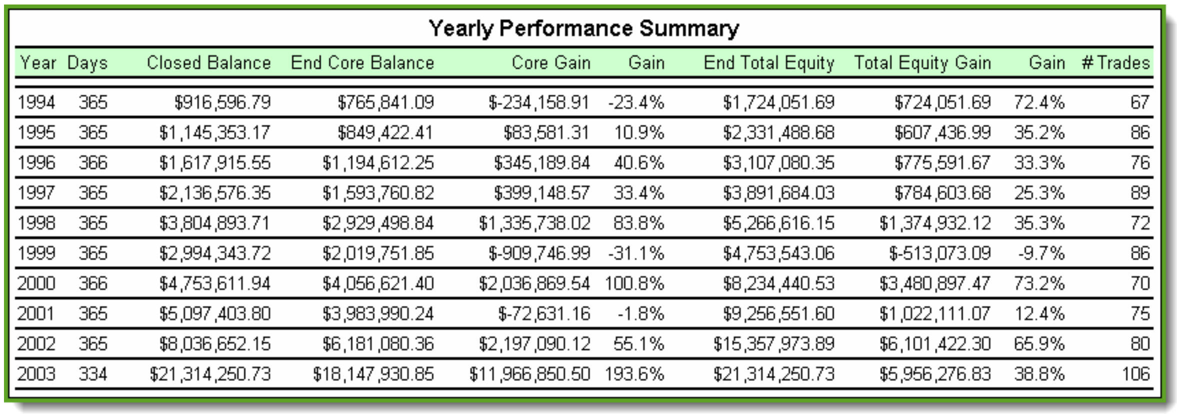
Each bar of the histogram simply represents the *sum* of the R-Multiple™ *value* of all the trades in that particular bin.

For instance, the R-Multiple™ Profit Contribution for the 14R bin below (which contains all trades in the range 14R < 15R) ...is 57. This comes from 4 trade with slightly more than 14 each, 4 trades x 14R = 56R.

As in the previous graph, the thin lines above the bars of the histogram are cumulative plots. Again, note that two scales apply. On the right vertical axis is the Total R Contribution, on the left vertical axis is the Cumulative Percentage, and they be interpreted in the same way as in the preceding example: For the test reflected in the chart below, about 60% of the total profits-in R-came from the cumulative contribution of the bins of 8R or greater.

Section 11 – Yearly Performance Summary

The Yearly Performance Summary shows the performance for each individual year of the test.



Yearly Performance Summary

Year	Days	Closed Balance	End Core Balance	Core Gain	Gain	End Total Equity	Total Equity Gain	Gain	# Trades
1994	365	$916,596.79	$765,841.09	$-234,158.91	-23.4%	$1,724,051.69	$724,051.69	72.4%	67
1995	365	$1,145,353.17	$849,422.41	$83,581.31	10.9%	$2,331,488.68	$607,436.99	35.2%	86
1996	366	$1,617,915.55	$1,194,612.25	$345,189.84	40.6%	$3,107,080.35	$775,591.67	33.3%	76
1997	365	$2,136,576.35	$1,593,760.82	$399,148.57	33.4%	$3,891,684.03	$784,603.68	25.3%	89
1998	365	$3,804,893.71	$2,929,498.84	$1,335,738.02	83.8%	$5,266,616.15	$1,374,932.12	35.3%	72
1999	365	$2,994,343.72	$2,019,751.85	$-909,746.99	-31.1%	$4,753,543.06	$-513,073.09	-9.7%	86
2000	366	$4,753,611.94	$4,056,621.40	$2,036,869.54	100.8%	$8,234,440.53	$3,480,897.47	73.2%	70
2001	365	$5,097,403.80	$3,983,990.24	$-72,631.16	-1.8%	$9,256,551.60	$1,022,111.07	12.4%	75
2002	365	$8,036,652.15	$6,181,080.36	$2,197,090.12	55.1%	$15,357,973.89	$6,101,422.30	65.9%	80
2003	334	$21,314,250.73	$18,147,930.85	$11,966,850.50	193.6%	$21,314,250.73	$5,956,276.83	38.8%	106

The following table lists the columns and a description of their contents:

Year
The Calendar Year for which the line of information applies.

Days
The number of calendar days in the trading year. This column will have a value of 365 for complete years (366 for Leap Years, less for partial years at the beginning or end of the simulation).

Closed Balance
The end-of-year Closed Equity balance. Composed of the Starting Balance, plus the cumulative profit (or loss) from all closed-out trades during the calendar year.

End Total Equity
Total value of the investment account for the year (includes open positions using current market prices).

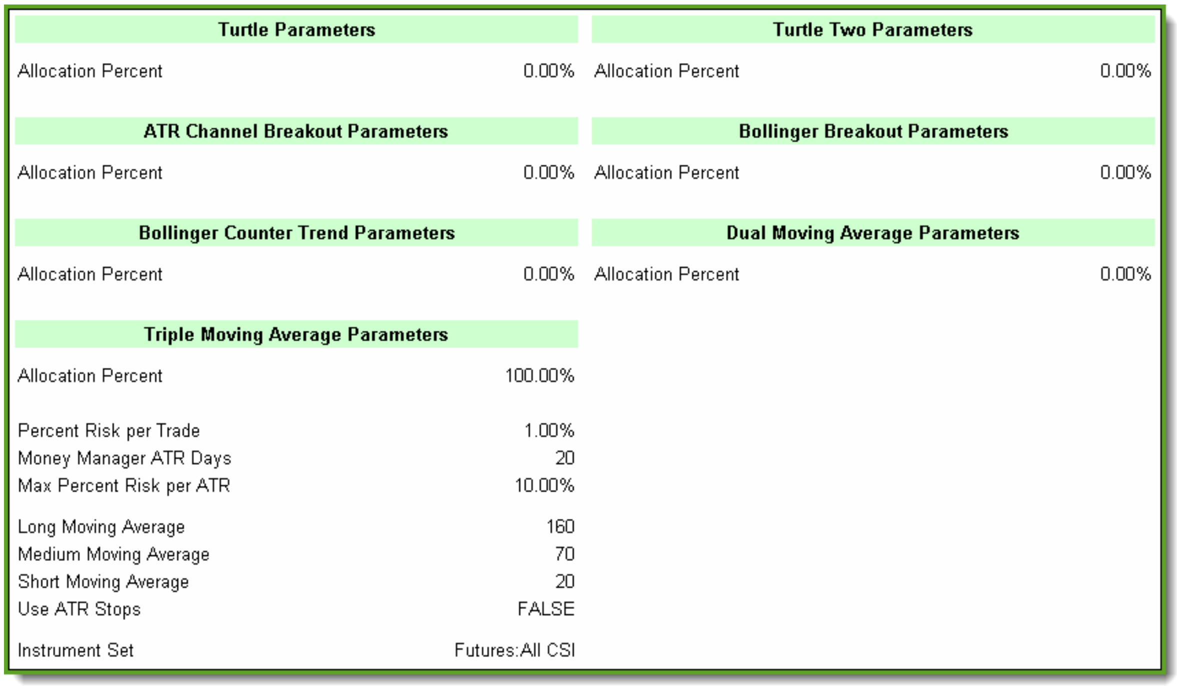
Total Equity Gain
Equals End Total Equity - Starting Balance. (After the 1st calendar year, Starting Balance is simply End Total Equity from the previous year.)

Gain%
Total Equity Gain divided by Starting Balance for the calendar year. (After the 1st calendar year, Starting Balance is simply End Total Equity from the previous year.)

Trades
The number of trades for the calendar year.

Section 12 – System Parameter Settings

The System Run Parameters section lists the specific values of the parameters for a given test:

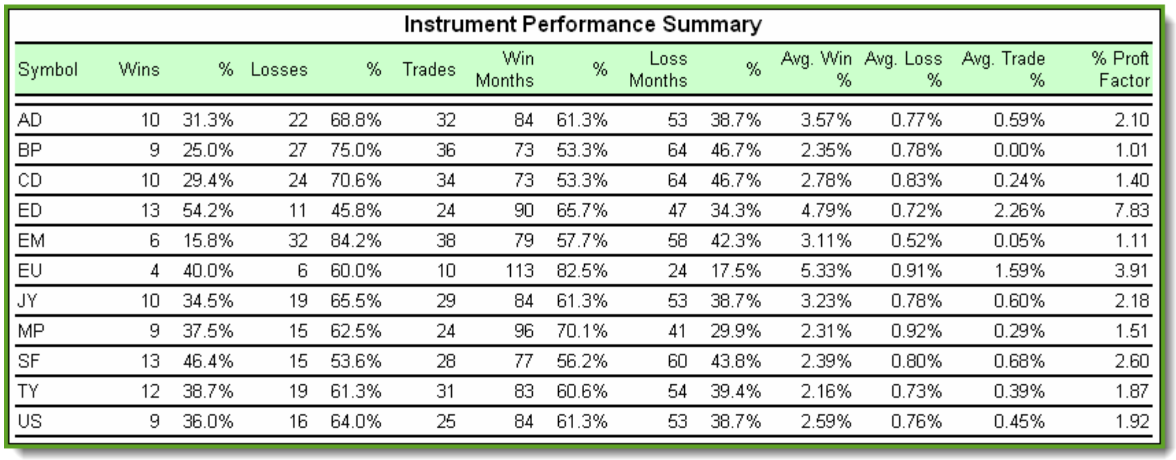


Turtle Parameters		Turtle Two Parameters	
Allocation Percent	0.00%	Allocation Percent	0.00%
ATR Channel Breakout Parameters		**Bollinger Breakout Parameters**	
Allocation Percent	0.00%	Allocation Percent	0.00%
Bollinger Counter Trend Parameters		**Dual Moving Average Parameters**	
Allocation Percent	0.00%	Allocation Percent	0.00%
Triple Moving Average Parameters			
Allocation Percent	100.00%		
Percent Risk per Trade	1.00%		
Money Manager ATR Days	20		
Max Percent Risk per ATR	10.00%		
Long Moving Average	160		
Medium Moving Average	70		
Short Moving Average	20		
Use ATR Stops	FALSE		
Instrument Set	Futures:All CSI		

Notice that this section shows the 0% allocations for each system except the Triple Moving Average system which has a 100% allocation. Since the Triple Moving Average is an active system, the values for the parameters for that system are displayed.

Section 13 – Instrument Performance Summary

The Instrument Performance Summary shows the performance for each individual market for each portfolio in the test.



Instrument Performance Summary

Symbol	Wins	%	Losses	%	Trades	Win Months	%	Loss Months	%	Avg. Win %	Avg. Loss %	Avg. Trade %	% Proft Factor
AD	10	31.3%	22	68.8%	32	84	61.3%	53	38.7%	3.57%	0.77%	0.59%	2.10
BP	9	25.0%	27	75.0%	36	73	53.3%	64	46.7%	2.35%	0.78%	0.00%	1.01
CD	10	29.4%	24	70.6%	34	73	53.3%	64	46.7%	2.78%	0.83%	0.24%	1.40
ED	13	54.2%	11	45.8%	24	90	65.7%	47	34.3%	4.79%	0.72%	2.26%	7.83
EM	6	15.8%	32	84.2%	38	79	57.7%	58	42.3%	3.11%	0.52%	0.05%	1.11
EU	4	40.0%	6	60.0%	10	113	82.5%	24	17.5%	5.33%	0.91%	1.59%	3.91
JY	10	34.5%	19	65.5%	29	84	61.3%	53	38.7%	3.23%	0.78%	0.60%	2.18
MP	9	37.5%	15	62.5%	24	96	70.1%	41	29.9%	2.31%	0.92%	0.29%	1.51
SF	13	46.4%	15	53.6%	28	77	56.2%	60	43.8%	2.39%	0.80%	0.68%	2.60
TY	12	38.7%	19	61.3%	31	83	60.6%	54	39.4%	2.16%	0.73%	0.39%	1.87
US	9	36.0%	16	64.0%	25	84	61.3%	53	38.7%	2.59%	0.76%	0.45%	1.92

The following table lists the columns and a description of their contents:

Symbol
The symbol for the market.

Wins
The number of winning trades and the percentage of the total trades for a given market.

Losses
The number of losing trades and the percentage of the total trades for a given market.

Trades
The total number of trades for a given market.

Win Months
The number of months and the percentage of the total testing months a given market is profitable

Loss Months
The number of months and the percentage of the total testing months a given market is unprofitable

Avg. Win %
The average winning trade return.

Avg. Loss %
The average losing trade return.

Avg. Trade%
The average trade return including both winning and losing trades.

% Profit Factor
The Percent Profit Factor for the given market in isolation.

Section 14 – Trading Performance and Statistics

The following section provides detailed trading performance statistics, as well as a list of the assumptions used for the particular test.

Many of the statistics listed below deal with risk and return. Appendix I - Risk and Return contains a supplement that discusses these concepts in more detail than is permitted here. If you are new to the world of trading system design, or would like an overview of the concepts underlying the use of risk-adjusted return measurements, please skip ahead to Appendix I - Risk and Return before proceeding with this section.

Summary Performance and Ratios

CAGR%
Compounded Annual Growth Rate; this is the annual percentage rate at which an account must continually grow over the period of the study, in order to attain the ending value from the starting value. Synonymous with Geometric Mean Return.

RAR%
Regressed Annual Return.Slope of the linear regression of all the points in the equity curve. See page 186 of Way of the Turtle for more information.

Maximum Total Equity Drawdown%
Max Total Equity Drawdown (Max DD) is a one-time event that reflects the largest retracement relative to a previous equity high in the entire simulation. It is based on a daily, marked-to-the-market assessment. Expressed as a percentage, this statistic is sampled peak-to-valley, using *Total Equity*. It conveys the maximum pain an investor would have had to endure over the life of the study in order to achieve the resulting return.

MAR Ratio
Often referred to as a pain-to-gain ratio, this risk-adjusted return metric was developed by Managed Accounts Review for ranking Commodity Trading Advisors (CTAs).

MAR Ratio = CAGR / Max Total Equity Drawdown

R-Cubed (R3)
Robust Risk/Reward Ratio. uses the RAR% in the numerator and the length-adjusted average maximum draw down in the denominator. The average maximum draw down is computed by taking the five largest draw downs and dividing by 5. The length adjustment is made by taking the average maximum draw down length in days and dividing it by 365 and then multiplying that number by the average maximum draw down. See page 188 of Way of the Turtle for more information.

Modified Sharpe Ratio
The Sharpe Ratio is the classic measure of return vs. risk. Developed by Nobel Laureate and (now) Stanford professor William F. Sharpe, it divides excess return by standard deviation to determine reward per unit of risk. (The higher the Sharpe ratio, the better the risk-adjusted performance.) This modified version divides annualized return by annualized standard deviation of returns, using monthly data points. Exclusion of the risk-free-rate in the numerator makes this measure less sensitive to changes in leverage:

Modified Sharpe Ratio = Annualized Return / Annualized Std Dev

Return = 12* Average Monthly Return

Risk Free Rate (excluded)

Annualized Std Dev = Square Root (12) * Std Dev Monthly Returns

Annual Sharpe Ratio
This more traditional form of the Sharpe Ratio divides excess annual return by the standard deviation of annual returns to determine reward per unit of risk. Uses actual annual calendar returns:

Annual Sharpe Ratio = (Return - RFR) / Std Dev

Return = Compounded Annual Growth Rate

Risk Free Rate = Computed from the annual risk free rate as set in the TradingBlox.ini file converted to a compounded daily rate times 365.251.

Std Dev = Standard Deviation of Annual Returns

Robust Sharpe Ratio
The robust sharpe ratio is RAR% divided by the annualized standard deviation of the monthly returns. See page 189 of Way of the Turtle for more information.

Annual Sortino Ratio
Identical in form to the Annual Sharpe Ratio, but uses only downside deviation (negative data points) in its denominator, whereas the Sharpe Ratio uses overall standard deviation (which contains both upside and downside deviation). The Sortino Ratio is typically favored by those who believe that the performance of trend-following CTAs, and others, is unfairly penalized by the Sharpe Ratio's inclusion of upside volatility in its calculation. The higher the Sortino Ratio, the better:

Annual Sortino Ratio = (Return - RFR) / Semi-Deviation

Return = Compounded Annual Growth Rate

Risk Free Rate = Computed from the annual risk free rate as set in the TradingBlox.ini file converted to a compounded daily rate times 365.251.

Semi-Deviation = Std Dev of negative Annual returns

Note: In the calculation of standard deviation for the Sortino Ratio, N equals the number of negative data points, and not the number of data points in the entire population.

Monthly Sharpe Ratio
Identical in form to the Annual Sharpe Ratio, but based on monthly data points:

Monthly Sharpe Ratio = (Return - RFR) / Std Dev

Return = Average of the monthly returns.

Risk Free Rate = Converts the annual risk free rate as set in the TradingBlox.ini file to a compounded monthly return.

Std Dev = Standard Deviation of Monthly (calendar) Returns

Daily Sharpe Ratio
Identical in form to the Annual Sharpe Ratio, but based on daily data points:

Daily Sharpe Ratio = (Return - RFR) / Std Dev

Return = Average of the daily returns.

Risk Free Rate = Converts the annual risk free rate as set in the TradingBlox.ini file to a compounded daily return.

Std Dev = Standard Deviation of Monthly (calendar) Returns

Daily Geometric Sharpe Ratio
Identical in form to the Annual Sharpe Ratio, but based on daily data points:

Daily Sharpe Ratio = (Return - RFR) / Std Dev

Return = Compounded Daily Return.

Risk Free Rate = Converts the annual risk free rate as set in the TradingBlox.ini file to a compounded daily return.

Std Dev = Standard Deviation of Monthly (calendar) Returns

Monthly Sortino Ratio
Identical in form to the Annual Sortino Ratio, but based on monthly data points:

Monthly Sortino Ratio = (Return - RFR) / Semi-Deviation

Return = Average of the monthly returns.

Risk Free Rate = Converts the annual risk free rate as set in the TradingBlox.ini file to a compounded monthly return.

Semi-Deviation = Std Dev of negative Monthly data points

Calmar Ratio
Another gain-to- pain ratio, used to determine return relative to drawdown (downside) risk. The higher the Calmar ratio, the better: The Calmar Ratio is identical to the MAR Ratio with the exception of their denominators. MAR uses the Max DD over the life of the simulation, based on a daily, marked-to-the-market assessment. While the Calmar's denominator uses the Max DD over the life of the simulation, it is based on the worst month-end to month-end *Total Equity* data points. Note: Unlike some forms of the Calmar Ratio, which consider only the last 3 years of returns, Trading Blox uses the entire duration of the simulation.

Calmar Ratio = CAGR / Max Drawdown (monthly data points)

Drawdown Statistics

Longest Total Equity Drawdown
A one time event that reflects the longest duration drawdown in *Total Equity* over the life of the simulation. It is measured from previous equity peak to new equity peak, and expressed in months.

Maximum Monthly Total Equity DD%
Like Maximum Drawdown, this is a one-time event that reflects the largest retracement relative to a previous equity high in the entire simulation. It is based on a month-end to month-end, marked-to-the-market assessment. Expressed as a percentage, this statistic is calculated peak-to-valley using *Total*

Equity.

Average Max Drawdown (%)
The average of the 5 largest percent drawdowns based on total equity.

Average Max Drawdown Length
The average of the 5 largest total equity drawdowns by test days, converted to months.

Maximum Closed Equity Drawdown%
A one-time event that reflects the largest retracement relative to a previous equity high in the entire simulation. It is based on a daily, marked-to-the-market assessment. Expressed as a percentage, this statistic is calculated peak-to-valley using Closed Equity.

Average Closed Equity Drawdown%
The average of all Closed Equity drawdowns. If Monday is a Closed Equity peak, Tuesday shows a 2% retracement, and new peak is hit on Wednesday, then this one-day, 2% drawdown is stored, and averaged with all other drawdowns. This is significantly different from the way the other drawdown statistics are calculated.

Miscellaneous Performance Statistics

Round Turn per Million
The number of contracts traded per million currency on average. This number takes the average number of contracts traded divided by the average equity in millions.
Trading Blox computes the round turns per million as follows:

At the end of each month, the total round turns for the month are multiplied by 1,000,000 divided by the starting total equity for the month.

```
    turnsPerMillion = monthRoundTurns * (ONE_MILLION / monthStartingTotalEquity)
```

The total turns per million is then incremented by this monthly number.

```
            turnsPerMillionTotal = turnsPerMillionTotal + turnsPerMillion
```

At the end of the test, the average turns per million is computed as follows:

```
    averageTurnsPerMillion = ( turnsPerMillionTotal / totalTestingMonths ) * 12
```

End Account Balance
The ending balance after all trades are closed out at the end of the simulation (at this point Closed Equity and Total Equity are equal).

Highest Total Equity
Highest Total Equity achieved at any point during the simulation, using current market prices to value open positions.

Highest Closed Equity
Highest *Closed Equity* achieved at any point during the simulation. (Sometimes referred to as Closed Trade Equity.)

Total Commissions
Total commissions paid out for all positions.

Total Slippage
The total slippage (in currency) incurred, for all trades. See "Slippage Percent" for definition.

Total Forex Carry
The money earned or paid as a result of Forex Carry (see: Forex Carry Calculations for details)

Earned Interest
Total amount of interest (in currency), hypothetically earned on cash, or on T-Bill interest in a futures account.

Margin Interest
Total amount of interest (in currency), hypothetically paid for money borrowed on margin.

Win/Loss, Profit Factor, and Expectancy

Wins
Number of winning trades; also displayed as a percentage of Total Trades.

Losses
Number of losing trades; also displayed as a percentage of Total Trades.

Total Trades
Total number of trades.

Winning Months
Number of profitable months; also displayed as a percentage of total months in the simulation.

Losing Months
Number of unprofitable months; also displayed as a percentage of total months in the simulation.

Total Win Dollars
The total won on profitable trades, in system wide base currency.

Total Loss Dollars
The total lost on unprofitable trades, in system wide base currency.

Profit Factor
Total Win Dollars / Total Loss Dollars

Average Risk Percent
The average percent of equity risked per trade.

Average Win Percent
The average winning trade as a percentage of equity.

Average Loss Percent

The average losing trade as a percentage of equity.

Average Trade Percent
The average trade as a percentage of equity. This number will be positive for winning systems and negative for losing systems.

Percent Profit Factor
Total Wins in Percent / Total Losses in Percent - This is a more useful statistic than Profit Factor since it does not weight later trades
more heavily than earlier trades like Profit Factor does. To compute Total Wins in Percent, we add up the percentage won for each trade that is a winning trade as a percentage of the account equity at the time the trade was initiated. To compute Total Losses in Percent, we add up the percentage lost for each trade that is a losing trade as a percentage of the account equity at the time the trade was initiated.

Expectation
Sometimes known as Expectancy, this statistic shows how much one expects to gain for every amount bet or risked on a given trade. Numbers greater than 0.0 are winning systems, less than 0.0 are losing systems.

Average Gain of All Trades / Average Risk of All Trades, or,

```
( TotalWinPercent - TotalLossPercent ) / TotalRiskPercent
```

Margin to Equity Ratio

```
The summary margin to equity ratio is the MarginEquityTotal /
TotalTradingDays, where the MarginEquityTotal is a sum of each day's
margin to equity percent. The equity used for this is system total equity.
So this number is the average margin to equity percent over the course of
the test, using the system total equity as the equity basis.
```

Part X

Part 10 – Monte Carlo Simulations

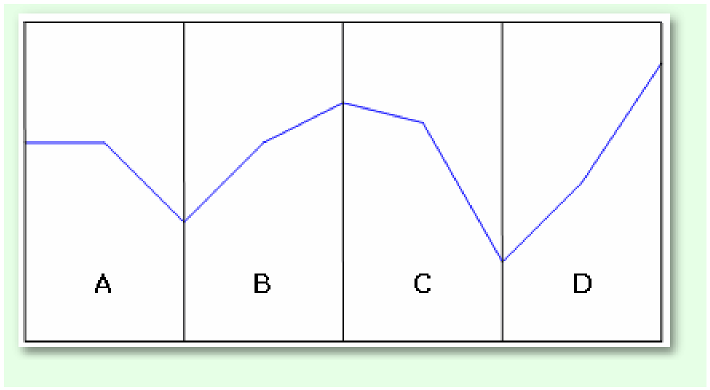
A Monte Carlo simulation is one way of determining the robustness of a system and of answering the question: "What if the past had been just slightly different"? Trading Blox 's Monte Carlo feature simulates new equity curves that are similar to the actual test equity curve but different in certain random ways. A typical Monte Carlo simulation might generate hundreds or thousands of new equity curves.

Trading Blox supports two different algorithms for the simulation of the Monte Carlo equity curves: reordering and sampling with replacement. In each case, the new equity curves are generated by taking portions of the actual test equity curve.

In real trading, bad days occur together with a frequency that is much higher than one might attribute to purely random chance. This is because at the end of large trends many markets seem to get carried along and then reverse at the same time. The period of a system's maximum drawdown is usually a statistically improbable series of down periods.
Trading Blox allows you to specify the number of days which make up each portion using the preference "Sample Grouping Days". This feature allows you to generate equity curves which maintain some of the autocorrelation found in actual equity curves. Without this feature, Monte Carlo simulations tend to underestimate the potential for large or lengthy drawdowns because the randomization process will result in equity curves that don't often show the lengthy periods of negative returns since they are statistically unlikely.

Reordering

The equity curve is randomly rearranged so that each portion appears in a different order.



Consider an equity curve like the above divided into four sections. If you simply reorder then you could have orders:

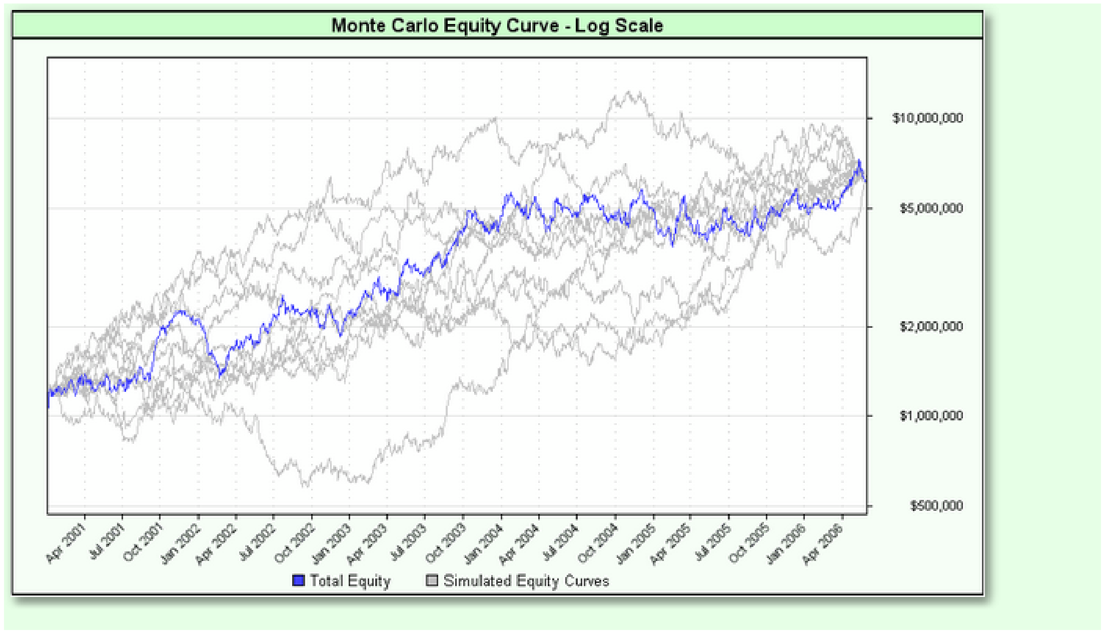
ABDC
ABCD
ACBD
ACBC
BCDA
BDCA
DCAB
DCBA
CDBA

etc.

Notice how an equity curve that contains AC adjacent to each other, i.e. ACDB, BACD, etc. will have a larger drawdown than the original equity curve because the reordered curve combines to down periods which were not adjacent in the original equity curve.

However since each of the individual sections of the equity curve represents a net percentage change in equity any reordering of the equity curves will result in exactly the same endpoint and therefore will result in the same CAGR%

The following shows the results of plotting the reordered equity curves (in gray) against the actual simulated equity curve (in blue):



Notice how all the curves each end at the same point.

Sampling with Replacement

The other method of simulating a new equity curve is called "Sampling with Replacement". This method generates a new equity curve by randomly selecting from the actual test equity curve with the additional provision that portions of the original equity curve can be used move than once.

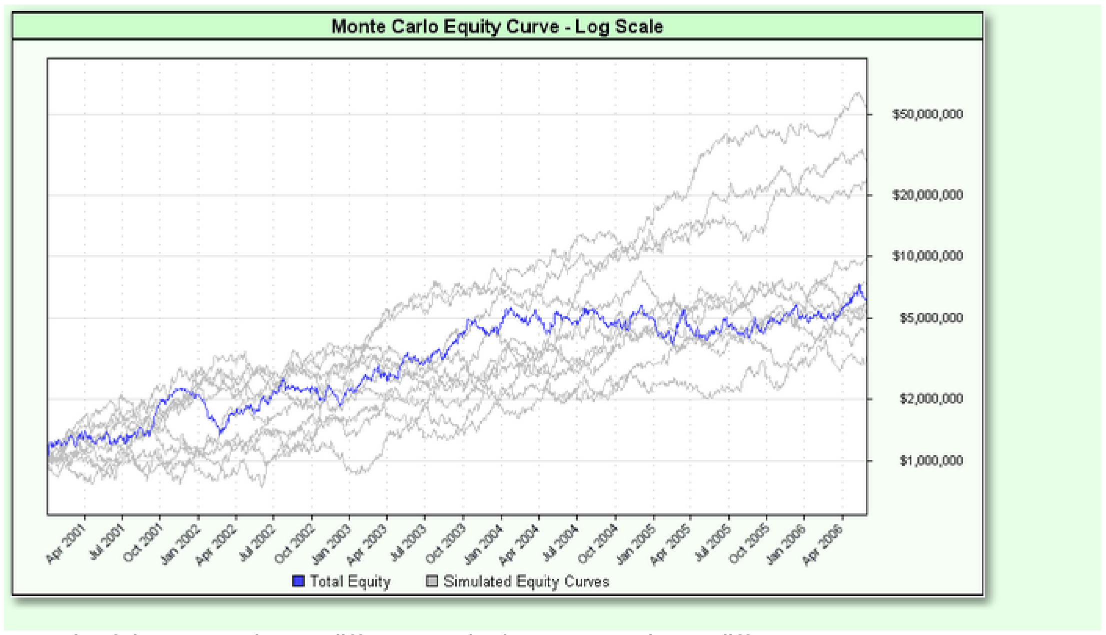
Thus you could have excellent curves like:

BDBD
BDDB
BBBB

or even

DDDD

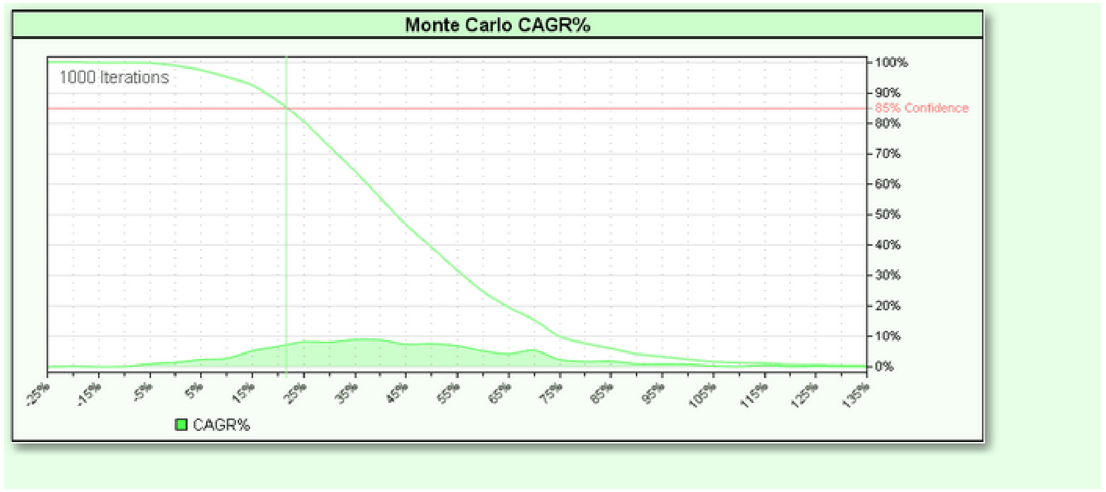
The following shows the results of plotting the equity curves from Sampling with Replacement (in gray) against the actual simulated equity curve (in blue):



Notice how each of the curves has a different endpoint representing a different CAGR%.

Many of the Monte Carlo graphs plot the distribution and cumulative distribution figures for various test measures, CAGR%, MAR Ratio, Maximum Drawdown etc.

The following graph shows the CAGR% graph:



Note the red line which is labeled 85% confidence. This line intersects with the cumulative curve on the graph at about the 21% level. This means that 85% of the simulation's tests showed an equity curve with a CAGR% greater than 21%.

Some Cautions

Beginning traders and system testers often look at particular test results and think that they mean more than they actually do.The future will not look like the past but will probably look something like the past. A simulation graph like the above helps one think in terms of the reality that the future will be one of many different possibilities, some better than the test results some worse than the test results.

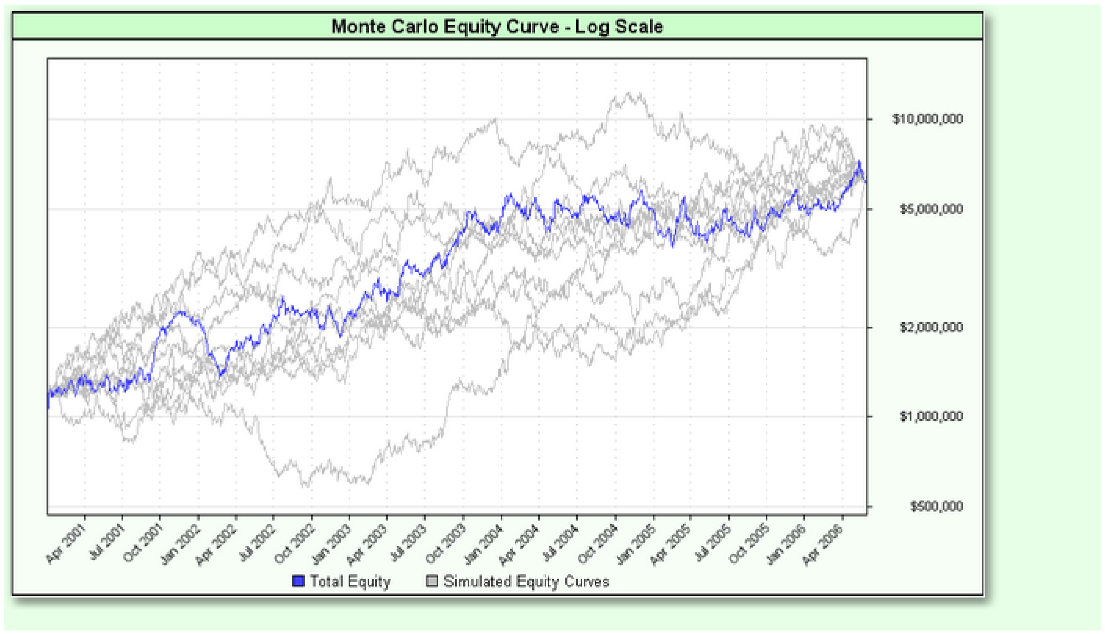
The confidence levels used by Trading Blox indicate only that a certain percentage of the results of the simulation are better for the given measure. The use of the term Confidence Level is common in the industry for this purpose. The use of the term does not imply that the test indicates that the future will result in a certain probability of a particular outcome. In other words, just because a test shows a 95% confidence level of a certain return or other measure does not mean that Trading Blox is predicting that there is a 95% chance of exceeding that measure in actual trading.

The confidence level can be set by the Monte Carlo preference settings. Some traders prefer to use 90% or 95% confidence levels.

Section 1 – Equity Curve Graph

The Monte Carlo Equity Graph plots a number of the Monte Carlo simulation alternate equity curves on the same graph with the tests actual equity plotted in blue.

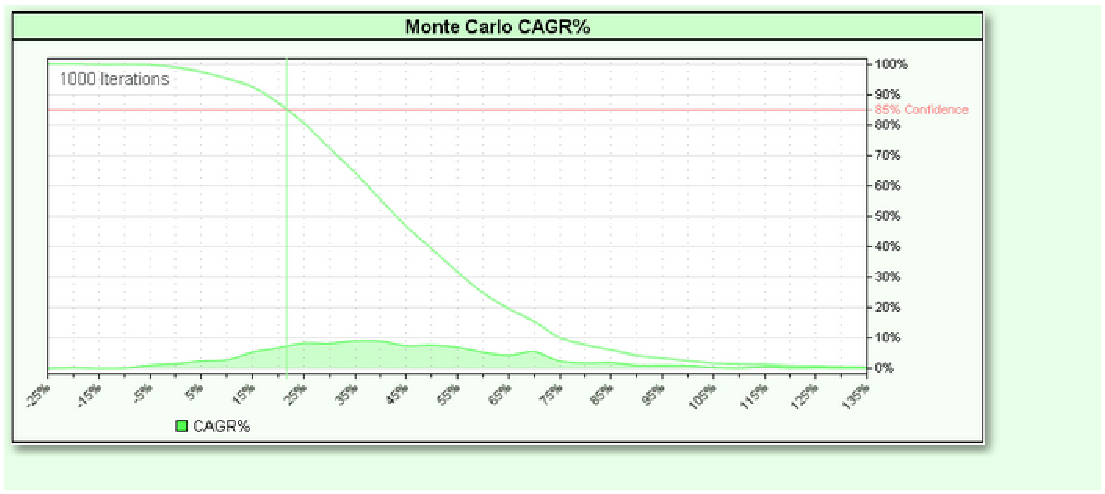
The following is a Monte Carlo Equity Graph.



The number of equity curves to plot can be set with the "Maximum Equity Curves to Graph" preference found here: Monte Carlo. The above graph shows a value of 10 for "Maximum Equity Curves to Graph".

Section 2 – Returns Graph

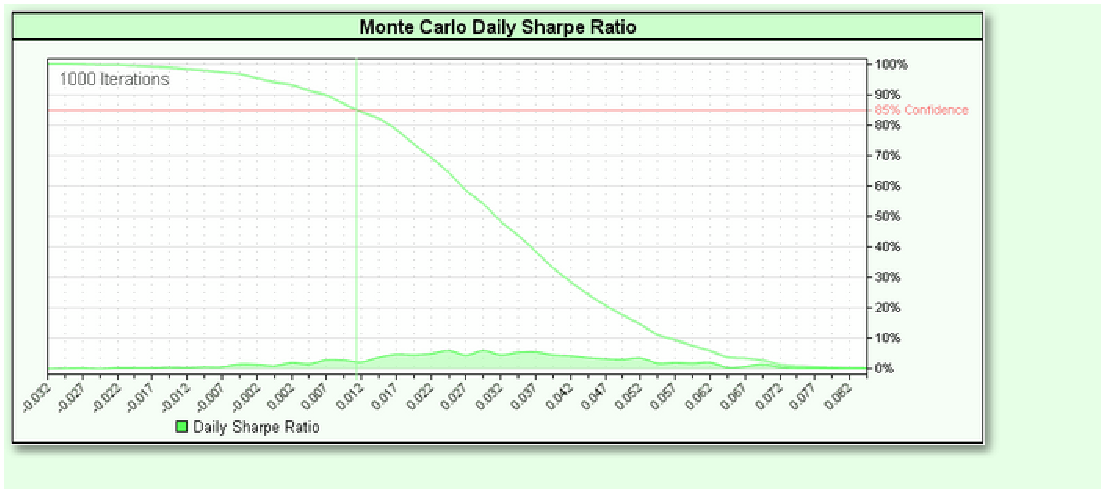
The Monte Carlo Returns Graph shows a histogram of the distribution and cumulative distribution for the returns expressed as CAGR%.



In the above example, the 85% confidence level corresponds to about 22% CAGR, meaning that 85% of the simulated equity curves showed a return of 22% or better. The distribution is centered around 35% but shows some simulated curves with a CAGR% as low as -25% and as high as 125%.

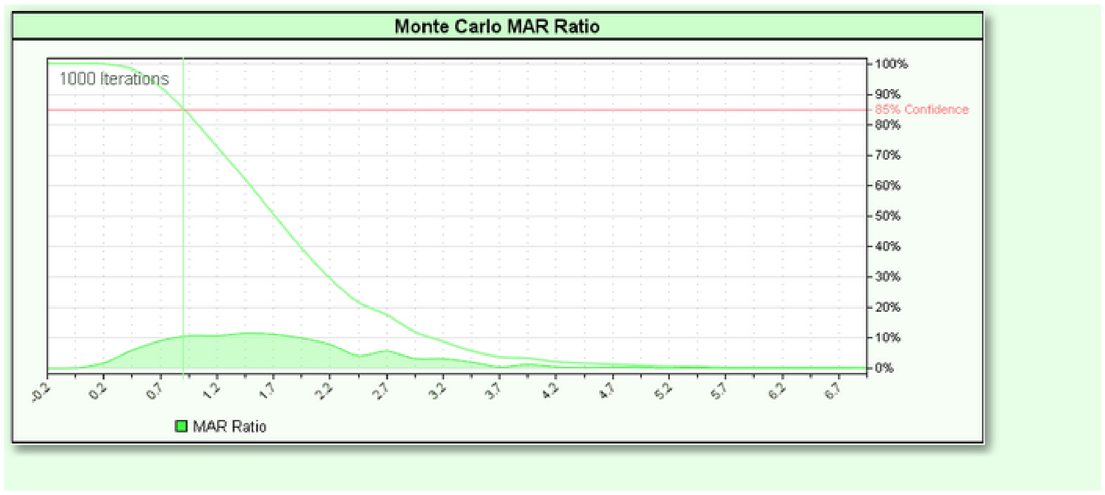
Section 3 – Sharpe Graph

The Monte Carlo Sharpe Graph shows a histogram of the distribution and cumulative distribution for the Daily Geometric Sharpe Ratio of the simulated equity curves.



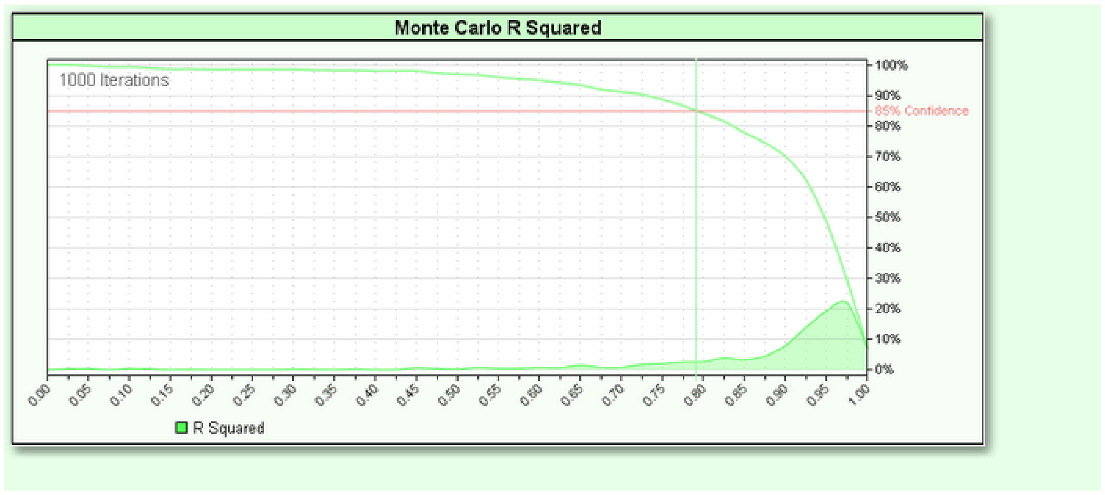
Section 4 – MAR Graph

The Monte Carlo MAR Graph shows a histogram of the distribution and cumulative distribution for the MAR ratio of the simulated equity curves.



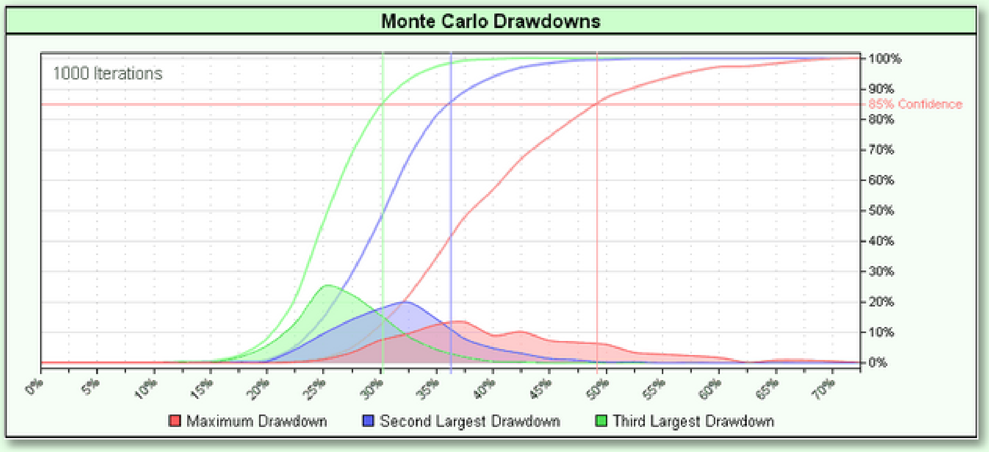
Section 5 – R Squared Graph

The Monte Carlo R Squared Graph shows a histogram of the distribution and cumulative distribution for the R Squared values for the simulated equity curves.



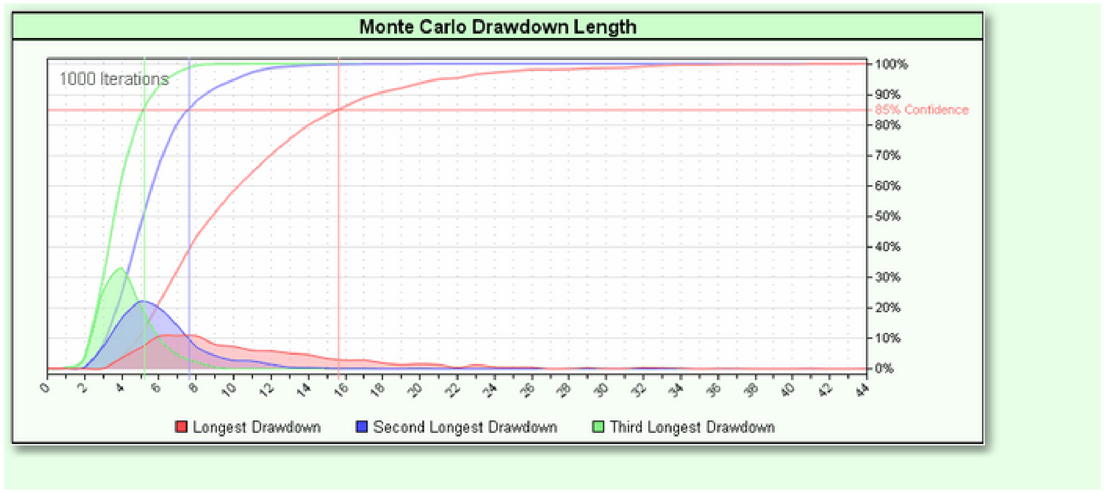
Section 6 – Drawdown Graph

The Monte Carlo Drawdown Graph shows a histogram of the distribution and cumulative distribution for the drawdowns for the simulated equity curves.



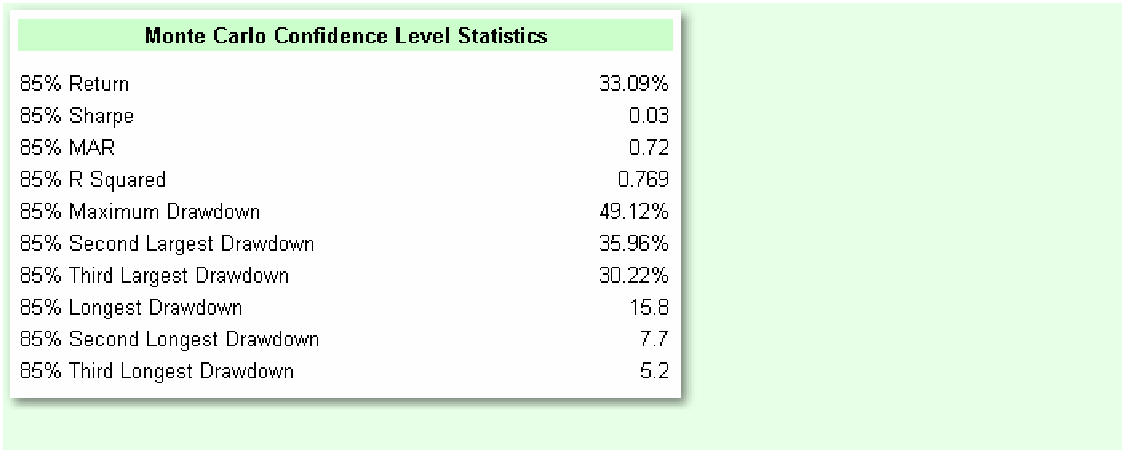
Section 7 – Drawdown Length Graph

The Monte Carlo Drawdown Length Graph shows a histogram of the distribution and cumulative distribution for the lengths of the drawdowns (in months) for the simulated equity curves.



Section 8 – Confidence Level Report

The Monte Carlo Confidence Level Report shows the values for various measures at the confidence level specified in Monte Carlo Preferences.



Monte Carlo Confidence Level Statistics	
85% Return	33.09%
85% Sharpe	0.03
85% MAR	0.72
85% R Squared	0.769
85% Maximum Drawdown	49.12%
85% Second Largest Drawdown	35.96%
85% Third Largest Drawdown	30.22%
85% Longest Drawdown	15.8
85% Second Longest Drawdown	7.7
85% Third Longest Drawdown	5.2

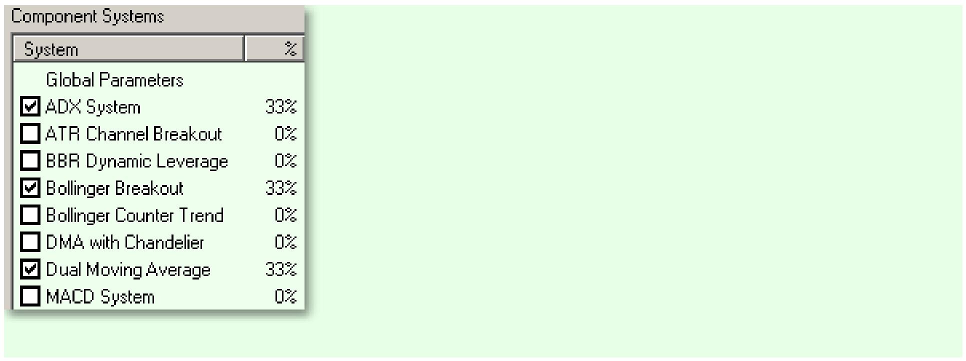
CAUTION: The confidence levels used here mean that 85% of the simulated equity curves had measures which were at the indicated level or better. This is not a predictive number. It does not mean that there is an 85% chance that a given measure will be exceeded in the future.

Part XI

Part 11 – Built-in Systems and Blox

Trading Blox comes with several complete trading systems (depending on which version of the software you own). Each of these systems can be tested separately or mixed together with any of the other trading systems. For example, it is possible to run a test with the Turtle System using 50% of the equity and a Triple Moving Average System using the other 50%. Trading Blox also includes the ability to test each system using a separate data set. You can trade the Turtle System using one portfolio and the Triple MA system using another portfolio. Each system has it's own parameters which can be varied and tested separately or in conjunction with those from the other systems.

To select a system or systems, check the box next to it:



Component Systems

System	%
Global Parameters	
✓ ADX System	33%
☐ ATR Channel Breakout	0%
☐ BBR Dynamic Leverage	0%
✓ Bollinger Breakout	33%
☐ Bollinger Counter Trend	0%
☐ DMA with Chandelier	0%
✓ Dual Moving Average	33%
☐ MACD System	0%

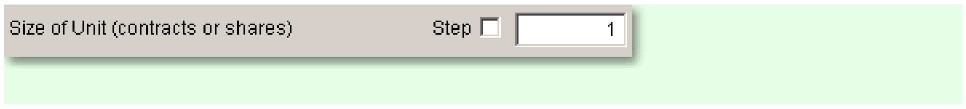
You can then set the allocation percentages in the Global Parameters.

Trading Blox Builder also has other built-in Blox that are used in our systems for money management, portfolio sizing, risk, etc. Remember that one of the most powerful features in Trading Blox Builder is the ability to use the same Blox in different systems! For information on how to customize our built-in systems or build your own, please see the Blox Builder manual. Each system topic includes a description of the system, the parameters, and the code.

Section 1 – Money Managers

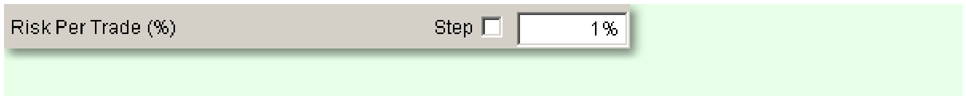
Basic Money Manager

True to its title, this Money Manager is very simple. The number of contracts or shares will be the number you input here.


Size of Unit (contracts or shares) Step ☐ 1

Fixed Fractional Money Manager

This Money Manager sets the unit size to a certain percent of your total trading equity. This Block is used by many of our built-in systems.


Risk Per Trade (%) Step ☐ 1%

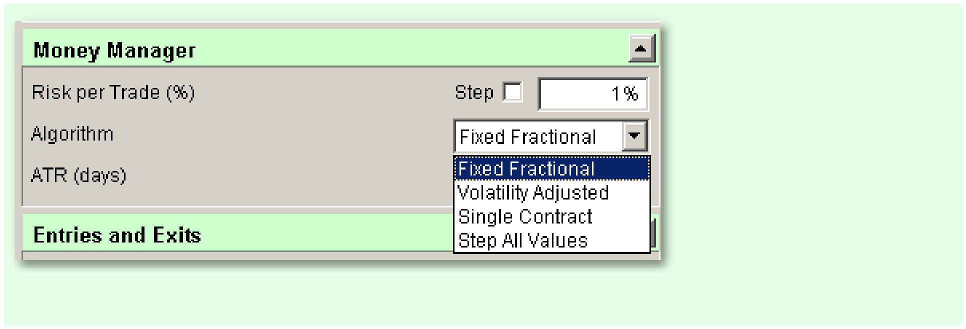
The position size is computed using the following formula:

$$\text{Position Size} = \frac{(\text{Percent Risk per Trade x Trading Equity})}{(\text{Entry Risk Points x Dollars per Point})}$$

If you have no stops, there will be no risk. If there is no risk, there can be no trades with this money manager!

Multi Money Manager

This money manager allows you to choose from three choices of money management:



Risk per Trade (%) Step ☐ 1%
Algorithm Fixed Fractional
ATR (days)

1. Single Contract - The number of contracts or shares will be one.
2. Fixed Fractional - Described above. Remember that you must have stops in your system to get trades with this method of money management.
3. Volatility Adjusted - This method can be used without stops, and uses the Average True Range to size positions. The greater the volatility, the smaller the position size.

Section 2 – Risk Managers

Trading Blox provides a variety of risk limiting controls that can be added to a system as a source of ideas on which to build custom blox that are more tailored to the trader's goals.

Each of the risk managing blox provide their own approach to filtering orders, or by reducing the distance between the protective price and the close of each date,

Risk Managing Blox:	Description:
Correlation Risk Manager	This Risk Manager is used in the ADX, MACD, and Turtle built-in systems to limit the maximum number of correlated units.
Correlation Risk Manager Check Fills	This Risk Manager is used in the ADX, MACD, and Turtle built-in systems to limit the maximum number of correlated units.
Group Risk Manager	Entry orders are rejected based on violations of threshold values for each of the group risk sections enabled.
Total Risk Limiter	This risk manager allows you to select a maximum amount of risk you want at any given time, and then a method of reducing positions if your risk is too great. If you choose "Reduce Positions" this block will sell a portion of your units to reduce to below the maximum. If you select "Move Stops" the system will set new stops based on the maximum risk and place orders accordingly.
Unit Limiter	This block rejects entry orders for new units if the number of current positions in the same direction in closely or loosely correlated markets is greater than or equal to the user set maximum.

Order Fill Predictability:

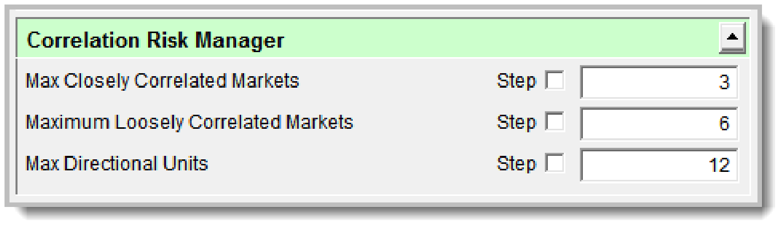
Success in limiting new entry orders to reduce risk, limit position or unit correlations and the number of open position is highly dependent upon the order entry order's selected execution method. Success also requires the orders be controlled ahead of them being reported and which orders will be filled in the next trading session. Knowing an order is going to be filled ahead of it being released is limited to market orders that only need a trading session to execute. Orders that add price and or session timing conditions cannot be predicted with any reasonable level of certainty.

Entry Order Categories:	**Execution Conditions:**
Market: On-Open On-Close	These orders are easy to apply order restrictions because their ability to be filled by the market is certain in most situations. All that has to happen for these orders to be filled is for the market to price another trading session and the order will be filled.
Conditional: On-Stop On-Limit On-Stop Open At-Limit Open On-Stop Close At-Limit Close	All of these order execution types require trading session conditions that are not possible to reliably predict their ability for being executed. Without predictability rigid order counts will create below expectation limiting results. Controlling these types of orders so they allow results that are closers to the parameter values used to control orders can be closer when there is a percentage overage allowance, but the results will the control will allow values above and below the parameter values applied.

2.1 Correlation Risk Manager

This Risk Manager is used in the ADX, MACD, and Turtle built-in systems to limit the maximum number of correlated units.

Correlation Risk Manager Parameters:



Correlation Risk Manager

Max Closely Correlated Markets	Step ☐	3
Maximum Loosely Correlated Markets	Step ☐	6
Max Directional Units	Step ☐	12

Parameter:	Description:
Max Closely Correlated Markets	Sets the max number of units the system can have on in closely correlated markets.
Maximum Loosely Correlated Markets	Sets the max number of units the system can have on in loosely correlated markets
Max Directional Units	Sets the max number of units the system can have on in each direction, Long or Short.

Correlation Risk Manager Control Logic:

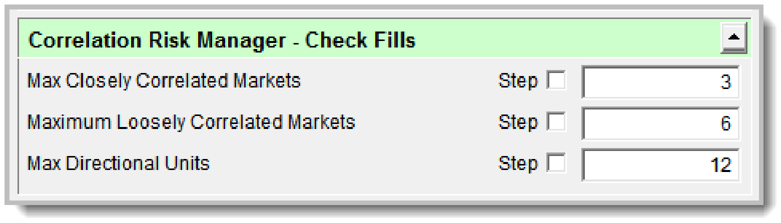
```
CAN ADD UNIT:
```

```
'  ==============================================================
'  Correlation Risk Mgr
'  CAN ADD UNIT SCRIPT - START
'  ==============================================================
'  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
'  Check correlations, and total directional positions
'  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
If order.position = LONG THEN
   '  LONG ORDER Correlation Filter
   If   instrument.closelyCorrelatedLongUnits >= maxCloselyCorrelatedMarkets O
      instrument.looselyCorrelatedLongUnits >= maxLooselyCorrelatedMarkets THE
      order.Reject( "Too many correlated long units" )
   ENDIF
   '  -----------------------------------------------------------
   '  LONG Position Count Filter
   If system.totalLongUnits >= maxDirectionalUnits THEN
      order.Reject( "Too many long directional units" )
   ENDIF
ELSE' o.position = LONG
'  --------------------------------------------------------------
   '  SHORT ORDER Correlation Filter
   If   instrument.closelyCorrelatedShortUnits >= maxCloselyCorrelatedMarkets
      instrument.looselyCorrelatedShortUnits >= maxLooselyCorrelatedMarkets TH
      order.Reject( "Too many correlated short units" )
   ENDIF
   '  -----------------------------------------------------------
   '  SHORT Position Count Filter
   If system.totalShortUnits >= maxDirectionalUnits THEN
      order.Reject( "Too many short directional units" )
   ENDIF
ENDIF' o.position = SHORT
'  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
'  ==============================================================
'  CAN ADD UNIT - END
'  Correlation Risk Mgr
'  ==============================================================
```

2.2 Correlation Risk Manager - Check Fills

This version or the Correlation Risk Manager has the same functionality of the Correlation Risk Manager plus the addition of an ability check the process after the order has been filled.

Correlation Risk Manager - Check Fills Parameters:


Correlation Risk Manager - Check Fills

Max Closely Correlated Markets	Step ☐	3
Maximum Loosely Correlated Markets	Step ☐	6
Max Directional Units	Step ☐	12

Parameter:	Description:
Max Closely Correlated Markets	Sets the max number of units the system can have on in closely correlated markets.
Maximum Loosely Correlated Markets	Sets the max number of units the system can have on in loosely correlated markets
Max Directional Units	Sets the max number of units the system can have on in each direction, Long or Short.

Correlation Risk Manager - Check Fills Control Logic:

```
CAN ADD UNIT:
    '  ==============================================================
    '  Correlation Risk Mgr wFills
    '  CAN ADD UNIT SCRIPT - START
    '  ==============================================================
    '  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    ' Check LONG correlations, and Total LONG Position Count
    '  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    If order.position = LONG THEN
       '  LONG ORDER Correlation Filter
       If instrument.closelyCorrelatedLongUnits >= maxCloselyCorrelatedMarkets OR
          instrument.looselyCorrelatedLongUnits >= maxLooselyCorrelatedMarkets THE
          '  Remove Order and Send Rejection Msg to Filter Log
          order.Reject( "Too many correlated long units" )
       ENDIF
       '  -----------------------------------------------------------
       '  LONG Position Count Filter
       If system.totalLongUnits >= maxDirectionalUnits THEN
          '  Remove Order and Send Rejection Msg to Filter Log
          order.Reject( "Too many long directional units" )
       ENDIF
    ELSE' o.position = LONG
    '  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    ' Check SHORT correlations, and Total SHORT Position Count
    '  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
       '  SHORT ORDER Correlation Filter
       If instrument.closelyCorrelatedShortUnits >= maxCloselyCorrelatedMarkets OR
          instrument.looselyCorrelatedShortUnits >= maxLooselyCorrelatedMarkets TH
          '  Remove Order and Send Rejection Msg to Filter Log
          order.Reject( "Too many correlated short units" )
       ENDIF
       '  -----------------------------------------------------------
       '  SHORT Position Count Filter
       If system.totalShortUnits >= maxDirectionalUnits THEN
          '  Remove Order and Send Rejection Msg to Filter Log
          order.Reject( "Too many short directional units" )
       ENDIF
    ENDIF '  .position = SHORT
    '  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    '  ==============================================================
    '  CAN ADD UNIT - END
    '  Correlation Risk Mgr wFills
    '  =============================================================
```

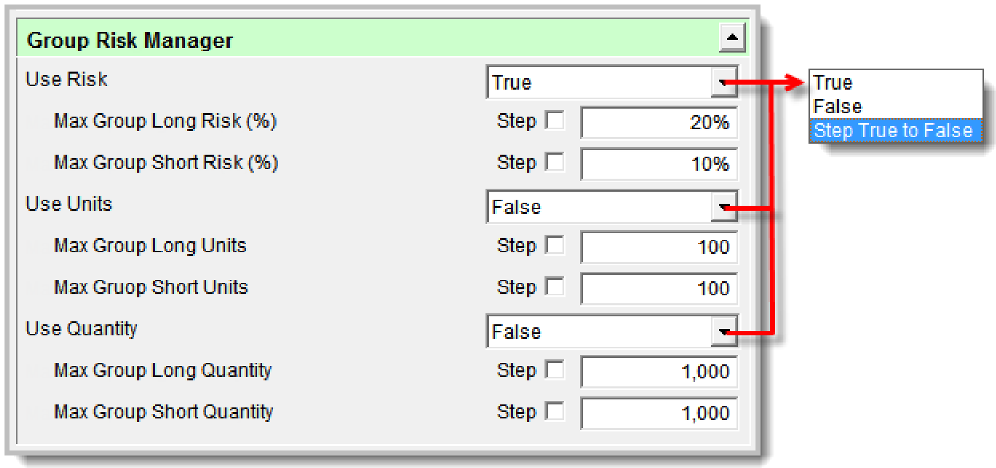
CAN FILL ORDER:

```
'  ==============================================================
'  Correlation Risk Mgr wFills
'  CAN FILL ORDER SCRIPT - START
'  ==============================================================
'  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
' Check correlations, and total directional positions
'  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
'  Only filter Order Correlationons on Entry Fill.
'  There are no count or Correlcation filter for exits
If order.isEntry THEN
   '  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
   ' Check LONG Order correlations, and Total LONG Position Count
   '  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
   If order.position = LONG THEN
      '  --------------------------------------------------------
      '  LONG Order-Filled Correlation Filter
      '  --------------------------------------------------------
      If instrument.closelyCorrelatedLongUnits >= maxCloselyCorrelatedMarkets
         instrument.looselyCorrelatedLongUnits >= maxLooselyCorrelatedMarkets
         order.Reject( "Too many correlated long units" )
      ENDIF
      '  --------------------------------------------------------
      ' Check Total SHORT Position Count
      '  --------------------------------------------------------
      If system.totalLongUnits >= maxDirectionalUnits THEN
         order.Reject( "Too many long directional units" )
      ENDIF
   ELSE
      '  --------------------------------------------------------
      ' Check SHORT Order correlations
      '  --------------------------------------------------------
      If instrument.closelyCorrelatedShortUnits >= maxCloselyCorrelatedMarkets
         instrument.looselyCorrelatedShortUnits >= maxLooselyCorrelatedMarkets
         order.Reject( "Too many correlated short units" )
      ENDIF
      '  --------------------------------------------------------
      ' Check Total SHORT Position Count
      '  --------------------------------------------------------
      If system.totalShortUnits >= maxDirectionalUnits THEN
         order.Reject( "Too many short directional units" )
      ENDIF
   ENDIF
ENDIF
'  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
'  ==============================================================
'  CAN FILL ORDER - END
'  Correlation Risk Mgr wFills
'  ==============================================================
```

2.3 Group Risk Manager

Entry orders are rejected based on violations of threshold values for each of the group risk sections enabled.

Group Risk Manager Parameters:



Group Risk Manager

Use Risk		True
Max Group Long Risk (%)	Step ☐	20%
Max Group Short Risk (%)	Step ☐	10%
Use Units		False
Max Group Long Units	Step ☐	100
Max Gruop Short Units	Step ☐	100
Use Quantity		False
Max Group Long Quantity	Step ☐	1,000
Max Group Short Quantity	Step ☐	1,000

Parameters:	Descriptions:
Use Risk	True enables Max Group Risk parameters.
Max Group Long Risk %	Max risk for Long instruments in the group.
Max Group Short Risk %	Max risk for Short instruments in the group.
Use Units	True enables Max number of Long and Short Group Units.
Max Group Long Units	Max number of Long units in the group.
Max Group Short Units	Max number of Short units in the group.
Use Quantity	True enables Max number of Long and Short Group positions.
Max Group Long Quantity	Max quantity of Long positions.
Max Group Short Quantity	Max quantity of Short positions.

Group Risk Limiter Control Logic:

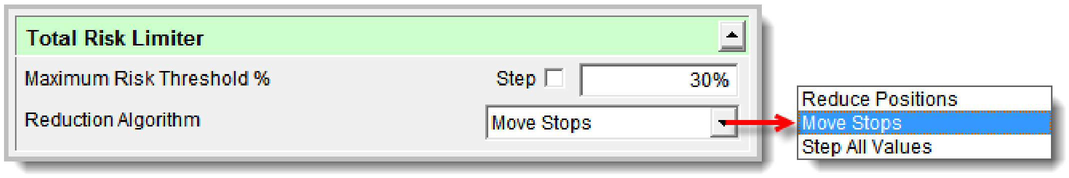
```
CAN ADD UNIT:
```

```
'  ==============================================================
'  Group Risk Manager
'  CAN ADD UNIT SCRIPT - START
'  ==============================================================
'  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
'  LONG Order Filtering Logic
'  --------------------------------------------------------------
' Check the long positions in this group to see if we can add a new unit
If order.position = LONG AND
   ( ( useRisk AND instrument.groupLongRisk >= maxLongRisk * system.tradingEqu
     ( useUnits AND instrument.groupLongUnits >= maxLongUnits ) OR
     ( useQuantity AND instrument.groupLongQuantity >= maxLongQuantity ) ) THE
   order.Reject( "Long group risk greater than maximum" )
ENDIF
'  --------------------------------------------------------------
'  SHORT Order Filtering Logic
'  --------------------------------------------------------------
' Check the short positions in this group to see if we can add a new unit
If order.position = SHORT AND
   ( ( useRisk AND instrument.groupShortRisk >= maxShortRisk * system.tradingE
     ( useUnits AND instrument.groupShortUnits >= maxShortUnits ) OR
     ( useQuantity AND instrument.groupShortQuantity >= maxShortQuantity ) ) T
   order.Reject( "Short group risk greater than maximum" )
ENDIF
'  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
'  ==============================================================
'  CAN ADD UNIT - END
'  Group Risk Manager
'  ==============================================================
```

2.4 Total Risk Limiter

This risk manager allows you to select a maximum amount of risk you want at any given time, and then a method of reducing positions if your risk is too great. If you choose "Reduce Positions" this block will sell a portion of your units to reduce to below the maximum. If you select "Move Stops" the system will set new stops based on the maximum risk and place orders accordingly.

Total Risk Limiter Parameters:



Total Risk Limiter

Maximum Risk Threshold % — Step ☐ — 30%

Reduction Algorithm — Move Stops

Parameters:	Descriptions:
Maximum Risk Threshold %	Enter the maximum amount of risk as a percentage that this blox should limit.
Reduction Algorithm	Blox provide two methods for reducing the percentage of risk being publised by the open risk to current trading equity amount. **Option 1: Reduce Position:** This option will reduce the position's current quantity using the percentage value entered into the "**Maximum Risk Threashold %.**" **Option 2: Move Stops:** Distance between the current close and the position's protective risk price determines how many position points are at risk. Risk points are converted to a monetary value so that it can be expanded by the number of quantity contained in the position. Once the amount of risk in monetary terms is understood it is divided by the System's Trading Equity series element for the each trade day so that a current open risk rate can be determined.

Note:

The following expression determines the amount by which risk is reduced when the threshold is breached:

```
        Reduction Amount = (Portfolio Risk - Max Risk Threshold) / Portfolio
Risk
```

Example:

When current risk is 40% and the maximum threshold is 30%, then risk will be reduced by: (40% - 30%) / 40% = 25%.

Reducing all current position sizes by 25%, or reducing each stop offset distance (current price to the current exit) by 25% brings the position risk back to 30%.

Total Risk Manager Control Logic:

INITIALIZE RISK MANAGER:

```
' ==============================================================
' Total Risk Limiter
' INITIALIZE RISK MANAGER SCRIPT - START
' ==============================================================
' Clear Total Risk Accumulator
totalRisk = 0
' ==============================================================
' INITIALIZE RISK MANAGER SCRIPT - END
' Total Risk Limiter
' ==============================================================
```

COMPUTE INSTRUMENT RISK

```
' ==============================================================
' Total Risk Limiter
' COMPUTE INSTRUMENT RISK SCRIPT - START
' ==============================================================
' Add the instrument risk to the total risk.
totalRisk = totalRisk + instrument.currentPositionRisk
' ==============================================================
' COMPUTE INSTRUMENT RISK SCRIPT - END
' Total Risk Limiter
' ==============================================================
```

COMPUTE RISK ADJUSTMENTS:

```
' ==============================================================
' Total Risk Limiter
' COMPUTE RISK ADJUSTMENTS SCRIPT - START
' ==============================================================
' ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
If system.tradingEquity > 0 THEN
   ' Compute the current risk.
   riskPercent = totalRisk / system.tradingEquity

   ' If the risk is above our threshold...
   If riskPercent > maximumRiskThreshold THEN
      reductionPercent = (riskPercent - maximumRiskThreshold) _
                              / riskPercent
   ELSE
      reductionPercent = 0.0
   ENDIF
ELSE
   reductionPercent = 0.0
ENDIF
' ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
' ==============================================================
' COMPUTE RISK ADJUSTMENTS SCRIPT - END
' Total Risk Limiter
' ==============================================================
```

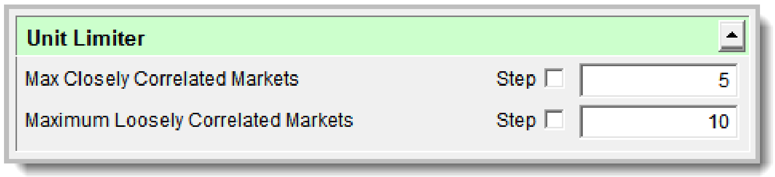
ADJUST INSTRUMENT RISK:

```
'  ==============================================================
'  Total Risk Limiter
'  ADJUST INSTRUMENT RISK SCRIPT - START
'  ==============================================================
'  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
'  REDUCE POSITION QUANTITY
'  --------------------------------------------------------------
'  If we need to reduce risk
If reductionPercent > 0.0 THEN
   If reductionAlgorithm = REDUCE_POSITIONS THEN
      '  Reduce the position by this amount.
      broker.AdjustPositionOnOpen( 1.0 - reductionPercent )
   ENDIF
   '  -----------------------------------------------------------
   '  LONG POSITION - MOVE STOP CLOSER
   '  -----------------------------------------------------------
   If reductionAlgorithm = MOVE_STOPS THEN
      If instrument.position = LONG THEN
         '  Adjust the stops for each unit.
         For index = 1 TO instrument.currentPositionUnits
            '  Determine the current risk.
            risk = instrument.close - instrument.unitExitStop[ index ]
            '  Determine the stop that corresponds with the reduced risk.
            newStop = instrument.close - ((1 - reductionPercent) * risk)
            '  Set the new stop.
            instrument.SetExitStop( index, newStop )
            broker.ExitUnitOnStop( index, newStop )
         Next
      ENDIF ' Long
   '  -----------------------------------------------------------
   '  SHORT POSITION - MOVE STOP CLOSER
   '  -----------------------------------------------------------
      If instrument.position = SHORT THEN
         '  Adjust the stops for each unit.
         For index = 1 TO instrument.currentPositionUnits
            '  Determine the current risk.
            risk = instrument.unitExitStop[index] - instrument.close
            '  Determine the stop that corresponds with the reduced risk.
            newStop = instrument.close + ((1 - reductionPercent) * risk)
            ' Set the new stop.
            instrument.SetExitStop( index, newStop )
            broker.ExitUnitOnStop( index, newStop )
         Next
      ENDIF '  Short
   ENDIF '  Algorithm Move Stops
ENDIF '  There is a reduction required
'  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
'  ==============================================================
'  ADJUST INSTRUMENT RISK SCRIPT - END
'  Total Risk Limiter
'  ==============================================================
```

2.5 Unit Limiter

This block rejects entry orders for new units if the number of current positions in the same direction in closely or loosely correlated markets is greater than or equal to the user set maximum.

Unit Limiter Parameters:



Unit Limiter

Max Closely Correlated Markets	Step ☐	5
Maximum Loosely Correlated Markets	Step ☐	10

Parameter:	Description:
Max Closely Correlated Markets	Integer value that controls the maximum number of closely correlated markets allowed at any time during the test period.
Maximum Loosely Correlated Markets	Integer value that controls the maximum number of loosely correlated markets allowed at any time during the test period.

Unit Limiter Control Logic:

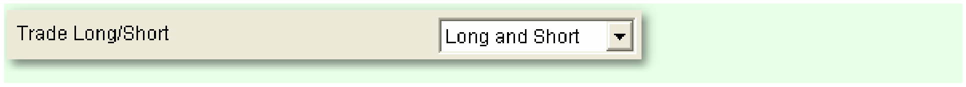
```
CAN ADD UNIT SCRIPT:
```

```
'  ==============================================================
'  Unit Limiter
'  CAN ADD UNIT SCRIPT - START
'  ==============================================================
'  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
'  If this is a proposed new Long unit...
'  --------------------------------------------------------------
If order.position = LONG THEN

   '  Compute the total correlated units.
   closelyCorrelatedUnits = instrument.currentPositionUnits _
                              + instrument.closelyCorrelatedLongUnits
   looselyCorrelatedUnits = closelyCorrelatedUnits _
                              + instrument.looselyCorrelatedLongUnits

   ' If the total correlated units are below their respective limits...
   If   closelyCorrelatedUnits >= maxCloselyCorrelatedMarkets AND
      looselyCorrelatedUnits >= maxLooselyCorrelatedMarkets THEN
      '  Remove Order and Send Rejection Msg to Filter Log
      order.Reject( "Correlated long units greater than maximum" )
   ENDIF
ELSE  ' o.position = LONG

'  --------------------------------------------------------------
'  If this is a proposed new Short unit...
'  --------------------------------------------------------------
   '  Compute the total correlated units.
   closelyCorrelatedUnits = instrument.currentPositionUnits _
                              + instrument.closelyCorrelatedShortUnits
   looselyCorrelatedUnits = closelyCorrelatedUnits _
                              + instrument.looselyCorrelatedShortUnits

   '  If the total correlated units are below their respective limits...
   If   closelyCorrelatedUnits >= maxCloselyCorrelatedMarkets AND
      looselyCorrelatedUnits >= maxLooselyCorrelatedMarkets THEN

      '  Remove Order and Send Rejection Msg to Filter Log
      order.Reject( "Correlated short units greater than maximum" )
   ENDIF
ENDIF '  ' o.position = LONG
'  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
'  ==============================================================
'  CAN ADD UNIT SCRIPT - END
'  Unit Limiter
'  ==============================================================
```

Section 3 – Portfolio Managers

Trade Direction Portfolio Manager

The *Trade Long/Short* parameter controls which types of trades the system will take. The possible values are: *Long and Short*, *Long Only*, *Short Only*, and *Step All Values*. Choosing *Step All Values* will result in a three parameter run test with each of the other three values. Most of our built-in systems use this Portfolio Manager.



Trade Long/Short: Long and Short

Strength Filter

The *Strength Filter* filters a pool of candidates which the systems can trade from the current portfolio. While this parameter is normally used for stocks, it can be applied to futures or forex.

The *Strength* used in the *Strength Filter* is simply a volatility-adjusted measure of the price increase (or decrease) of an instrument over a finite time period. The parameter Strength Days defines this measurement period (in days).

Filter Threshold tells Trading Blox how many instruments to filter out of the system's portfolio and lay aside as candidates for possible entries by the system. Instruments are ranked based on the formula:

$$\text{Strength} = \frac{(\text{price today - price } \textit{Strength Days} \text{ ago})}{(\text{the value of ATR } \textit{Strength Days} \text{ ago})}$$

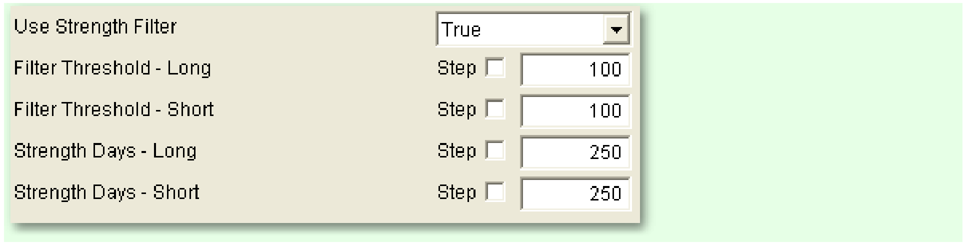
Strength Filter has two sections: *Long* which selects for strength, and *Short* which selects for weakness.

The *Long* section filters instruments (individual futures, stocks, or forex) based on price strength. It culls the highest strength instruments over the time period specified by *Strength Days*, and sets aside the number of candidates specified by the *Filter Threshold* value.

The *Short* section filters for low price strength. It culls the lowest strength instruments over the time period specified by Strength Days, and sets aside the number of candidates specified by the *Filter Threshold* value.

The *Strength Filter* simply provides a pool of candidates which the systems can trade. The particular instruments ultimately traded out of the pool provided by the strength filter are a function of the entry rules for that system.

If Use Strength Filter is set to *True* or *Step True to False* then four additional parameters will be shown:



Use Strength Filter		True
Filter Threshold - Long	Step ☐	100
Filter Threshold - Short	Step ☐	100
Strength Days - Long	Step ☐	250
Strength Days - Short	Step ☐	250

Filter Threshold - Long
Applicable only if *Use Strength Filter* is True, this parameter defines the number of instruments to be filtered out as candidates for long trades.

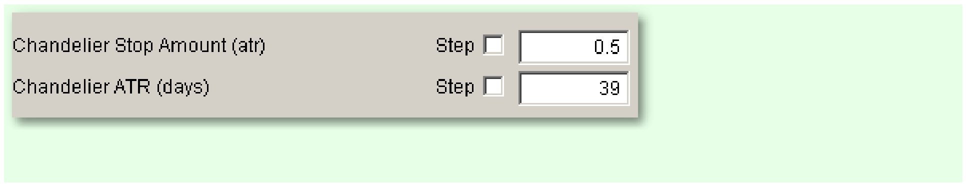
Filter Threshold - Short
Applicable only if *Use Strength Filter* is True, this parameter defines the number of instruments to be filtered out as candidates for short trades.

Strength Days - Long
Applicable only if *Use Strength Filter* is True, this parameter defines the length of time in days over which performance is examined to determine strength for long trades.

Strength Days - Short
Applicable only if *Use Strength Filter* is True, this parameter defines the length of time in days over which instrument performance is examined to determine weakness for short trades.

Section 4 – Chandelier Exit

The Chandelier Exit system is only an Exit Block, so it does not place entry orders. It does place stop exit orders based on ATR if a new high or low is hit during an open trade.

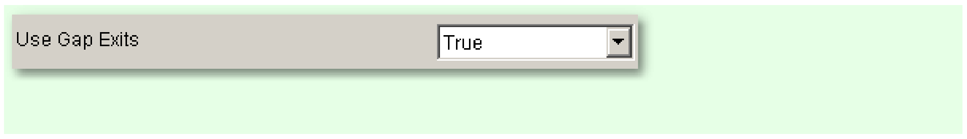


Chandelier Stop Amount (atr)	Step ☐	0.5
Chandelier ATR (days)	Step ☐	39

You can input (or step) values for the number of days to calculate ATR and the ATR stops it uses to place stop orders.

Section 5 – Gap Against Exit

This Exit Block exits all open positions if today's high is below yesterday's close (for a long trade) or if today's low is below yesterday's close (for a short trade). It's designed to protect against loss. It has only one parameter that functions like an on/off switch.



Use Gap Exits True

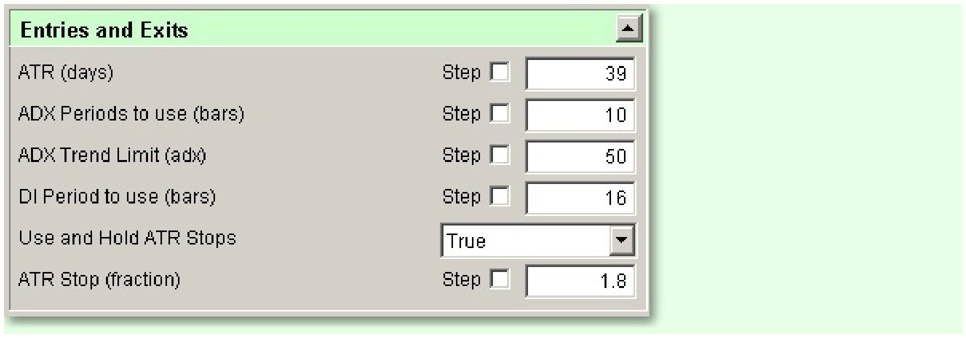
Section 6 – ADX System

The Average Directional Index (ADX) was developed by J. Welles Wilder to help traders determine the *strength* of the trend. Two components that help you determine the *direction* of the trend are the Positive Directional Indicator (+DI) and Negative Directional Indicator (-DI).

The built-in ADX System uses two factors to signal and entry:
- The strength of the trend (ADX) needs to be greater than a certain threshold
- The cross over of the +DI and -DI

It uses a fraction of the ATR plus/minus the close to set the stop price.

The ADX System uses the following parameters:


Entries and Exits

ATR (days)	Step ☐	39
ADX Periods to use (bars)	Step ☐	10
ADX Trend Limit (adx)	Step ☐	50
DI Period to use (bars)	Step ☐	16
Use and Hold ATR Stops		True
ATR Stop (fraction)	Step ☐	1.8

Average True Range (days)
Sets the number of days to use when calculating the Average True Range

ATR Stop (fraction)
Sets the fraction (.5, 1.5, 3) of ATR to use for the stop width

ADX Periods to use (bars)
Sets the number of bars to use in calculating the ADX

ADX Trend Limit (adx)
Sets the minimum threshold for entering a trade

DI Period to use (bars)
Sets the number of bars to use in calculating the +DI and -DI

Entry Script

```
IF  adxIndicator > adxTrendLimit AND
    positiveDirectionalIndicator > negativeDirectionalIndicator AND
    instrument.position <> LONG THEN

    IF useATRStops THEN
        broker.EnterLongOnOpen( instrument.close - averageTrueRange * atrStop )
    ELSE
        broker.EnterLongOnOpen
    ENDIF

ENDIF
```

```
IF  adxIndicator > adxTrendLimit AND
    positiveDirectionalIndicator < negativeDirectionalIndicator AND
    instrument.position <> SHORT THEN

    IF useATRStops THEN
        broker.EnterShortOnOpen( instrument.close + averageTrueRange * atrStop )
    ELSE
        broker.EnterShortOnOpen
    ENDIF

ENDIF
```

Adjust Stops

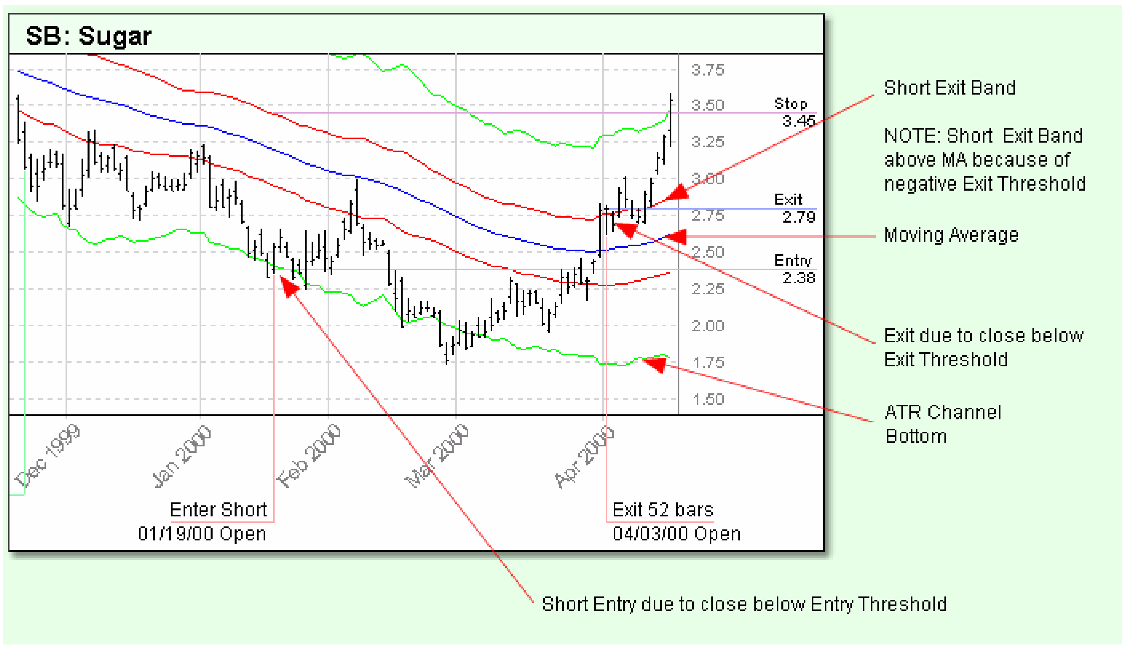
```
' ---------------------------------------------
' Enter stop if "Use ATR Stops" is true
' ---------------------------------------------

IF useATRStops THEN

    broker.ExitAllUnitsOnStop( instrument.unitExitStop )

ENDIF
```

Section 7 – ATR Channel Breakout System

This system is a variation on the Bollinger Breakout System which uses Average True Range instead of standard deviation as a measure of the volatility which defines the width of the channels or bands.

A variation of the *ATR Channel Breakout System* was popularized as the PGO system by trader *Mark Johnson* on Chuck LeBeau's System Trader's Club forum and elsewhere. This version, the ATR Channel Breakout System, is more flexible and permits the testing of different entry and exit thresholds.

The system is a form of breakout system that buys on the next open when the price closes above the top of the ATR Channel, and exits when the price closes back inside the channel. Short entries are the mirror opposite with selling taking place when the price closes below the bottom of the *ATR Channel*.



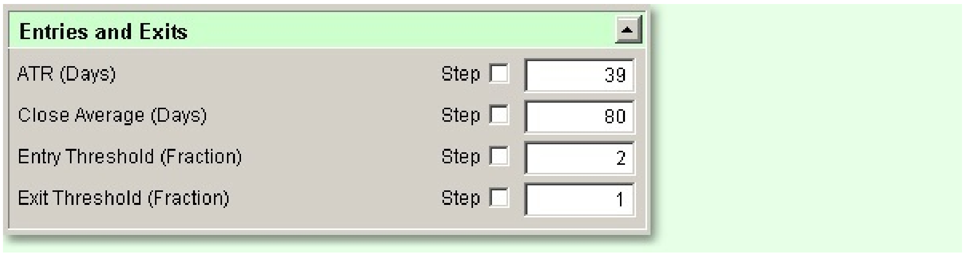
The system trades based on a volatility-band breakout where volatility is measured using Average True Range (ATR). The center of the *ATR Channel* is defined by an *Exponential Moving Average* of the closing prices using a number of days defined by the parameter *Close Average Days*. The top and bottom of the *ATR Channel* are defined using a fixed-multiple of ATR from the moving average specified by the parameter *Entry Threshold*.

The system enters at the open following a day that closes over the top of the *ATR Channel* or below the bottom of the *ATR Channel*. The system exits following a close below the *Exit Channel* which is defined using a fixed-multiple of *ATR* from the moving average specified by the parameter *Exit Threshold*.

This ATR Channel Breakout system is similar to the Bollinger Breakout System except that it uses Average True Range instead of standard deviation as a measure of the volatility which defines the width of the channel.

For example, an *Entry Threshold* of 3 and an *Exit Threshold* of 1 would cause the system to enter the market when the price closed more than 3 ATR above the moving average and to exit when the price subsequently dropped below 1 ATR above the moving average. NOTE: *Exit Threshold* can be a negative number which will cause the system to exit only after the price comes some amount through the moving average.

The ATR Channel Breakout system includes four parameters which affect the entries:



Entries and Exits		
ATR (Days)	Step ☐	39
Close Average (Days)	Step ☐	80
Entry Threshold (Fraction)	Step ☐	2
Exit Threshold (Fraction)	Step ☐	1

ATR Days
The number of days in the *Exponential Moving Average* for the *Average True Range* itself.

Close Average Days
The number of days in the *Exponential Moving Average* of daily closes which forms the center of the ATR channel.

Entry Threshold
The width of the channel in ATR. This defines both the top and bottom of the channel. The system buys or sells to initiate a new position when the closing price crosses the price defined by this threshold.

Exit Threshold
If set to zero, the system will exit when the price closes below the moving average. If set to some higher number the system will exit when the price closes below the given threshold. A negative Exit Threshold means that the exit channel is below the moving average for a long position.

Entry Script

```
' ---------------------------------------------
' Enter orders if channel is breached
' ---------------------------------------------

IF  instrument.position <> LONG AND
    instrument.close > channelTop THEN

    broker.EnterLongOnOpen( exitTop )
ENDIF

IF  instrument.position <> SHORT AND
    instrument.close < channelBottom THEN

    broker.EnterShortOnOpen( exitBottom )
ENDIF
```

Exit Script

```
' ---------------------------------------------
' Exit orders if exit threshold is breached
' ---------------------------------------------

IF  instrument.position = LONG AND
    instrument.close < exitTop THEN
```

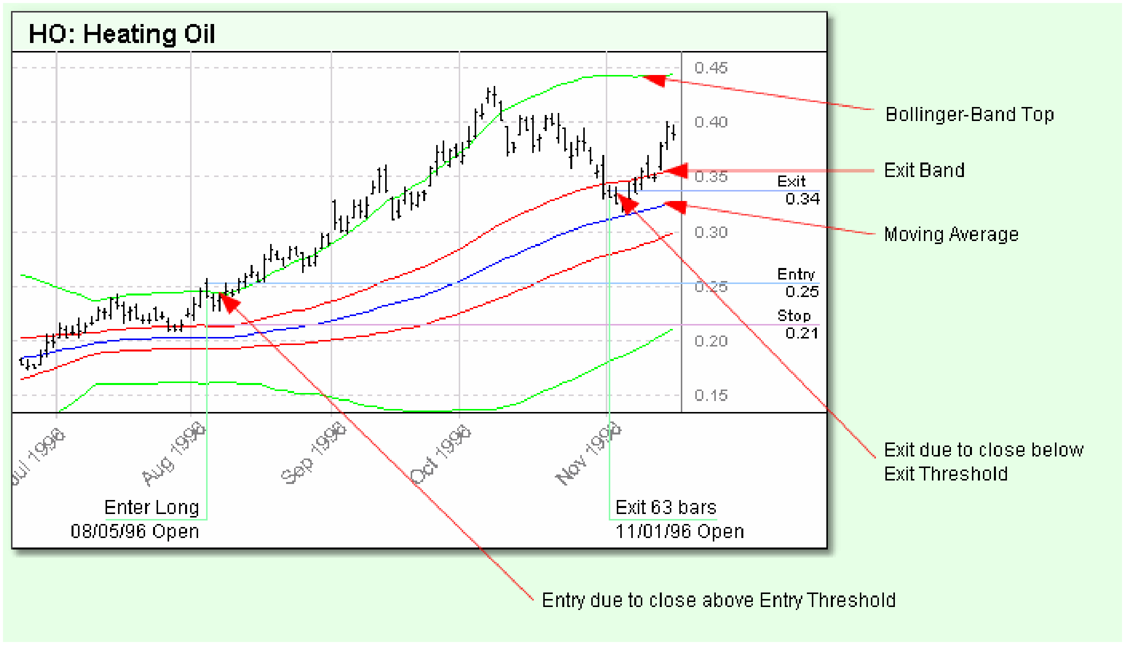
```
        broker.ExitAllUnitsOnOpen
    ENDIF

    IF  instrument.position = SHORT AND
        instrument.close > exitBottom THEN

        broker.ExitAllUnitsOnOpen
    ENDIF
```

Adjust Stops

```
    ' ---------------------------------------------
    ' Adjust stops so money manager knows core equity
    ' ---------------------------------------------

    IF instrument.position = LONG THEN
        instrument.SetExitStop( exitTop )
    ENDIF

    IF instrument.position = SHORT THEN
        instrument.SetExitStop( exitBottom )
    ENDIF
```

Section 8 – Bollinger Breakout System

This system was described by Chuck LeBeau and David Lucas in their 1992 book: "*Technical Traders Guide to Computer Analysis of the Futures Markets*". The system is a form of breakout system that buys on the next open when the price closes above the top of the *Bollinger Band* and exits when the price closes back inside the band. Short entries are the mirror opposite with selling taking place when the price closes below the bottom of the *Bollinger Band*.

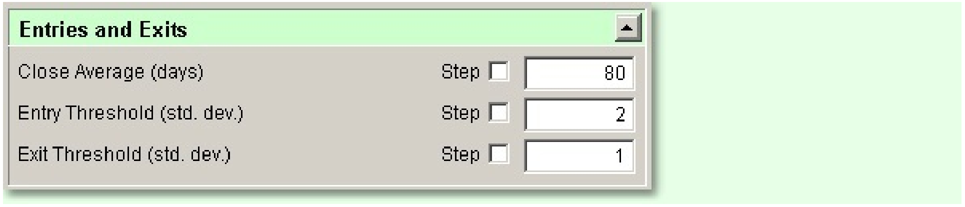


The center of the *Bollinger Band* is defined by an *Simple Moving Average* of the closing prices using a number of days defined by the parameter *Close Average Days*. The top and bottom of the *Bollinger Band* are defined using a fixed-multiple of the standard deviation from the moving average specified by the parameter *Entry Threshold*.

The system enters at the open following a day that closes over the top of the *Bollinger Band* or below the bottom of the *Bollinger Band*. The system exits following a close below the *Exit Band* which is defined using a fixed-multiple of the standard deviation from the moving average specified by the parameter *Exit Threshold*.

The value of the *Exit Band* on the day of entry is used as the stop for the purpose of determining position size using the standard *Fixed Fractional* position sizing algorithm.

The Bollinger Breakout System system includes three parameters that affect the entry and exit:



Entries and Exits

Close Average (days)	Step ☐	80
Entry Threshold (std. dev.)	Step ☐	2
Exit Threshold (std. dev.)	Step ☐	1

Close Average
The number of days in the *Simple Moving Average* which forms the center of the Bollinger Band channel.

Entry Threshold
The width of the channel in standard deviation. This defines both the top and bottom of the channel. The system buys or sells to initiate a new position when the closing price crosses the price defined by this threshold.

Exit Threshold
If set to zero, the system will exit when the price closes below the moving average. If set to some higher number the system will exit when the price closes below the given threshold. A negative Exit Threshold means that the exit channel is below the moving average for a long position.

For example, an *Entry Threshold* of 3 and an *Exit Threshold* of 1 would cause the system to enter the market when the price closed more than 3 standard deviations above the moving average and to exit when the price subsequently dropped below 1 standard deviation above the moving average.

Entry Script

```
' ---------------------------------------------
' Enter orders if they channels are breached
' ---------------------------------------------

IF  instrument.position <> LONG AND
    instrument.close > channelTop THEN

    broker.EnterLongOnOpen( exitTop )
ENDIF

IF  instrument.position <> SHORT AND
    instrument.close < channelBottom THEN

    broker.EnterShortOnOpen( exitBottom )
ENDIF
```

Exit Script

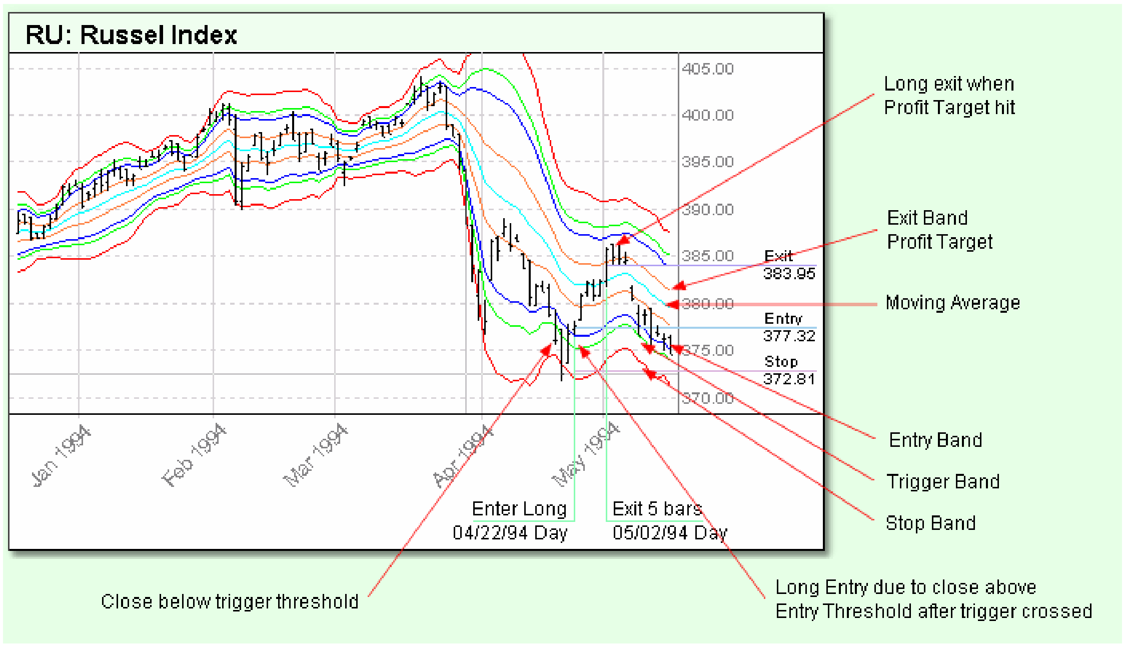
```
' ---------------------------------------------
' Exit orders if exit threshold is breached
' ---------------------------------------------

IF  instrument.position = LONG AND
    instrument.close < exitTop THEN

    broker.ExitAllUnitsOnOpen
ENDIF

IF  instrument.position = SHORT AND
    instrument.close > exitBottom THEN

    broker.ExitAllUnitsOnOpen
ENDIF
```

Adjust Stops

```
' --------------------------------------------
' Adjust stops so money manager knows core equity
' --------------------------------------------

IF instrument.position = LONG THEN
    instrument.SetExitStop( exitTop )
ENDIF

IF instrument.position = SHORT THEN
    instrument.SetExitStop( exitBottom )
ENDIF
```

Section 9 – Bollinger Counter-Trend System

This system trades counter-trend within Bollinger Bands.



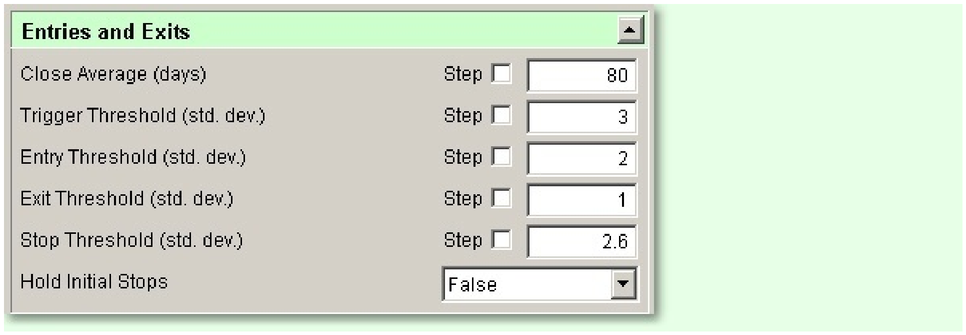
The center of the *Bollinger Band* is defined by an *Simple Moving Average* of the closing prices using a number of days defined by the parameter *Close Average Days*. The top and bottom of the *Bollinger Band* are defined using a fixed-multiple of the standard deviation from the moving average specified by the parameter *Entry Threshold*.

The system sells short when the price drops below the entry threshold after having exceeded the trigger threshold. It then exits using a profit target limit order at the price defined by the parameter *Exit Threshold*.

The system buys long when the price rises above the entry threshold after having gone below the trigger threshold. It then exits using a profit target limit order at the price defined by the parameter *Exit Threshold*.

The trigger is deactivated if the price crosses to the other size of the band and triggers in the opposite direction before a trade is entered. This might occur due to risk, equity, or volume constraints.

The Bollinger Counter-Trend System includes six parameters that affect the entry and exit:



Entries and Exits

Close Average (days)	Step ☐	80
Trigger Threshold (std. dev.)	Step ☐	3
Entry Threshold (std. dev.)	Step ☐	2
Exit Threshold (std. dev.)	Step ☐	1
Stop Threshold (std. dev.)	Step ☐	2.6
Hold Initial Stops		False

Close Average
The number of days in the moving average which forms the center of the Bollinger Band channel.

Trigger Threshold
The amount of standard deviation above or below the moving average which the price must reach before triggering a possible entry.

Entry Threshold
Expressed in terms of standard deviation from the moving average, the point below which the system will initiate a short position. When the price hits the trigger threshold and then subsequently goes below the entry threshold a short position is entered, vice versa for long entries.

Exit Threshold
The profit target for winning trades. The system will exit if the price reaches the exit threshold.

Stop Threshold
The number of standard deviation which will result in a stop loss.

For example, a *Trigger Threshold* of 4, *Entry Threshold* of 3, *Exit Threshold* of 1, and *Stop Threshold* of 4 would cause the system to enter the market on a stop when the price hit the 3 standard deviations above the moving average after having previously exceeded 4 standard deviations above the moving average. The system would exit when the price subsequently dropped below 1 standard deviation above the moving average or once again reached the 4 standard deviation level resulting in a stopped out loss..

NOTE: Exit Threshold can be a negative number which will cause the system to exit only after the price comes some amount through the moving average.

Hold Initial Stops
If set to True, stops will remain at the value on entry. This will prevent stops from moving as the Bollinger Band moves up and down with the moving average. If set to False, then the stop moves to the current value of the Stop Threshold multiplied by the current standard deviation and added to (or subtracted from) the current Close Average value.

Entry Script

```
' ---------------------------------------------
' Enter orders if triggered
' ---------------------------------------------

IF  instrument.position <> LONG AND
    longTriggered THEN

    broker.enterlongonstop( channelBottom, stopBottom )
ENDIF

IF  instrument.position <> SHORT AND
    shortTriggered THEN

    broker.entershortonstop( channelTop, stopTop )
ENDIF
```

Exit Script

```
' -------------------------------------------
' Reset the triggers if we are in a position
' -------------------------------------------

IF instrument.position = LONG THEN

    ' Reset triggers
    longTriggered = FALSE
    shortTriggered = FALSE

    ' exit on limit order
    broker.ExitAllUnitsAtLimit( exitBottom )

ENDIF

IF instrument.position = SHORT THEN

    ' Reset triggers
    longTriggered = FALSE
    shortTriggered = FALSE

    ' exit on limit order
    broker.ExitAllUnitsAtLimit( exitTop )

ENDIF
```

Adjust Stops

```
' -------------------------------------------
' Enter stop if "holdstops" is true
' -------------------------------------------

IF holdStops THEN

    broker.ExitAllUnitsOnStop( instrument.unitExitStop )

ENDIF
```

After Instrument Day

```
' -------------------------------------------
' Check the triggers
' -------------------------------------------

'Check the short trigger.
IF  NOT shortTriggered AND
    instrument.position = OUT AND
    instrument.high > triggerTop THEN

    shortTriggered = TRUE
    longTriggered = FALSE
ENDIF

'Check the long trigger.
```

```
IF  NOT longTriggered AND
    instrument.position = OUT AND
    instrument.low < triggerBottom THEN

    shortTriggered = FALSE
    longTriggered = TRUE
ENDIF
```

Section 10 – Donchian System

Donchian System

The Donchian System is based on the Turtle system. It uses the Turtle logic, except it is single unit, does not use the Last Trade is Winner rule, does not use correlations, and uses a MACD Portfolio Manager to filter trades.

The Donchian System trades on breakouts similar to a Donchian Dual Channel system. There are two breakout figures, a longer breakout for entry, and a shorter breakout for exit.

The Donchian system uses a stop based on the *Average True Range* (ATR).
Note that the Turtle concept of N has been replaced by the more common and equivalent term *Average True Range* (ATR) in this document.

Entry Breakout (days)
A trade is entered when the price hits the high or the low of the preceding X-days. For example, Entry Breakout = 20 means that a long position is taken if price hits the 20-day high; A short position is taken if the price hits the 20-day low.

Entry Offset (ATR)
If set to zero, this parameter has no effect. If *Entry Offset in ATR* is set to 1.0, a long position isn't entered until price hits the normal breakout price, plus 1.0 ATR. Likewise, a short position won't be entered until the price hits the normal breakout price, minus 1.0 ATR. Either a positive or negative value can be specified for this parameter. A positive value effectively delays entry until the specified point after the breakout threshold chosen; a negative value would enter before the breakout threshold chosen.

Stop (ATR)
This parameter defines the distance from the entry price to the initial stop, in terms of ATR. This system by default uses the order entry price, not the fill price, as a basis of the stop price. Since ATR is a measure of daily volatility and the Turtle System stops are based on ATR, this means that the Donchian System equalizes the position size across the various markets based on volatility.
According to the original Turtle Rules, long positions were stopped out if price fell 2 ATR from the entry price. Conversely, short positions were stopped out if the price rose 2 ATR from the entry price.
Unlike the *Exit Breakout* based stop, which moves up or down with the X-day high or low, the stop defined by *Stop in ATR* is a "hard" stop that is fixed above or below the entry price upon entry. Once set, it does not vary throughout the course of the trade

Note that trades are liquidated when price hits either the Exit Breakout, Entry Breakout for the opposite direction, or the Stop in ATR, whichever is closest to the price at the time.

Exit Breakout (days)
Trades in progress are exited when the price hits the high or the low of the preceding X-days. This concept is the identical to Entry Breakout, but the logic is reversed: Long trades are exited when price hits the X-day low, and short trades are exited when the price hits the X-day low. The Exit Breakout moves up (or down) with price. It protects against adverse price excursions, and also serves as a trailing stop that acts to lock in a profit when the trend reverses.

Note that trades are liquidated when price hits either the Exit Breakout, Entry Breakout for the opposite direction, or the Stop in ATR, whichever is closest to the price at the time.

These options can be enabled or disabled with the Hold Initial Stops and Use Reversal Exit parameters. If the initial stop is held, then the initial stop price will be used to exit during the trade.

If using the reversal exit, then the trade will be exited if the price hits the entry breakout for the opposite direction.

Exit Offset (ATR)
If set to zero, this parameter has no effect. If Exit Offset in ATR is set to 1.0, a long position isn't exited until price hits the normal breakout price, minus 1.0 ATR. Likewise, a short position won't be exited until the price hits the normal breakout price, plus 1.0 ATR. Either a positive or negative value can be specified for this parameter. A positive value effectively delays exit until the specified point after the breakout threshold chosen; a negative value would exit before the breakout threshold chosen.

ATR (days)
Defined the number of days for the ATR calculation. This is an exponential moving average of the True Range. 39 represents a 20 day Wilder ATR.

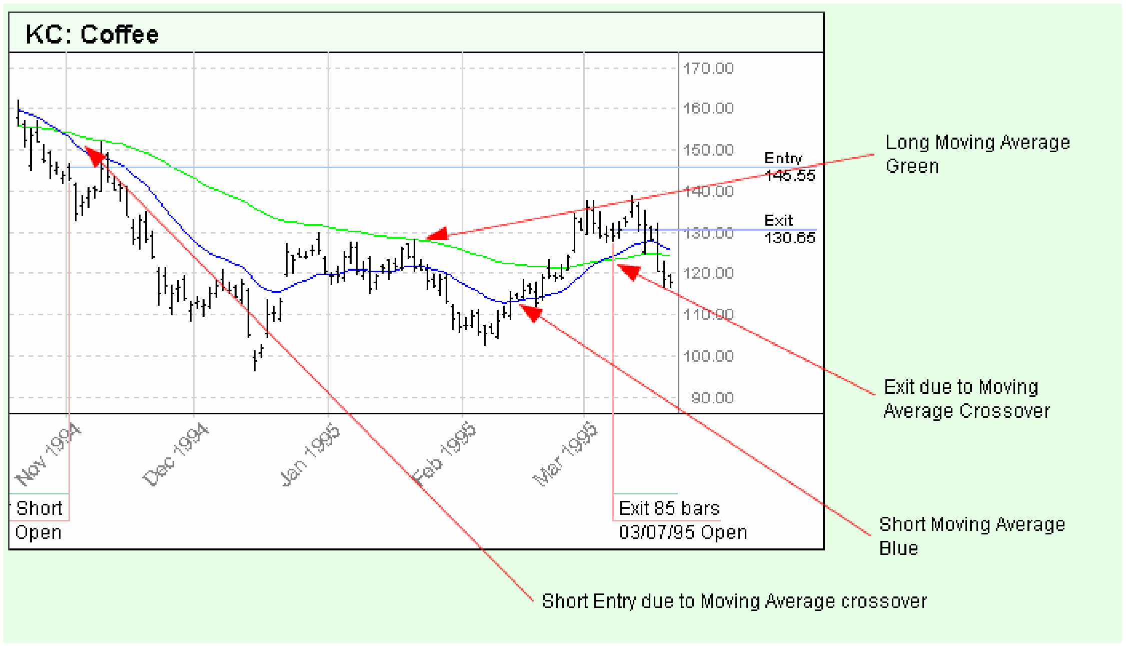
MACD Long Average (days)
This is the number of days for the long moving average portion of the MACD indicator.

MACD Short Average (days)
This is the number of days for the short moving average portion of the MACD indicator.
The MACD itself is the Short Moving Average minus the Long Moving Average.
The system will allow Long trades when the MACD is greater than zero and allow Short trades when the MACD is less than zero.

Money Manager
This system uses the Fixed Fractional Money Manager

Section 11 – Dual Moving Average System

This system uses two moving averages, one short and one long. The system trades when the short moving average crosses the long moving average.

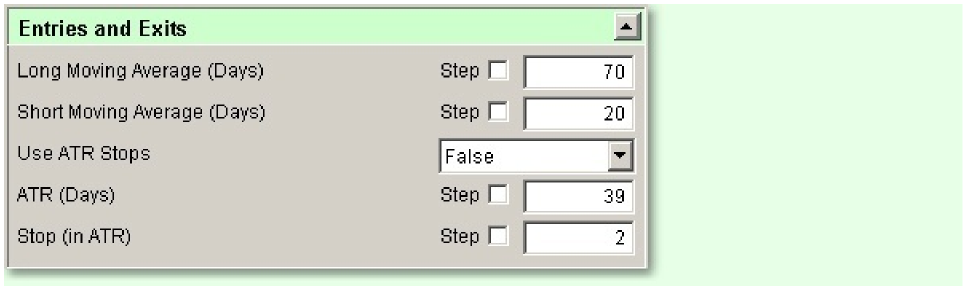


The system optionally uses a stop based on *Average True Range* (ATR). If the ATR stop is used, the system will exit the market when that stop is hit.

If the ATR stop is not used, the system does not have an explicit stop and will always be in the market, making it a reversal system. It will exit a position only when the moving averages cross. At that point, it will exit and enter a new position in the opposite direction. In this case, the positions are sized based only on ATR using a custom money manager.

If an ATR stop is not used, then the entry risk is essentially infinite. This will cause the R-Multiple™s relatively meaningless since all gains will be less than the infinite risk of entering without any stop.

The Dual Moving Average System includes five parameters that affect the entries:



Entries and Exits

Long Moving Average (Days)	Step ☐	70
Short Moving Average (Days)	Step ☐	20
Use ATR Stops		False
ATR (Days)	Step ☐	39
Stop (in ATR)	Step ☐	2

Long Moving Average
The number of days in the long moving average.

Short Moving Average
The number of days in the short moving average.

Use ATR Stops
If set to TRUE then the system will enter a stop based on a certain number of ATR from the entry point.

ATR Days
The number of days used for the ATR calculation. This parameter is visible and active only if *Use ATR Stops* is TRUE.

Stop
The stop width expressed in terms of ATR. This parameter is visible and active only if *Use ATR Stops* is TRUE.

If the *Use ATR Stops* is FALSE there are no stops, but the system uses a theoretical 1 ATR stop for the purposes of position sizing.

Entry Script

```
VARIABLES: currentclose, stopPrice TYPE: Price

' Get the current close.
currentClose = instrument.close

' If we are not long and the faster moving averages are above the slower one.
IF  instrument.position <> LONG AND
    shortMovingAverage > longMovingAverage THEN

    ' If we are using ATR stops...
    IF useATRStops THEN
        ' Compute the stop price.
        stopPrice = currentClose - (stopInATR * averageTrueRange)

        ' Enter a long on the open using this stop.
        broker.EnterLongOnOpen( stopPrice )

    ' NOT using ATR stops...
    ELSE
        ' Enter a long on the open with no stop.
        broker.EnterLongOnOpen
    ENDIF
ENDIF

' If we are not short and the faster moving averages are below the slower one.
IF  instrument.position <> SHORT AND
    shortMovingAverage < longMovingAverage THEN

    ' If we are using ATR stops...
    IF useATRStops THEN
        ' Compute the stop price.
        stopPrice = currentClose + (stopInATR * averageTrueRange)

        ' Enter a short on the open using this stop.
        broker.EnterShortOnOpen( stopPrice )
    ' NOT using ATR stops...
    ELSE
        ' Enter a short on the open with no stop.
        broker.EnterShortOnOpen
    ENDIF
ENDIF
```

Exit Script

```
    ' Exit position on open if moving averages cross

    IF instrument.position = LONG AND
        shortMovingAverage < longMovingAverage THEN
            broker.ExitAllUnitsOnOpen
    ENDIF

    IF instrument.position = SHORT AND
        shortMovingAverage > longMovingAverage THEN
            broker.ExitAllUnitsOnOpen
    ENDIF
```

Adjust Stops

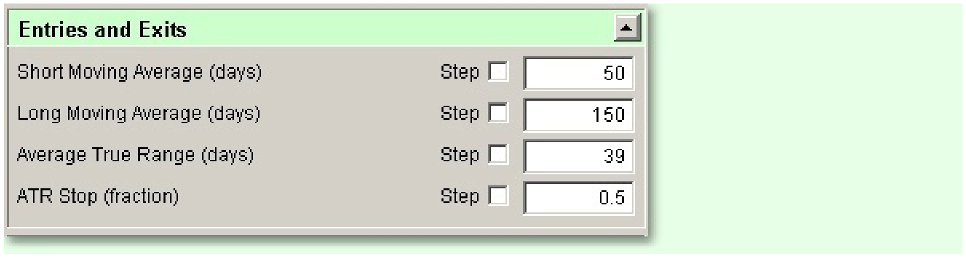
```
    ' ---------------------------------------------
    ' Enter stop if "Use ATR Stop" is true
    ' ---------------------------------------------

    IF useATRStops THEN

        broker.ExitAllUnitsOnStop( instrument.unitExitStop )

    ENDIF
```

Section 12 – MACD System

The Moving Average Convergence/Divergence indicator is a centered oscillator that shows the difference between two moving averages, typically 12 and 26 days. These values, like any other parameter, can be altered to show the MACD over a different period of time.

The MACD system buys when the MACD goes above zero and sells when it goes below zero. So it's always in the market. Very similar in function to the Dual Moving Average System.

The MACD system uses the following parameters:


Entries and Exits

Short Moving Average (days)	Step ☐	50
Long Moving Average (days)	Step ☐	150
Average True Range (days)	Step ☐	39
ATR Stop (fraction)	Step ☐	0.5

Short Moving Average Days
Number of days used to calculate the short moving average

Long Moving Average Days
Number of days used to calculate the long moving average

Average True Range Days
Number of days used to calculate the Average True Range

ATR Stop
ATR Stop (fraction): Fraction of the ATR used for stops

Entry Script

```
IF  macdIndicator > 0 AND
    instrument.position <> LONG THEN

    IF useATRStops THEN
        broker.EnterLongOnOpen( instrument.close - averageTrueRange * atrStop )
    ELSE
        broker.EnterLongOnOpen
    ENDIF

ENDIF

IF  macdIndicator < 0 AND
    instrument.position <> SHORT THEN

    IF useATRStops THEN
        broker.EnterShortOnOpen( instrument.close + averageTrueRange * atrStop )
    ELSE
        broker.EnterShortOnOpen
    ENDIF

ENDIF
```

Adjust Stops

```
' ---------------------------------------------
' Enter stop if "holdstops" is true
' ---------------------------------------------

IF useATRStops THEN

    broker.ExitAllUnitsOnStop( instrument.unitExitStop )

ENDIF
```

Section 13 – RSI Trend Catcher System

The RSI Trend Catcher uses the Relative Strength Index to generate buy and sell signals.

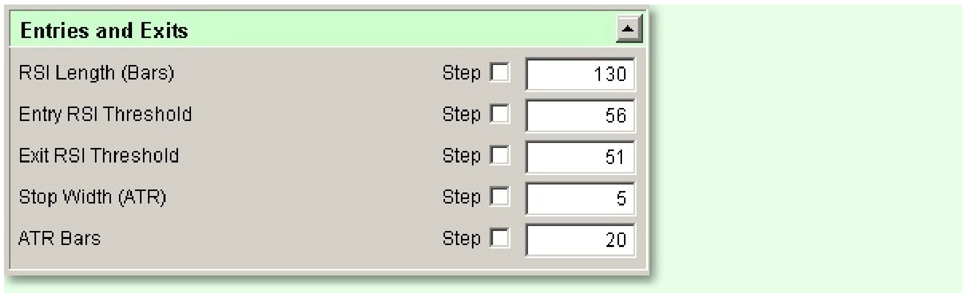
It uses a threshold range above which it will buy and below which it will sell. The range it determined as:

Top of range - entered by user as entryThreshold
Bottom of range - (100 - entryThreshold)

It sets the stops stopWidth fraction of the ATR from the close.

It uses a threshold range for exits. Above which it exits a short position, and below which it exits a long position.

The RSI Trend Catcher System uses the following parameters:



Entries and Exits

RSI Length (Bars)	Step ☐	130
Entry RSI Threshold	Step ☐	56
Exit RSI Threshold	Step ☐	51
Stop Width (ATR)	Step ☐	5
ATR Bars	Step ☐	20

RSI Length (Bars)
Number of bars used to calculate the RSI

Entry RSI Threshold
The top of the entry range above which the system will enter a long position. 100 - entryThreshold is the bottom of the entry range, below which the system will enter a short position.

Exit RSI Threshold
The bottom of the exit range, below which the system will exit a long position. 100 - exitThreshold is the top of the exit range, above which the system will exit a short position.

Stop Width (ATR)
Fraction of the ATR used for stop. Stops are based off the Close

ATR Bars
Number of bars used to calculate the Average True Range

Entry Script

```
' ----------------------------------------------
' Enter position if RSI crosses threshold.
' ----------------------------------------------
VARIABLES: exitStop TYPE: Floating

IF  instrument.position <> LONG AND
    relativeStrengthIndex > entryThreshold THEN
```

```
        ' Compute the stop.
        exitStop = instrument.close - (stopWidth * averageTrueRange)

        ' Enter a long on the open with this stop.
        broker.EnterLongOnOpen( exitStop )
    ENDIF

    IF  instrument.position <> SHORT AND
        relativeStrengthIndex < (100 - entryThreshold) THEN

        ' Compute the stop.
        exitStop = instrument.close + (stopWidth * averageTrueRange)

        ' Enter a long on the open with this stop.
        broker.EnterShortOnOpen( exitStop )
    ENDIF
```

Exit Script

```
    ' ----------------------------------------------
    ' Exit Position if RSI crosses Threshold
    ' ----------------------------------------------

    IF instrument.position = LONG AND
        relativeStrengthIndex <= exitThreshold THEN

        ' Exit the position.
        broker.ExitAllUnitsOnOpen
    ENDIF

    IF instrument.position = SHORT AND
        relativeStrengthIndex >= (100 - exitThreshold) THEN

        ' Exit the position.
        broker.ExitAllUnitsOnOpen
    ENDIF
```

Adjust Stops

```
    ' ----------------------------------------------
    ' Enter stop if "holdstops" is true
    ' ----------------------------------------------

    IF holdStops THEN

        broker.ExitAllUnitsOnStop( instrument.unitExitStop )

    ENDIF
```

Section 14 – Stochastic System

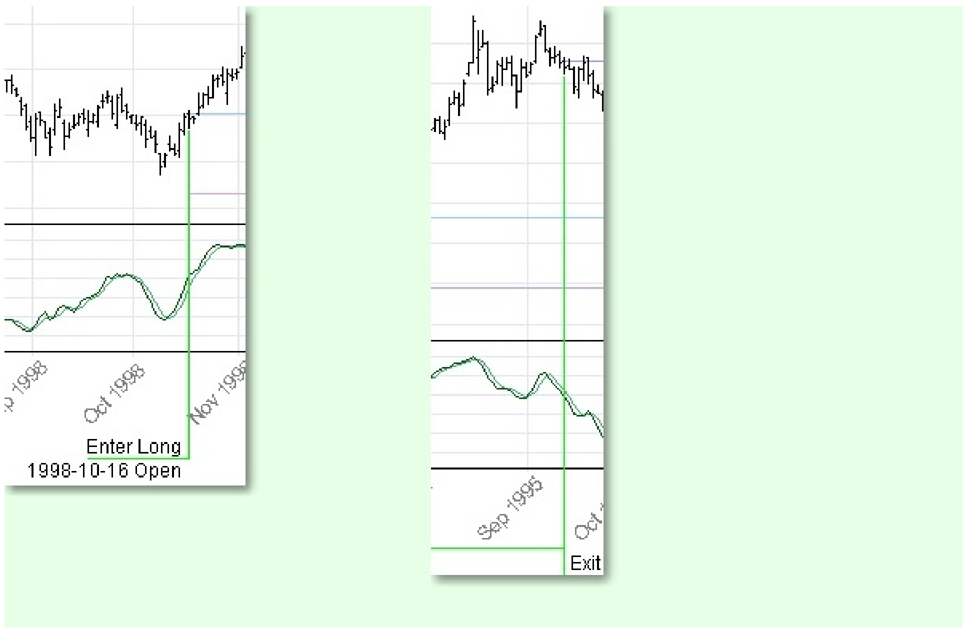
The Stochastic Oscillator is a centered momentum oscillator. It is a percentile that shows a recent close in relationship to a high and low over a certain period of time. Stochastics were originally designed for stocks in a trading market. Although the parameters can be tweaked to give good results in almost any market, the default values reflect good results with stocks over the past 10 years.

Our built-in stochastic system uses convergence for signals. Positive convergence of a certain strength (called a swing factor), during a certain number of bars, and rising generate a long signal. The opposite exits a long position. Our system does not take short trades at all.

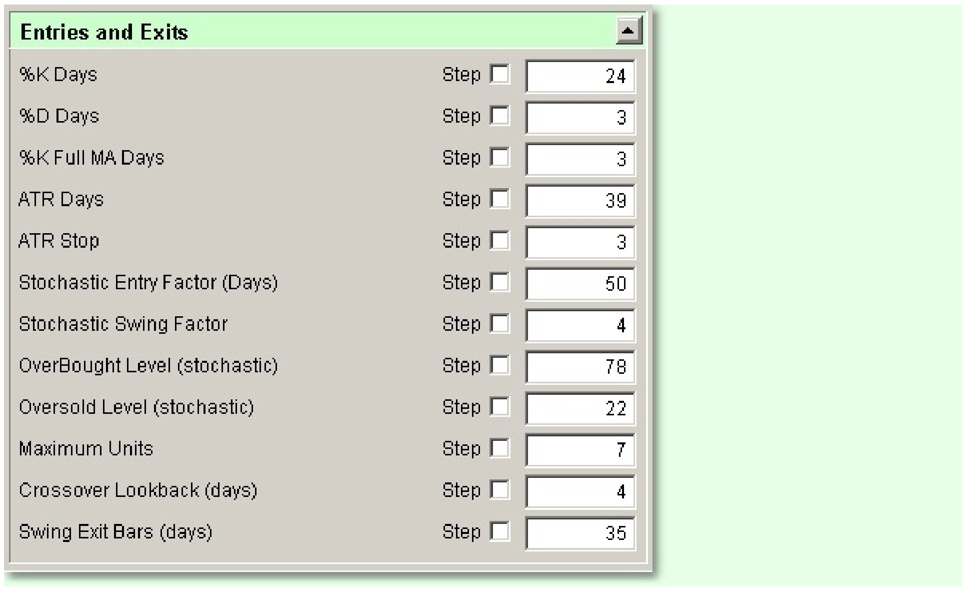
The conditions used to generate a long signal are as follows:

- There is a positive convergence that develops below a %K(Full) of 22.
- The %K(Full) is above 22 (this can be customized) and has started rising above %K(Slow) over the past 4 days.
- The convergence must take place over a certain time.
- The convergence must include highs and lows of a certain profundity to avoid meaningless signals.

The conditions for selling are much the same, except the convergence must be negative and falling below 78 (also customizable). Below are examples of both signals. Note the positive and negative convergences and rising or falling %K(Full).



The following values are used for the Stochastic System. The values shown are the default values.



%K Days	Step	☐	24
%D Days	Step	☐	3
%K Full MA Days	Step	☐	3
ATR Days	Step	☐	39
ATR Stop	Step	☐	3
Stochastic Entry Factor (Days)	Step	☐	50
Stochastic Swing Factor	Step	☐	4
OverBought Level (stochastic)	Step	☐	78
Oversold Level (stochastic)	Step	☐	22
Maximum Units	Step	☐	7
Crossover Lookback (days)	Step	☐	4
Swing Exit Bars (days)	Step	☐	35

%K Days
The number of days used to determine %K(Fast).

%D Days
The number of days used to determine %D, a moving average of %K.

%K Full MA Days
The number of days used to smooth %K(Full).

ATR Days
The number of days used to calculate *average true range*.

ATR Stop
A factor used to determine stops based on ATR.

Stochastic Swing Factor (Days)
The number of days high and low stochastics are considered in determining convergence.

Stochastic Swing Factor
The depth, in stochastic units, that highs and lows must be from each other. If this is too high, only very large fluctuations will produce signals. If this is too small, insignificant variations will generate signals.

Overbought Level
The range, in stochastic units, that is considered overbought.

Oversold Level
The range, in stochastic units, that is considered oversold.

Maximum Units
The maximum units that may be in the long position at any one time.

Crossover Lookback (Days)
The number of days that a crossover between %K(Slow) and %K(Full) is looked for.

Swing Exit Bars (Days)

Akin to Stochastic Swing Factor (Days), except only used to exit a long position.

Entry Script

```
' We are looking for a specific pattern here
' A convergence at a certain over-sold level, with a cross over

IF
    stochasticHigh2 > overSoldLevel AND
    stochasticHigh1 > overSoldLevel AND
    stochasticLow1 < overSoldLevel AND
    stochasticLow2 < overSoldLevel AND
    stochasticKSlow > stochasticKFull AND
    stochasticKSlow[crossoverLookback] < stochasticKFull[crossoverLookback] AND
    stochasticLow1 > stochasticLow2 AND
    stochasticHigh1 < stochasticHigh2 AND
    instrument.bar - stochasticLow2Bar < swingEntryBars THEN

    IF instrument.totalUnits < maxUnits THEN

        IF useATRStops THEN
            broker.EnterLongOnOpen( instrument.close - averageTrueRange * atrStop )
        ELSE
            broker.EnterLongOnOpen
        ENDIF

    ENDIF
ENDIF
```

Exit Script

```
' We are looking for an exit point -- opposite logic to the entry point

IF
    stochasticHigh2 > overBoughtLevel AND
    stochasticHigh1 > overBoughtLevel AND
    stochasticLow1 < overBoughtLevel AND
    stochasticLow2 < overBoughtLevel AND
    stochasticKSlow < stochasticKFull AND
    stochasticKSlow[crossoverLookback] > stochasticKFull[crossoverLookback] AND
    stochasticLow1 > stochasticLow2 AND
    stochasticHigh1 < stochasticHigh2 AND
    instrument.bar - stochasticLow2Bar < swingExitBars THEN

    broker.ExitAllUnitsOnOpen
ENDIF
```

Adjust Stops

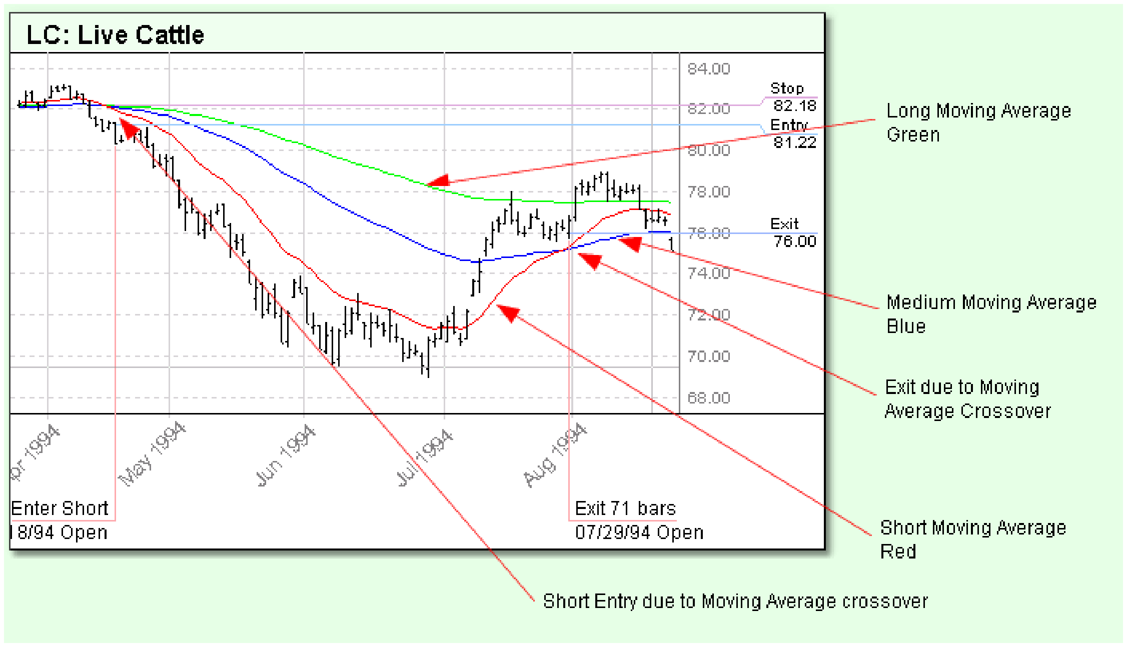
```
' ---------------------------------------------
' Enter stop if "Use ATR Stops" is true
' ---------------------------------------------

IF useATRStops THEN

    FOR unitIndex = 1 to instrument.totalUnits
        broker.ExitUnitOnStop( unitIndex, instrument.unitExitStop[ unitIndex ] )
    NEXT

ENDIF
```

After Instrument Day Script

```
    IF lookingForHigh = TRUE THEN
        IF stochasticKFull > stochasticHigh1 THEN
            'Finds new high

            stochasticHigh1 = stochasticKFull
            stochasticHigh1Bar = instrument.bar
        ENDIF

        IF stochasticHigh1 - stochasticKFull > swingFactor THEN
            'The high found is significant based on swingFactor

            lookingForLow = TRUE
            lookingforHigh = FALSE

            stochasticLow2 = stochasticLow1
            stochasticLow2Bar = stochasticLow1Bar

            stochasticLow1 = stochasticKFull
            stochasticLow1Bar = instrument.bar
        ENDIF
    ENDIF

    IF lookingForLow = TRUE THEN

        IF stochasticKFull < stochasticLow1 THEN
            stochasticLow1 = stochasticKFull
            stochasticLow1Bar = instrument.bar
        ENDIF

        IF stochasticKFull - stochasticLow1 > swingFactor THEN
            lookingForHigh = TRUE
            lookingForLow = FALSE

            stochasticHigh2 = stochasticHigh1
            stochasticHigh2Bar = stochasticHigh1Bar

            stochasticHigh1 = stochasticKFull
            stochasticHigh1Bar = instrument.bar
        ENDIF
    ENDIF
```

Section 15 – Triple Moving Average System

This system uses three *moving averages*, one short, one medium, and one long. The system trades long when the short *moving average* is higher than the medium *moving average* and the medium *moving average* is higher than the long moving average. When the short *moving average* is back below the medium *moving average*, the system exits. The reverse is true for short trades.



For this reason, unlike the Dual Moving Average system, this system is not always in the market. The system is out of the market when the relationship between the short MA and medium MA does not match the relationship between the medium MA and long MA.

For example, considering Long trades, if the short MA is over the medium MA but the medium MA is under the long MA, the system is out of the market. Likewise if the medium MA is over the long MA but the short MA is not over the medium MA the system is out of the market.

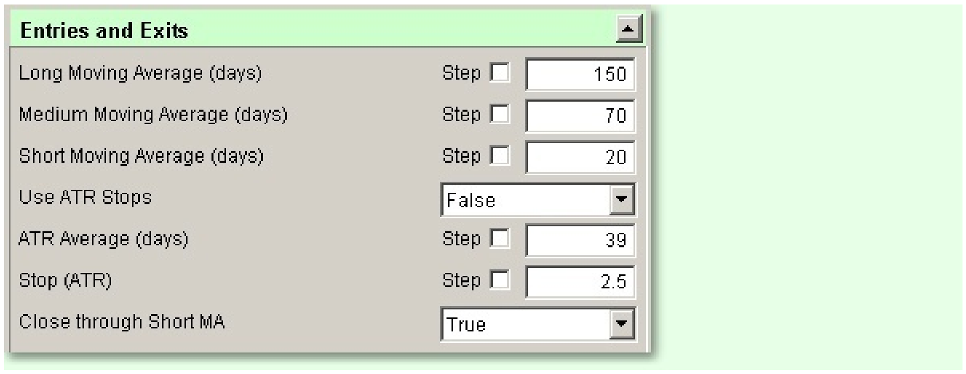
This means that the system can initiate trades due to either:

- Short MA is above the medium MA for Long entries or to below for Short entries. This is the most common case.
- Medium MA is above the long MA where the short MA is already over the medium MA for Long entries, or to below the long MA where the short MA is already under the medium MA for Short entries . This will happen when the market has been descending or ascending for a long time and then reverses direction. It takes longer for the medium MA to move to the other side of the long MA since they are both slower moving averages than the short MA.

The system optionally uses a stop based on *Average True Range* (ATR). If the ATR stop is not used, the system uses the value of the long moving average as the stop for the purpose of position sizing.

In the event of a stop out, the system will reenter whenever the above conditions are true, even if this is the following day's open.

The system includes seven parameters that affect the entries:


Entries and Exits

Long Moving Average (days)	Step ☐	150
Medium Moving Average (days)	Step ☐	70
Short Moving Average (days)	Step ☐	20
Use ATR Stops		False
ATR Average (days)	Step ☐	39
Stop (ATR)	Step ☐	2.5
Close through Short MA		True

Long Moving Average
The number of days in the long moving average.

Medium Moving Average
The number of days in the medium moving average.

Short Moving Average
The number of days in the short moving average.

Use ATR Stops
If set to TRUE then the system will enter a stop based on a certain number of ATR from the entry point.

ATR Average
The number of days used for the ATR calculation. This parameter is visible and active only if *Use ATR Stops* is TRUE.

Stop
The stop width expressed in terms of ATR. This parameter is visible and active only if *Use ATR Stops* is TRUE.

If the *Use ATR Stops* is FALSE the system computes a stop at the price of the long moving average for the purposes of position sizing. In this case, the stop is active only for the first day.

Close through Short MA
If set to True, Trading Blox won't take a trade unless the close is also on the right side of the short moving average. For example, with this parameter set to True, in addition to the short moving average being over the medium moving average and the medium moving average being over the long moving average, the close must be *above* the short moving average in order to trigger a Long position entry.

Entry Script

```
' Get the current close.
currentClose = instrument.close


' If we are not long and the faster moving averages are above the slower ones.
IF  mediumMovingAverage > longMovingAverage AND
    shortMovingAverage > mediumMovingAverage AND
    ((NOT closeThroughShortMA) OR currentClose > shortMovingAverage)  AND
```

```
        instrument.position != LONG THEN

        ' Compute the stop.
        IF stopType = ATR_STOP THEN
            stopPrice = currentClose - (stopInATR * averageTrueRange)
        ELSE
            stopPrice = longMovingAverage
        ENDIF

        ' Enter a long at the open using this stop.
        broker.EnterLongOnOpen( stopPrice )

    ENDIF

    ' If we are not short and the faster moving averages are below the slower ones.
    IF  mediumMovingAverage < longMovingAverage AND
        shortMovingAverage < mediumMovingAverage AND
        ((NOT closeThroughShortMA) OR currentClose < shortMovingAverage) AND
        instrument.position != SHORT THEN

        ' Compute the stop.
        IF stopType = ATR_STOP THEN
            stopPrice = currentClose + (stopInATR * averageTrueRange)
        ELSE
            stopPrice = longMovingAverage
        ENDIF

        ' Enter a short at the market using this stop.
        broker.EnterShortOnOpen( stopPrice )
    ENDIF
```

Exit Script

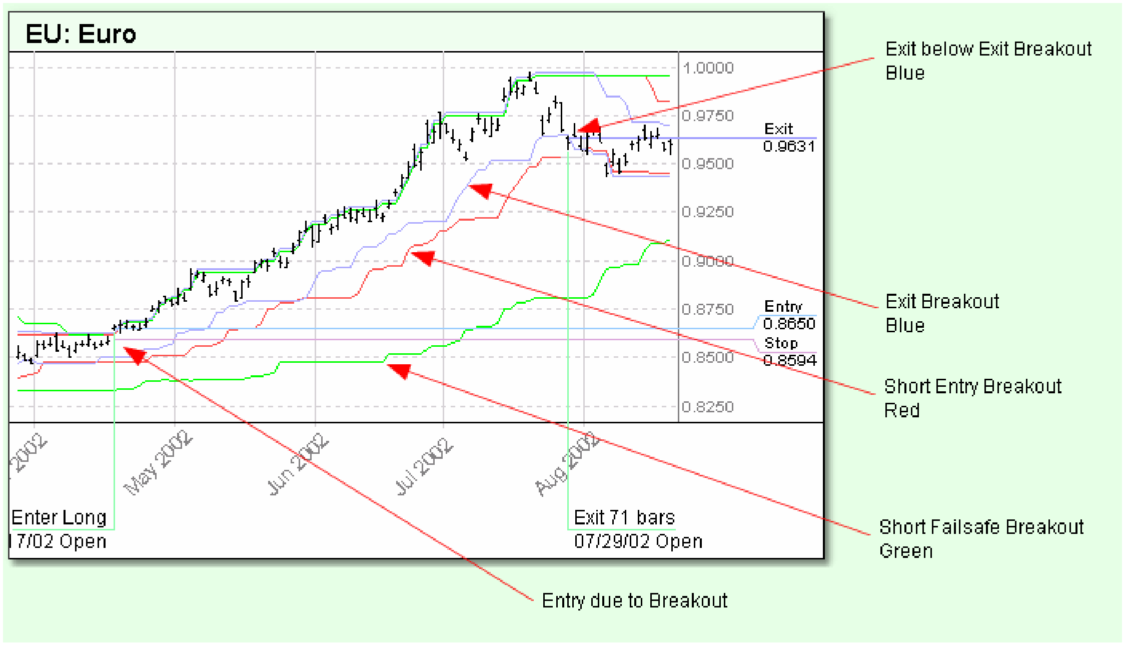
```
    ' If we are long and the short moving average crosses the medium.
    IF  instrument.position = LONG AND
        shortMovingAverage < mediumMovingAverage THEN

        ' Exit on the open
        broker.ExitAllUnitsOnOpen
    ENDIF

    ' If we are short and the short moving average crosses the medium.
    IF  instrument.position = SHORT AND
        shortMovingAverage > mediumMovingAverage THEN

        ' Exit on the open
        broker.ExitAllUnitsOnOpen
    ENDIF
```

Adjust Stops

```
    ' ---------------------------------------------
    ' Enter stop if "holdstops" is true
    ' ---------------------------------------------

    IF holdStops THEN

        broker.ExitAllUnitsOnStop( instrument.unitExitStop )

    ENDIF
```

Section 16 – Turtle System Rules

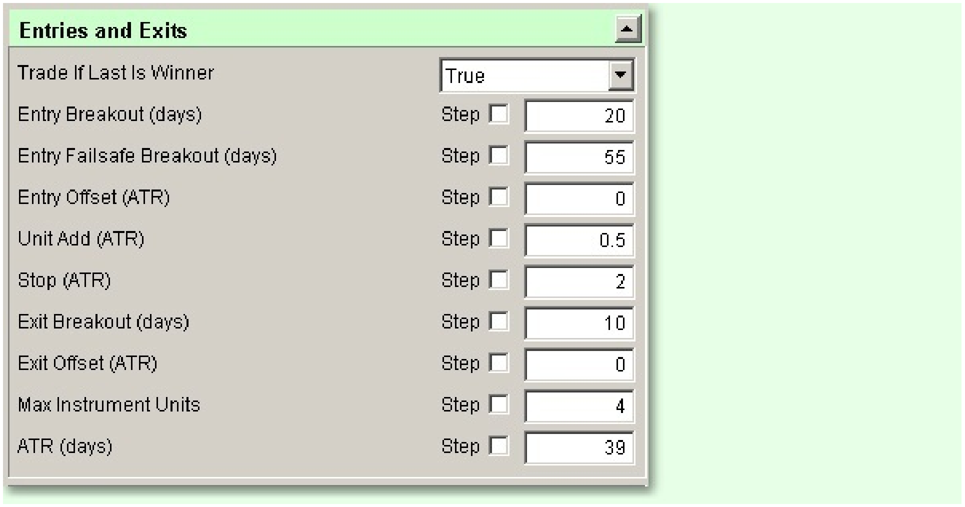
The rules to the Turtle System as implemented in Trading Blox are described in the Original Turtle rules document at the site: www.originalturtles.org. Please consult that document for a complete description of the Turtle System rules.



The Turtle System trades on breakouts similar to a Donchian Dual Channel system. There are two breakout figures, a longer breakout for entry, and a shorter breakout for exit. The system also optionally uses a dual-length entry where the shorter entry is used if the last trade was a losing trade.

The Turtle system uses a stop based on the *Average True Range* (ATR).

Note that the Turtle concept of N has been replaced by the more common and equivalent term *Average True Range* (ATR) in this document.



Entries and Exits

Trade If Last Is Winner		True
Entry Breakout (days)	Step ☐	20
Entry Failsafe Breakout (days)	Step ☐	55
Entry Offset (ATR)	Step ☐	0
Unit Add (ATR)	Step ☐	0.5
Stop (ATR)	Step ☐	2
Exit Breakout (days)	Step ☐	10
Exit Offset (ATR)	Step ☐	0
Max Instrument Units	Step ☐	4
ATR (days)	Step ☐	39

Trade Long/Short

This parameter tells Trading Blox whether or not trades in the short direction are to be taken.

Trade if Last is Winner
When this parameter is set to False (unchecked and disabled), Trading Blox looks back at the last entry breakout for that instrument and determines if it would have been a winner, either actually, or theoretically. If the last trade was, or would have been a winner, then the next trade is skipped, regardless of direction (long or short).

The last breakout is considered to be the last breakout in that market regardless of whether or not that particular breakout was actually taken, or was skipped because of this rule. (Trading Blox looks back only at "regular" breakouts, and not Entry Failsafe Breakouts.)

The direction of the last breakout-long or short-is irrelevant to the operation of this rule, as is the direction of the trade currently being considered. Thus, a losing long breakout or a losing short breakout, whether hypothetical or actual, would enable the subsequent new breakout to be taken as a valid entry, regardless of its direction (long or short):

Some traders believe that two large, consecutive wins are unlikely, or that a profitable trade is more likely to follow a losing trade. Trading Blox allows you to test this idea by setting this parameter to False.

Entry Breakout (days)
A trade is entered when the price hits the high or the low of the preceding X-days, as adjusted by the *Entry Offset*. For example, Entry Breakout = 20 means that a long position is taken if price hits the 20-day high; A short position is taken if price hits the 20-day low.

Entry Failsafe Breakout (days)
This parameter works in concert with Trade if Last is Winner, and is used only if Trade if Last is Winner = False (as is shown in the partial screen shot above).

For example, consider the following set of parameters and values:

With these settings, if a 20-day breakout entry was recently signaled, but was skipped because the prior trade was a winner (either actually, or theoretically), then if the price breaks out above or below the 55-day extreme high or low, an entry is initiated for that position regardless of the outcome of the prior trade.

Entry Failsafe Breakout keeps you from missing very strong trends due to the action of the Trade if Last is Winner rule.

Entry Offset (ATR)
If set to zero, this parameter has no effect. If *Entry Offset in ATR* is set to 1.0, a long position isn't entered until price hits the normal breakout price, plus 1.0 ATR. Likewise, a short position won't be entered until the price hits the normal breakout price, minus 1.0 ATR. Either a positive or negative value can be specified for this parameter. A positive value effectively delays entry until the specified point after the breakout threshold chosen; a negative value would enter before the breakout threshold chosen.

Unit Add (ATR)
This parameter defines the price at which additions to an existing position are made. The Turtles entered single Unit positions at the breakouts, and added to those positions at 1/2 ATR intervals following trade initiation. (Adding to existing positions is often referred to as "pyramiding.")

Following the initial breakout entry, Trading Blox will continue to add a Unit (or Units, in the case of a large price move in a single day), at each interval defined by *Unit Add in ATR*, as price progresses favorably, right up to the maximum permitted number of Units, as specified by the various *Max Units* rules (explained below).

During historical simulation tests, the theoretical entry price is adjusted up or down by *Slippage Percent* and/or *Minimum Slippage*, to obtain the simulated fill price. So each interval is based on the simulated fill price of the previous order. So if an initial breakout order slipped by 1/2 ATR, the new order would be moved to account for the 1/2 ATR slippage, plus the normal unit add interval specified by *Unit Add in ATR*.

The exception to this rule is when multiple Units are added in a single day during a trade in progress. For example, with *Unit Add in ATR* = 0.5, the initial breakout order is placed and incurs slippage of 1/2 ATR. Several days later, two more units are added on the same day. In this case, the order price of both the 2nd and 3rd Units is adjusted up by 1/2 ATR (to 1 full ATR past the breakout), based on the slippage incurred by the 1st Unit. Ordinarily, in the case where several Units have been added (each on a separate day), the order price of each Unit is adjusted by the cumulative slippage (in N) of all the Units that preceded it on the trade in progress.

Notes:

- Users who generate orders will need to adjust the order price of any Units that are to be added, based on the cumulative slippage (in ATR) of all preceding Units. Trading Blox does not account for slippage and actual fills.
- The initial Unit, as well as any Units added, are all sized based on a combination of the current value of ATR, and the current available account balance. (This differs slightly from the Original Turtle Trading Rules in that the Turtles were only given an updated value of ATR once a week). So do not be surprised, for example, if the initial Unit size of a Wheat trade is 7 contracts, and the size of the next Unit added to this trade is 5 contracts.
- As per the Original Turtle Rules, the prices for stops of previous units are raised as new units are added. However, since the Unit Add in ATR parameter can be larger than the Stop in ATR parameter, it is possible that prices for stops would be raised above the entry price. In keeping with the spirit of the Turtle System Trading Blox does not allow stops to be raised above the initial entry price when adjusting stops based on the addition of new units..

Stop (ATR)

This parameter defines the distance from the entry price to the initial stop, in terms of ATR. Since ATR is a measure of daily volatility and the Turtle System stops are based on ATR, this means that the Turtle System equalizes the position size across the various markets based on volatility.

According to the original Turtle Rules, long positions were stopped out if price fell 2 ATR from the entry price. Conversely, short positions were stopped out if the price rose 2 ATR from the entry price.

Unlike the *Exit Breakout* based stop, which moves up or down with the X-day high or low, the stop defined by *Stop in ATR* is a "hard" stop that is fixed above or below the entry price upon entry. Once set, it does not vary throughout the course of the trade, unless Units are added, in which case the for earlier units are raised by the amount specified by *Unit Add (ATR)*.

Trades are liquidated when price hits the stop defined by either the *Stop in ATR*, the *Entry Breakout* for the opposite direction, or the *Exit Breakout* (see above), whichever is closest to the price at the time.

In this system the initial entry stop for the trade entry day is based on the order price. This is for ease of placing the stop once the order is filled. Note that the stop is adjusted based on the actual fill price for the following day.

Exit Breakout (days)

Trades in progress are exited when the price hits the high or the low of the preceding X-days as adjusted by the *Exit Offset*. This concept is the identical to Entry Breakout, but the logic is reversed: Long trades are exited when price breaks out below the X-day low, and short trades are exited when price breaks out above the X-day low.

The Exit Breakout moves up (or down) with price. It protects against adverse price excursions, and also serves as a trailing stop that acts to lock in a profit when the trend reverses.

Trades are liquidated when price hits the stop defined by either the *Stop in ATR*, the *Entry Breakout* for the opposite direction, or the *Exit Breakout* (see above), whichever is closest to the price at the time.

Exit Offset (ATR)
If set to zero, this parameter has no effect. If Exit Offset in ATR is set to 1.0, a long position isn't exited until price hits the normal breakout price, minus 1.0 ATR. Likewise, a short position won't be exited until the price hits the normal breakout price, plus 1.0 ATR. Either a positive or negative value can be specified for this parameter. A positive value effectively delays exit until the specified point after the breakout threshold chosen; a negative value would exit before the breakout threshold chosen.

Max Instrument Units
This parameter defines the maximum number of Units that can be held at one time, in any single futures market, or any single stock. For instance, *Max Instrument Units* = 4 means that no more than 4 Units of Coffee may be held at one time; this includes the initial Unit, plus 3 Units added.

Entry Orders:

```
VARIABLES: longEntryPrice, shortEntryPrice, unitIncrementATR, unitEntryPrice TYPE: Price
VARIABLES: entryOffsetATR TYPE: Price
VARIABLES: unitsOn, unitIndex TYPE: Integer
VARIABLES: brokerPrice TYPE: Price

' Calculate the unit increment, stop amount, and entry offset
unitIncrementATR = unitAdd * averageTrueRange
stopWidth = stopInATR * averageTrueRange

'Total units currently on for this instrument
unitsOn = instrument.totalUnits

'-----------------------------------------------------------------
'Place initial entry orders when out of the market
'-----------------------------------------------------------------

' If we are not long (we are short or out) then place orders for tomorrow
' We will place orders for all potential units in case it's a big day
IF instrument.position <> LONG THEN

    ' Determine whether to use the regular entry high or the failsafe
    longEntryPrice = offsetAdjustedEntryHigh
    IF ( NOT tradeIfWinner ) AND ( lastTradeProfit > 0 ) THEN longEntryPrice = offsetAdjuste

    ' Loop over the potential units and place an order
    FOR unitIndex = 1 TO maxUnits
        brokerPrice = longEntryPrice + ( ( unitIndex - 1 ) * unitIncrementATR )
        broker.EnterLongOnStop( brokerPrice, brokerPrice - stopWidth )
    NEXT

ENDIF

' If we are not short (we are long or out) then place orders for tomorrow
' We will place orders for all potential units in case it's a big day
IF instrument.position <> SHORT THEN

    ' Determine whether to use the regular entry low or the failsafe
    shortEntryPrice = offsetAdjustedEntryLow
```

```
        IF ( NOT tradeIfWinner ) AND ( lastTradeProfit > 0 ) THEN shortEntryPrice = offsetAdjust

        ' Loop over the potential units and place an order
        FOR unitIndex = 1 to maxUnits
            brokerPrice = shortEntryPrice - ( ( unitIndex - 1 ) * unitIncrementATR )
            broker.EnterShortOnStop( brokerPrice, brokerPrice + stopWidth )
        NEXT

    ENDIF

    '-----------------------------------------------------------------
    'Add additional Units when already in the market
    '-----------------------------------------------------------------

    'If long and can add more units
    IF ( instrument.position = LONG ) AND ( unitsOn < maxUnits ) THEN

        ' Loop over potential units to add, placing orders at maximum of regular entryHigh or un
        FOR unitIndex = unitsOn + 1 TO maxUnits
            unitEntryPrice = instrument.unitEntryFill[ unitsOn ] + ( unitIndex - unitsOn ) * uni
            brokerPrice = max( offsetAdjustedEntryHigh, unitEntryPrice )
            broker.EnterLongOnStop( brokerPrice, brokerPrice - stopWidth)
        NEXT
    ENDIF

    'If short and can add more units
    IF ( instrument.position = SHORT ) AND ( unitsOn < maxUnits ) THEN

        ' Loop over potential units to add, placing orders at minimum of regular entryLow or uni
        FOR unitIndex = unitsOn + 1 TO maxUnits
            unitEntryPrice = instrument.unitEntryFill[ unitsOn ] - ( unitIndex - unitsOn ) * uni
            brokerPrice = min( offsetAdjustedEntryLow, unitEntryPrice )
            broker.EnterShortOnStop( brokerPrice, brokerPrice + stopWidth )
        NEXT
    ENDIF
```

Entry Order Filled:

```
    VARIABLES: stopAmount, stopPrice, originalATR TYPE: Price
    VARIABLES: unitFillPrice, newStopPrice TYPE:Price
    VARIABLES: newUnitExitStop, originalExitStop, newExitStop TYPE: Price
    VARIABLES: entryDayIndex TYPE: Integer
    VARIABLES: unitsOn, unitIndex TYPE: Integer

    '---------------------------
    ' Get the total number of units on now, including this new unit
    unitsOn = instrument.totalUnits

    ' Calculate the new exit stop for this new unit, based on the fill price
    if instrument.position = LONG then newUnitExitStop = order.fillPrice - order.entryRisk
    if instrument.position = SHORT then newUnitExitStop = order.fillPrice + order.entryRisk

    ' Set exit stop for this new unit added
    instrument.setExitStop( unitsOn, newUnitExitStop )


    '-----------------------------
    ' If more than one unit on, then
    ' Reset the stops for all units, except the current one, based on this new unit
    ' For each unit, look for the original ATR and calculate the original stop price
    ' Add to the original stop
```

```
'-----------------------------

IF unitsOn > 1 THEN
    IF instrument.position = LONG THEN
        FOR unitIndex = 1 TO unitsOn - 1

            ' Compute the ATR TYPE: of the entry for this unit.
            entryDayIndex = instrument.bar - instrument.unitEntryDayIndex[ unitIndex ]
            originalATR = averageTrueRange[ entryDayIndex ]

            ' Get the fill price.
            unitFillPrice = instrument.unitentryfill[ unitIndex ]

            ' Compute the original exit stop.
            originalExitStop = unitFillPrice - stopInATR * originalATR

            ' Now compute the new stop based on the original ATR.
            newStopPrice = originalExitStop + ( unitsOn - unitIndex ) * unitAdd * originalAT

            ' Constrain the stop to the fill price or the new unit stop.
            newExitStop = min( newStopPrice, unitFillPrice, newUnitExitStop )

            ' Give the instrument this new stop.
            instrument.setExitStop( unitIndex, newExitStop )
        NEXT
    ENDIF

    IF instrument.position = SHORT THEN
        FOR unitIndex = 1 TO unitsOn - 1

            ' Compute the ATR TYPE: of the entry for this unit.
            entryDayIndex = instrument.bar - instrument.unitEntryDayIndex[ unitIndex ]
            originalATR = averageTrueRange[ entryDayIndex ]

            ' Get the fill price.
            unitFillPrice = instrument.unitentryfill[ unitIndex ]

            ' Compute the original exit stop.
            originalExitStop = unitFillPrice + stopInATR * originalATR

            ' Now compute the new stop based on the original ATR.
            newStopPrice = originalExitStop - ( unitsOn - unitIndex ) * unitAdd * originalAT

            ' Constrain the stop to the fill price or the new unit stop.
            newExitStop = max( newStopPrice, unitFillPrice, newUnitExitStop )

            ' Give the instrument this new stop.
            instrument.setExitStop( unitIndex, newExitStop )
        NEXT
    ENDIF
ENDIF


' ---------------------------------
' Track theoretical positions
' ---------------------------------

' Only run these calcs if we have to
IF ( tradeIfWinner = FALSE ) THEN

    ' If this is the first unit on, then set the theoretical position at this point
    IF unitsOn = 1 THEN
```

```
            theoreticalPosition = instrument.position
            theoreticalEntryPrice = order.fillprice
            theoreticalExitStop = order.stopprice
        ENDIF
    ENDIF
```

Exit Orders;

```
    VARIABLES: brokerPrice, exitPrice, unitCurrentStopPrice TYPE: Price
    VARIABLES: unitsOn, unitIndex TYPE: Integer

    unitsOn = instrument.totalUnits

    ' If long...
    IF ( instrument.position = LONG ) THEN

        ' Get the best exit price based on exit low and entry low.
        exitPrice = max( offsetAdjustedExitLow, offsetAdjustedEntryLow )

        ' Loop over the units and place stop order at the best price,
        ' including current stop price.
        FOR unitIndex = 1 TO unitsOn
            unitCurrentStopPrice = instrument.unitExitStop[ unitIndex ]
            brokerPrice = max( exitPrice, unitCurrentStopPrice )
            broker.ExitUnitOnStop( unitIndex, brokerPrice )
        NEXT
    ENDIF

    ' If short...
    IF ( instrument.position = SHORT ) THEN

        ' Get the best exit price based on exit high and entry high.
        exitPrice = min( offsetAdjustedExitHigh, offsetAdjustedEntryHigh )

        'Loop over units and place stop order at best price,
        ' including curreen stop price.
        FOR unitIndex = 1 TO unitsOn
            unitCurrentStopPrice = instrument.unitExitStop[ unitIndex ]
            brokerPrice = min( exitPrice, unitCurrentStopPrice )
            broker.ExitUnitOnStop( unitIndex, brokerPrice )
        NEXT
    ENDIF
```

Adjust Stops:

```
    VARIABLES: stopPrice, bestExitPrice, unitCurrentStopPrice TYPE: Price
    VARIABLES: unitsOn, unitIndex TYPE: Integer

    ' Get the total units on.
    unitsOn = instrument.totalUnits

    ' If long...
    IF ( instrument.position = LONG ) THEN

        ' Get the best exit price based on exit low and entry low.
        bestExitPrice = max( offsetAdjustedExitLow, offsetAdjustedEntryLow )

        ' Loop over the units and set the stop to the best price,
        ' including current stop price.
        FOR unitIndex = 1 TO unitsOn

            ' Get the current exit stop.
            unitCurrentStopPrice = instrument.unitExitStop[ unitIndex ]

            ' If the best stop is better than our current stop then
            ' exit on this stop.
            IF bestExitPrice > unitCurrentStopPrice THEN
                instrument.SetExitStop( unitIndex, bestExitPrice )
                broker.exitUnitOnStop( unitIndex, bestExitPrice )
            ENDIF
        NEXT
    ENDIF

    ' If short...
    IF ( instrument.position = SHORT ) THEN

        ' Get the best exit price based on exit high and entry high.
        bestExitPrice = min( offsetAdjustedExitHigh, offsetAdjustedEntryHigh )

        ' Loop over the units and set the stop to the best price,
        ' including curreen stop price.
        FOR unitIndex = 1 TO unitsOn

            ' Get the current exit stop.
            unitCurrentStopPrice = instrument.unitExitStop[ unitIndex ]

            ' If the best stop is better than our current stop then
            ' exit on this stop.
            IF bestExitPrice < unitCurrentStopPrice THEN
                instrument.SetExitStop( unitIndex, bestExitPrice )
                broker.exitUnitOnStop( unitIndex, bestExitPrice )
            ENDIF
        NEXT
    ENDIF
```

After Instrument Day:

```
    VARIABLES: theoreticalExitPrice TYPE: PRICE

    ' Calc stop width
    stopWidth = stopInATR * averageTrueRange

    ' ------------------------------
    ' Track theoretical positions
    ' ----------------------------

    IF ( tradeIfWinner = FALSE ) THEN

        ' Update theoretical profit mark to market
        IF ( theoreticalPosition = LONG ) THEN
                lastTradeProfit = ( instrument.close - theoreticalEntryPrice )
        ENDIF

        if ( theoreticalPosition = SHORT ) THEN
                lastTradeProfit = ( theoreticalEntryPrice - instrument.close )
        ENDIF



        ' If not long, then watch for long entry
        IF ( theoreticalPosition <> LONG ) THEN
            IF ( instrument.high > offsetAdjustedEntryHigh[1] ) THEN
                theoreticalPosition = LONG
                theoreticalEntryPrice = offsetAdjustedEntryHigh[1]
                theoreticalExitStop = theoreticalEntryPrice - stopWidth
            ENDIF
        ENDIF


        ' If not short, then watch for short entry
        IF ( theoreticalPosition <> SHORT ) THEN
            IF instrument.low < offsetAdjustedEntryLow[1] THEN
                theoreticalPosition = SHORT
                theoreticalEntryPrice = offsetAdjustedEntryLow[1]
                theoreticalExitStop = theoreticalEntryPrice + stopWidth
            ENDIF
        ENDIF


        ' If long, then watch for exits
        IF ( theoreticalPosition = LONG ) THEN
            theoreticalExitPrice = max( theoreticalExitStop, offsetAdjustedExitLow, offsetAdjust
            IF ( instrument.low < theoreticalExitPrice ) THEN
                lastTradeProfit = ( theoreticalExitPrice - theoreticalEntryPrice )
                theoreticalPosition = OUT
            ENDIF
        ENDIF


        ' If short, then watch for exits
        IF ( theoreticalPosition = SHORT ) THEN
            theoreticalExitPrice = min( theoreticalExitStop, offsetAdjustedExitHigh, offsetAdjus
            if ( instrument.high > theoreticalExitPrice ) THEN
                lastTradeProfit = ( theoreticalEntryPrice - theoreticalExitPrice )
                theoreticalPosition = OUT
            ENDIF
        ENDIF

    ENDIF
```

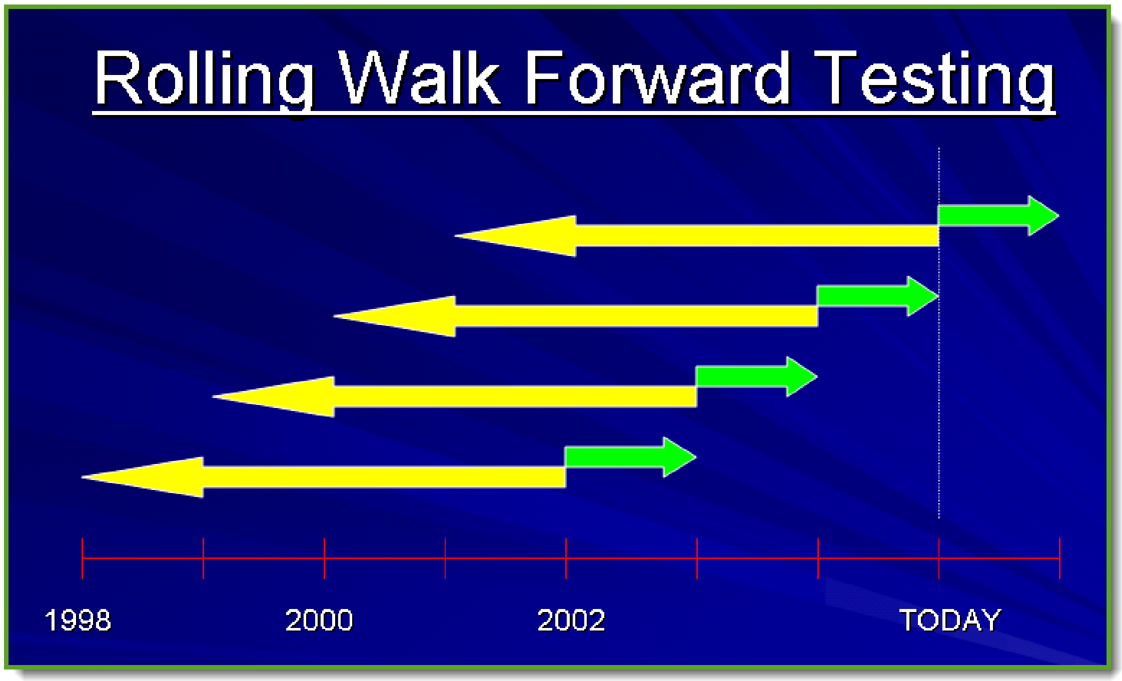
Section 17 – Walk Forward

Walk Forward testing can be performed with any suite name that has a system attached. In other words the suite name doesn't need to be "Walk Forward" in order to use the Out of Sample (OOS) walk forward optimization procedure.

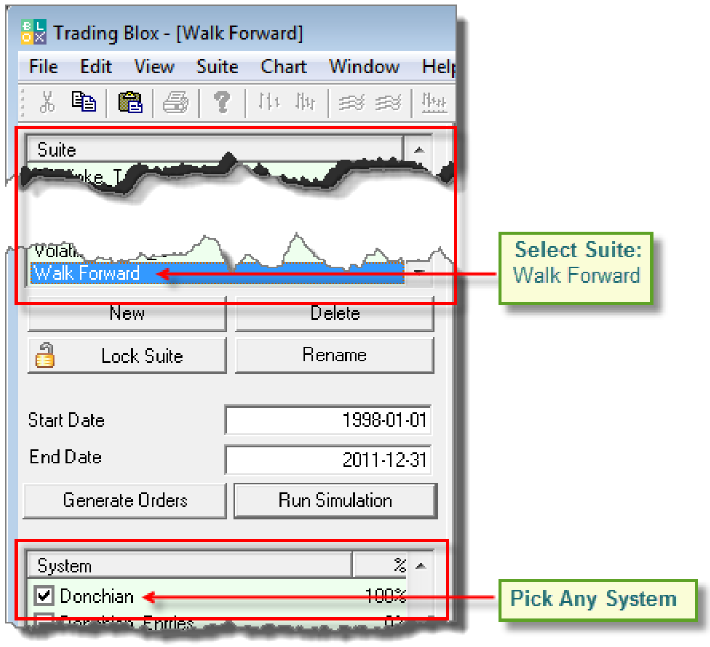
Walk Forward testing is a concept that uses a segment of historical data for system parameter optimization and then uses those optimized set of parameters selected using the maximum goodness statistic assigned in **Trading Blox Preferences -> Data Folders and Options -> Multi-Parameter Graph** section. Using the statistic index number assigned to the Measure of Goodness Index as the best performance guide, Walk Forward feature will then test an out of sample segment of data to generate a test result.

In the chart shown next the bottom yellow arrow represents an optimization test from 1998 to 2002. When the system is set to step parameters, the test could be hundreds of stepped tests over this time frame to find the optimal parameters for the system. When the optimization ends the optimal system parameters are used in an out of sample test for 2002 to 2003.

because there is more data available the process will begin again with an optimization run from 1999 to 2003, and an out of sample test for 2003 to 2004.



Using Walk Forward Optimizations:
Select the Walk Forward Suite, and then select any working system name from the system listing you want to test. If there isn't a listing for the Walk Forward, then create it. If the Walk Forward Suite doesn't exist, create it. Also create a system in the System Editor using the same name after you create the new suite. When you create the Walk Forward system you will be notified there is a suite by the same name and the system will be considered a Global suite.



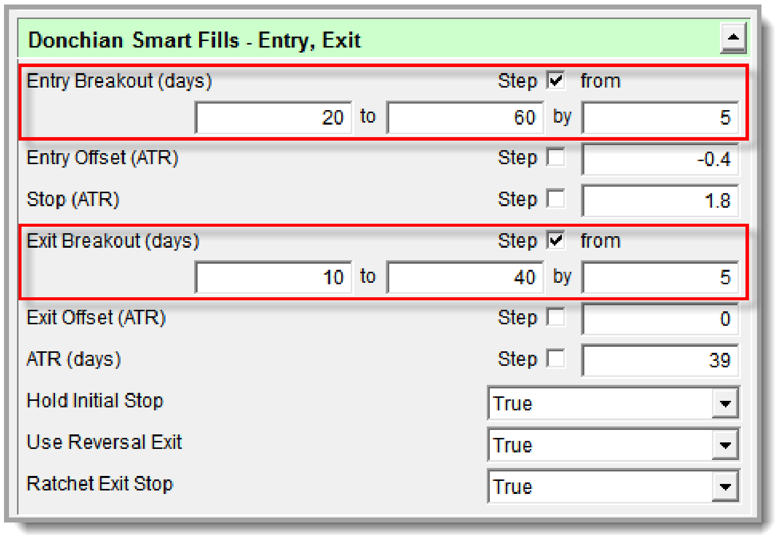
If it does exists and another system has been assigned, remove that system if it isn't the system you wish to optimize.

Go into the **System Editor** and check if the **Walk Forward Stats 2** blox is attached to the **Walk Forward** system. This blox module will change how a Walk Forward simulation reports its results.

Note:

Do not allow the **Walk Forward Stats 2** blox to be attached to a standard simulation system as the error checking internal to this blox will cause Trading Blox to halt with error informing the blox is attached in the wrong place.

With the system selected its parameter section will appear and you can set the important system parameters to Step as necessary. In this example the **Donchian SmartFills - EntryExit** Blox has been selected for our optimization test.

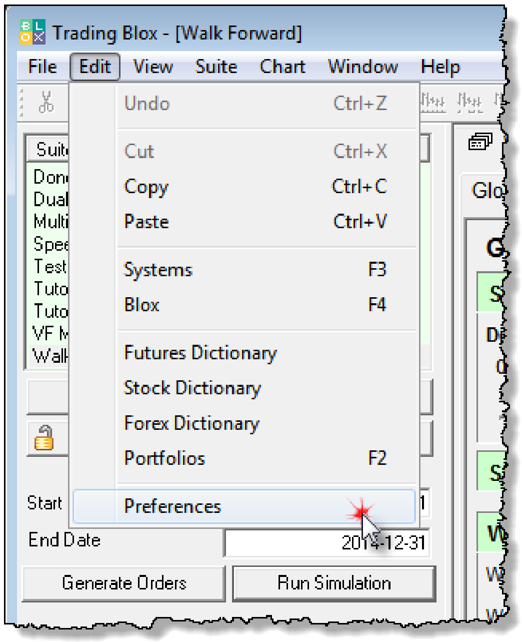


Donchian Smart Fills - Entry, Exit

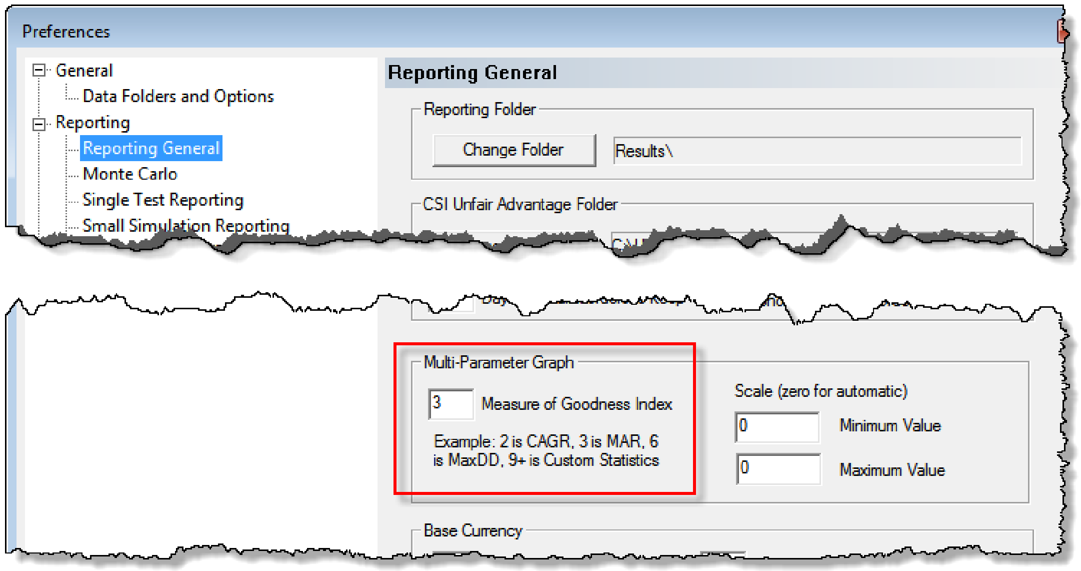
Entry Breakout (days)				Step [✓]	from	
	20	to	60	by		5
Entry Offset (ATR)				Step [ ]		-0.4
Stop (ATR)				Step [ ]		1.8
Exit Breakout (days)				Step [✓]	from	
	10	to	40	by		5
Exit Offset (ATR)				Step [ ]		0
ATR (days)				Step [ ]		39
Hold Initial Stop						True
Use Reversal Exit						True
Ratchet Exit Stop						True

Enable stepping for the parameters you want to optimize.

Checking Preferences:
The optimal run from the optimization runs is determined by the maximum goodness value. Goodness value is a statistic set in Preferences , and can be any of the built in statistics, or custom statistics. Access to Trading Blox Preference settings is available under the main screen's Edit menu option:



When Preferences settings appear, click on **Data Folders and Options** so you see this information:



Results\

3 Measure of Goodness Index

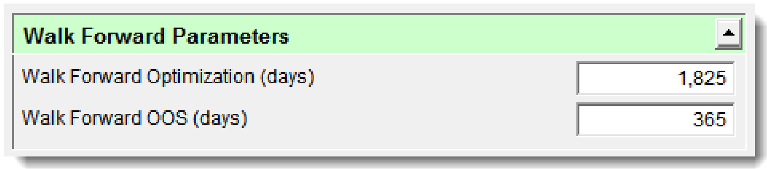
Example: 2 is CAGR, 3 is MAR, 6 is MaxDD, 9+ is Custom Statistics

0 Minimum Value

0 Maximum Value

Enter a statistic value if you need to change it. Then click **OK**.

Setting Global Parameters:
In the Global Parameters section these Walk Forward Parameters are available:



Walk Forward Parameters	
Walk Forward Optimization (days)	1,825
Walk Forward OOS (days)	365

- **Walk Forward Optimization (days)**, determines how many calendar days over which to optimize the parameters.

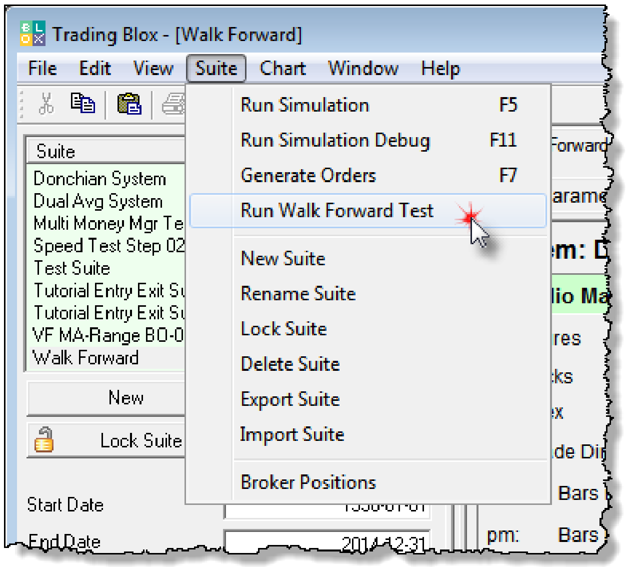
- **Walk Forward OOS (days)**, determines the **OOS** (Out of Sample) calendar days over which to use to test the optimized parameters.

Check the other Global Parameters listed to ensure they are they represent the values you want applied to this optimization simulation. Check the Thread Count parameter field to ensure it is a value greater than 1. Threads will allow the optimization to complete faster.

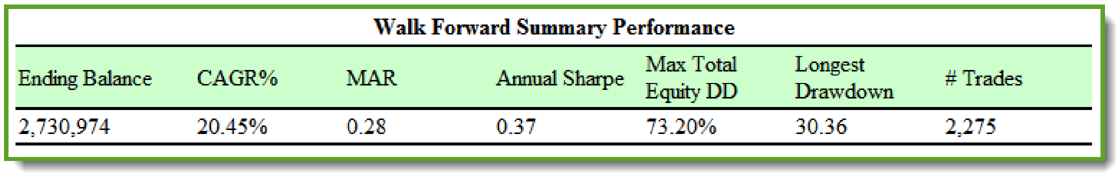
Note:

Saving of open positions from one OOS test to the next has limitation because any variables that might be tracking trailing stops and such are not brought over.

Creating a Walk Forward Simulation:
Go to the Suite menu and select the option **Run Walk Forward Test**:



Walk Forward Performance Results:
This table shows Summary Performance results from the Walk Forward Simulation:



Walk Forward Summary Performance

Ending Balance	CAGR%	MAR	Annual Sharpe	Max Total Equity DD	Longest Drawdown	# Trades
2,730,974	20.45%	0.28	0.37	73.20%	30.36	2,275

When the above report appears it also shows the period results of each of the periods shown in the stepped value performance table near the top of the screen.

Section 18 – Stats for Multi Parameter Charts

This block sets the statistics to use for the multi parameter charts. Enter the names of the statistics, either built in or custom, separated by commas or spaces.

Here is the code;

```
' Sets the goodness stats to use for multi parameter charts.
' These are the built-ins. Custom stats can be used as well.
' "End Balance, CAGR%, MAR, Sharpe, Ann. Sharpe, Max TE DD, Longest DD, Trades"
test.SetGoodnessToChart( goodnessString )
```

Part XII

Part 12 – Generating Orders

To generate orders, select the Generate Orders command from the Suite menu. Trading Blox will run a test and generate broker orders for the next bar. The next bar is the bar after the last bar -- for intraday data this would be the next hour or minute, and for daily data this would be the next day. The order date and time will be noted on the order file. The results will be stored in the Orders folder as well as displayed.

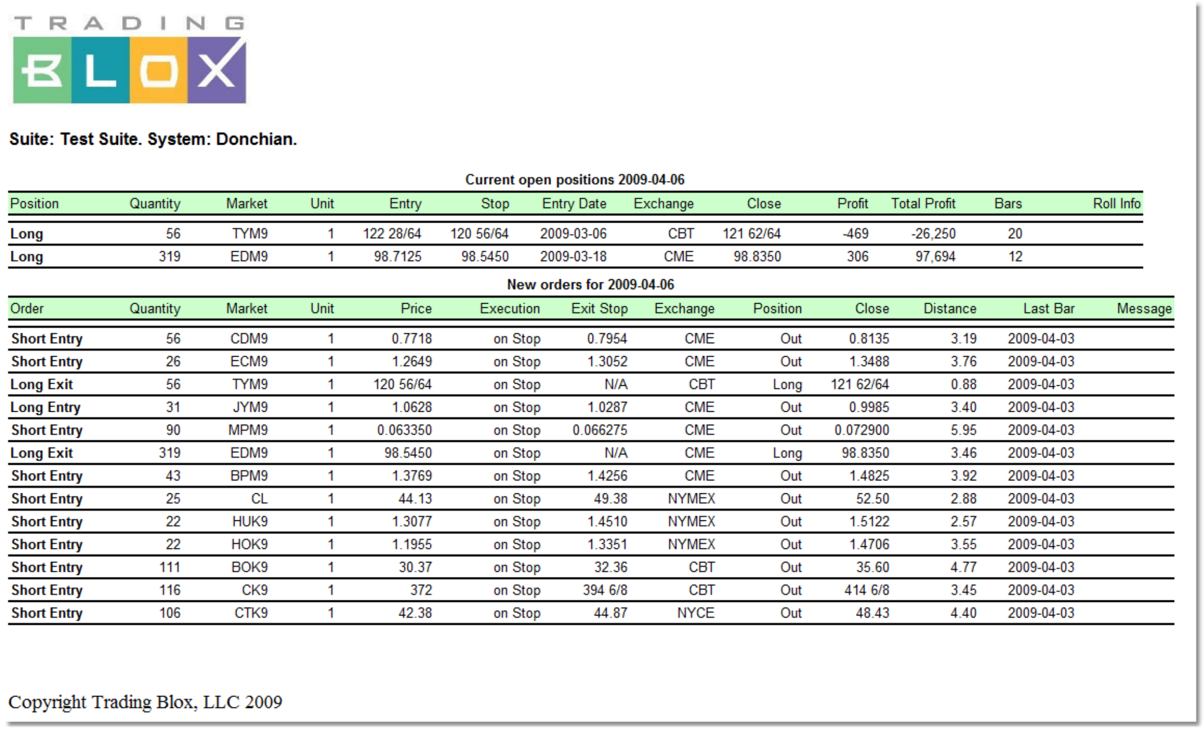
If Order Generation Equity is set to zero (default), the amount of equity available for order generation will be determined by the ending equity of a test run for the period Start Date to End Date. If the Order Generation Equity is non zero, the value you entered will be used for order generation equity. The test start and end date are used to run a test just prior to Order Generation. This pre test is used to get theoretical positions and equity. Typically the start date is set to the date actual trading has started, so starting equity matches the start date. This start date is left constant as orders are generated in the futures. The end date can be any date in the future, often set a year ahead so it does not have to be set every time order generation is run.

The filter for volume, margin, and Risk Manager (Can Add Unit script) are run as orders are generated. So orders generated will be pre filtered for these factors. Note that scripts like After Instrument Open, Can Fill Order, and After Instrument Day, will not be run for the order generation bar. The instrument.tradeDayOpen is also not available for the order generation bar.

The name of the file will be: Orders <Timestamp> .mht. The Timestamp has the format YYYY-MM-DD_HH_MM_SS.

The order file will look like the following. It shows the open positions at the top, and the orders for tomorrow at the bottom.
For each order, the Current Position is listed as well: it shows either LONG, SHORT, or OUT; where OUT represents no current open position in the market.

An additional file, "Orders.csv" is created (and overwritten) for every order run. You can open this file in excel, and filter your orders as necessary.


Suite: Test Suite. System: Donchian.

Current open positions 2009-04-06

Position	Quantity	Market	Unit	Entry	Stop	Entry Date	Exchange	Close	Profit	Total Profit	Bars	Roll Info
Long	56	TYM9	1	122 28/64	120 56/64	2009-03-06	CBT	121 62/64	-469	-26,250	20	
Long	319	EDM9	1	98.7125	98.5450	2009-03-18	CME	98.8350	306	97,694	12	

New orders for 2009-04-06

Order	Quantity	Market	Unit	Price	Execution	Exit Stop	Exchange	Position	Close	Distance	Last Bar	Message
Short Entry	56	CDM9	1	0.7718	on Stop	0.7954	CME	Out	0.8135	3.19	2009-04-03	
Short Entry	26	ECM9	1	1.2649	on Stop	1.3052	CME	Out	1.3488	3.76	2009-04-03	
Long Exit	56	TYM9	1	120 56/64	on Stop	N/A	CBT	Long	121 62/64	0.88	2009-04-03	
Long Entry	31	JYM9	1	1.0628	on Stop	1.0287	CME	Out	0.9985	3.40	2009-04-03	
Short Entry	90	MPM9	1	0.063350	on Stop	0.066275	CME	Out	0.072900	5.95	2009-04-03	
Long Exit	319	EDM9	1	98.5450	on Stop	N/A	CME	Long	98.8350	3.46	2009-04-03	
Short Entry	43	BPM9	1	1.3769	on Stop	1.4256	CME	Out	1.4825	3.92	2009-04-03	
Short Entry	25	CL	1	44.13	on Stop	49.38	NYMEX	Out	52.50	2.88	2009-04-03	
Short Entry	22	HUK9	1	1.3077	on Stop	1.4510	NYMEX	Out	1.5122	2.57	2009-04-03	
Short Entry	22	HOK9	1	1.1955	on Stop	1.3351	NYMEX	Out	1.4706	3.55	2009-04-03	
Short Entry	111	BOK9	1	30.37	on Stop	32.36	CBT	Out	35.60	4.77	2009-04-03	
Short Entry	116	CK9	1	372	on Stop	394 6/8	CBT	Out	414 6/8	3.45	2009-04-03	
Short Entry	106	CTK9	1	42.38	on Stop	44.87	NYCE	Out	48.43	4.40	2009-04-03	

Copyright Trading Blox, LLC 2009

Delivery Month

If the data source includes the Delivery Month, the month of the positions will also be listed. The delivery month of the orders generated will also be listed. If they do not match, ie if you have a position in a back month while the data has rolled to the new front month, an * will appear next to the order price to highlight this roll. A note will be printed in the positions section that you should roll this position from the old month to the new month. The roll alert message and * will only show for 1 day. Thereafter it is assumed that the position has rolled from the old position to the new position.

All prices are in current month prices, including the entry price of the position, and all the new order prices. Trading Blox assumes the position will be rolled on the open, because the data has already rolled. It is important to do this in that you want to 'trade what you test.' So if your historical testing assumed a certain roll algorithm, and your data is setup for such, you should roll your positions at the same time.

Distance

Distance is the number of ATR from the last Close price to the Order price. This gives you some idea of how much the market would need to move to fill this order, so you can place the orders into the market as necessary.
The ATR used is the Wilder 20-day ATR (39 day Exponential Moving Average of the True Range).

Last Bar

The Last Bar is the date of the last bar of data for the market. If this date is not the current date, or as expected due to holidays, then you should double check your data. It's possible the data for a particular market was not updated on schedule. In this case the Order Report is for April 6 (Monday) and all the markets have a Last Bar of April 3 (Friday).

Message

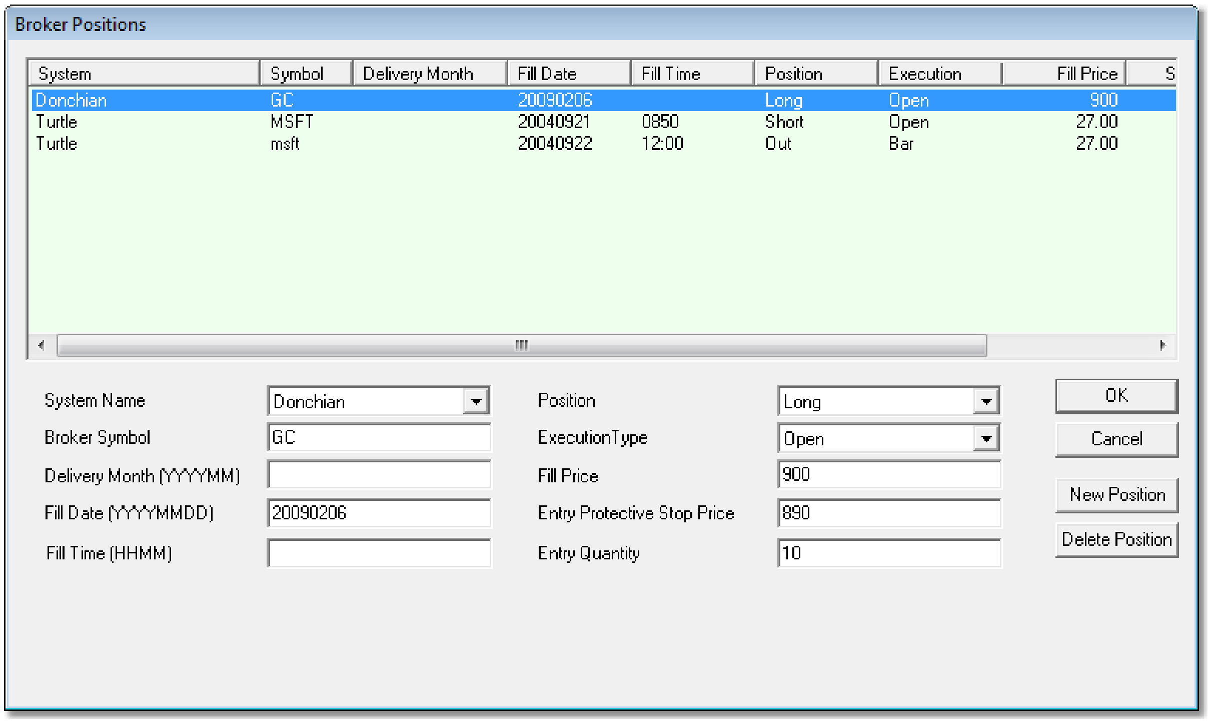
The Message is a custom value set in Blox Scripting. This can be any message (string value) as needed by the system.

Section 1 – Broker Positions

Very often, you will not get the exact fill that a historical simulation might assume when trading an actual account. This is to be expected. Sometimes you forget to roll a contract when the underlying data source rolls to a new month. Other times, your broker may fill an incorrect order or you may enter an order incorrectly.

To enable these positions and have them insert into the test, use the User Broker Positions global parameter.

The Broker Positions screen lets you tell Trading Blox what your actual positions are so that Trading Blox can generate orders using those existing positions rather than using positions based on historical testing assumptions. Although it should be noted that a preferred approach for many traders is to match their actual trading to the simulation. So in other words, let the simulation create the positions and make sure your broker positions match the system. We do not recommend using this Broker Positions feature because it adds more complexity than it is generally worth, but the option is here for you.



System	Symbol	Delivery Month	Fill Date	Fill Time	Position	Execution	Fill Price	S
Donchian	GC		20090206		Long	Open	900	
Turtle	MSFT		20040921	0850	Short	Open	27.00	
Turtle	msft		20040922	12:00	Out	Bar	27.00	

System Name	Donchian	Position	Long
Broker Symbol	GC	ExecutionType	Open
Delivery Month (YYYYMM)		Fill Price	900
Fill Date (YYYYMMDD)	20090206	Entry Protective Stop Price	890
Fill Time (HHMM)		Entry Quantity	10

Enter your actual positions, fills, and quantities that you have with your broker. For stocks be sure to enter the actual fill prices from your broker statement, not the stock split adjusted prices, for historical trades. Stock prices will be adjusted by the stock split ratio to match the stock split adjusted data. Likewise the quantity will be adjusted on the stock split date to represent the new adjusted quantity.

These positions get inserted into the test on the Order Date specified, and any existing positions will be exited as a result. You can enter a quantity of zero to lock out any further positions in this instrument. You can enter a position of OUT to exit any existing positions on the Order date.

NOTE: Positions entered here are locked -- so subsequent exits in the system code will not cause an exit of these positions. Entering a subsequent position of "OUT" will exit the position and leave it locked so no further entries will be filled by the simulation. Enter a position of "Exit" to exit the position and unlock the instrument so that further entries and exits by the simulation can occur. The order generation report will still have new entries and exits as needed for this positions. The order generation is not locked, just the simulation fill process.

Position Elements
Each position entry consists of the following items:

- System Name - must match the system name exactly or the trade will be ignored. Drop down box populates from selected systems for the current suite.

- Symbol - must match exactly.

- Delivery month - optional. Used only to suppress the alert message once you have rolled from the old month to the new month. Available only if the data also has a delivery month.

- Fill Date - must be a real trading day within the test range for that instrument or it will be ignored.
  YYYY-MM-DD.
  YYYYMMDD
  YYMMDD
  MM/DD/YY
  MM/DD/YYYY

- Fill Time -- must be an available time in the data file. For daily data this can be ignored -- only used for intraday data.
  HHMM format

- Position - must be Long, Short, Out, or Exit. Drop down box has these options. Out exits the position and leaves the instrument locked, Exit exits the positions and leaves the instrument unlocked.

- Execution Type -- If set to Open, then this position will be 'filled' before any simulated positions for the bar. If set to Bar, then the simulation open positions for the bar will be filled first, then this position. Likewise for the Close execution type, which would be filled last.

- Fill Price - required for entries and exits. It is the fill price for the order. No commas.

- Entry Protective Stop Price - optional. If left blank the system will assume no stop was used. This is the original entry day protective stop. The stop your system uses on a daily basis will likely be adjusted from this as time goes by.

- Entry Quantity

Futures: Rolling from one month to the next
Enter only your current actual positions. So when you roll, keep the entry date and symbol the same, but change the delivery month and adjust the entry price and stop price by the spread amount. Since these prices are relative to the current month, you need to adjust them accordingly.

Old position: GC, delivery month 200704, entry date 20070305, entry price 640, initial stop 635, position long, quantity 10.
GCK07 is at 655 and GCJ07 is at 653 on the close the day of the roll, so the spread is +$2.
New position: GC, delivery month 200705, entry date 20070305, entry price 642, initial stop 637, position long, quantity 10.

The position entry delivery month is only presented as long as the data is still priced in that month plus 5 days. For 5 days after the roll, a roll alert is presented indicating that the position should be rolled from the old month to the new month. Then after the 5 days the system assumes you have rolled and the entry delivery month is changed to the current data delivery month.

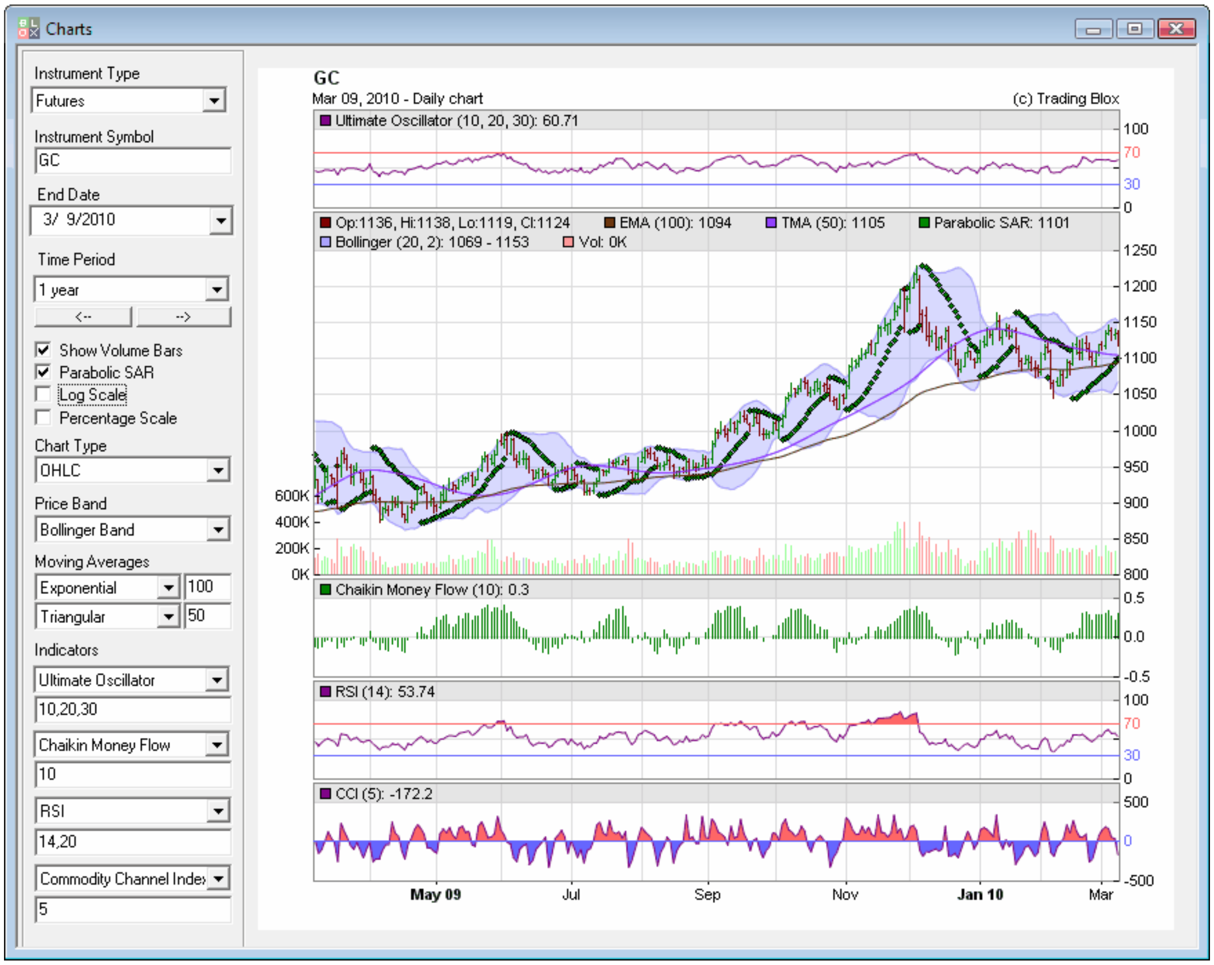
The broker position delivery month is only used if it represents the new delivery month as used by the data. In this way you can suppress the roll alert once you have rolled from the old month to the new month.

Quick Charts

Part XIII

Part 13 – Quick Charts

Use the View / Quick Charts menu item to access this feature. Available only in the Builder Edition.


Charts

Instrument Type: Futures
Instrument Symbol: GC
End Date: 3/ 9/2010
Time Period: 1 year
<-- -->
[✓] Show Volume Bars
[✓] Parabolic SAR
[ ] Log Scale
[ ] Percentage Scale
Chart Type: OHLC
Price Band: Bollinger Band
Moving Averages: Exponential 100; Triangular 50
Indicators: Ultimate Oscillator 10,20,30; Chaikin Money Flow 10; RSI 14,20; Commodity Channel Index 5

GC
Mar 09, 2010 - Daily chart
(c) Trading Blox
Ultimate Oscillator (10, 20, 30): 60.71
Op:1136, Hi:1138, Lo:1119, Cl:1124
EMA (100): 1094
TMA (50): 1105
Parabolic SAR: 1101
Bollinger (20, 2): 1069 - 1153
Vol: 0K
Chaikin Money Flow (10): 0.3
RSI (14): 53.74
CCI (5): -172.2
May 09
Jul
Sep
Nov
Jan 10
Mar

Instrument Type
Stock/Futures/Forex

Instrument Symbol
The symbol as entered into the Dictionary

End Date
Defaults to today, but can be set to any date using the calendar picker

Time Period
Set to the time period for the chart. The back and forward arrows move that amount forward and back in time, to scroll the chart.

Chart Options
Show Volume Bars
Parabolic SAR
Log Scale
Percentage Scale

Chart Type
Candlestick
Closing Price

Median Price
OHLC Bars
Typical Price
Weighted Close

Price Band
Bollinger Band
Donchian Band
Envelope

Moving Averages
Up to two moving averages can be plotted on the chart. Select the type, and enter the bars to use to the right.
Simple
Exponential
Triangular
Weighted

Indicators
Up to four indicators can be plotted with the chart. The first one will be on top. Use the space below to enter the parameters such as 14 or 14,20 or 14,20,10 depending on how many parameters the indicator needs.

Note that these indicators are all computed by the third party charting package and are for visual reference only. They are not necessarily computed the same way as the indicators in the Trading Blox Systems.

Appendices

Part XIV

Part 14 – Appendices

Section 1 – Glossary of Terms

This is not a complete glossary because *some knowledge of trading concepts is assumed*. This Glossary contains the most important definitions used in Trading Blox.

Average True Range (ATR)
The Average True Range is a measure of volatility which indicates the average range of price movement in a single day. This measure was developed by J. Welles Wilder and introduced in his book, *New Concepts in Technical Trading Systems* (1978).

The average of the *True Range* where True Range is the range including opening gaps, or the greatest of:

- The current high minus the current low
- The current high minus the previous close
- The previous close minus the current low

The last two entries account for price gaps where the entire day's prices do not overlap the previous day's close.

Trading Blox allows you to set the number of days to use when creating an Average True Range Indicator, and we use the *Exponential Moving Average* to calculate this indicator. This is a different formula than the *Wilder Moving Average Formula*. As an example, the 20 day Wilder ATR is the equivalent to a 39 day Exponential ATR.

```
Exponential MA Days = (Wilder MA Days X 2) - 1
```

Autocorrelation
A term referring to the correlation of a time series with a lagged version of itself. Sometimes referred to as serial correlation or lagged correlation.

Breakout
A chart pattern used to indicate a rise in a stock's price above its resistance level or drop below its support level. It is typically measured in terms of days. An N-Day breakout is a breakout of the highest high of the last N Days or the lowest low of the last N Days, where N is the number of days for the breakout. Thus a 20-Day breakout of the highs would occur when the price exceeded the highest high of the last 20 days.

Breakouts are used in trend-following systems to indicate a potential start of a trend.

Bollinger Band
A Bollinger Band is a pricing channel which consists of a top and bottom price which are defined in terms of a *Simple Moving Average* and the price a fixed number of *Standard Deviations* above that moving average, and below that moving average.

Closed Equity
Closed Equity is a measure of equity that take the starting balance and adjusts for the profits of *closed out trades only*. It does not take into account the value of open positions.

Confidence Level
A Confidence Level is used during Monte Carlo Simulations to indicate that a certain percentage of the simulated equity curves demonstrated results for a particular measure that were greater than the confidence level. For example, a 95% Confidence Level MAR Ratio of 0.8 indicates that 95% of the simulation iterations showed a MAR Ratio of 0.8 or better.

Core Equity
Core Equity is a measure of equity that includes *Closed Equity* plus that portion of *Open Equity* that would be realized if all the current positions were exited at their current stop values (i.e. *Locked-In Profits*).

Core Equity = Closed Equity + Locked-In Profits

Expectation

```
( TotalWinPercent - TotalLossPercent ) / TotalRiskPercent
```

Total Win Percent = sum of (gain as percent of trade entry equity) of all winning trades, including washes
Total Loss Percent = sum of (loss as percent of trade entry equity) of all losing trades
Total Risk Percent = sum of (percent of trade entry equity risked) of all trades

Exponential Moving Average
Similar to a simple moving average (SMA), except that more weight is given to the latest data. Also known as an exponentially weighted moving average, this type of moving average reacts more quickly to recent price changes than a *Simple Moving Average*. Our Exponential Moving Averages start with a simple moving average of the number of days in the moving average, and then use the Exponential formula thereafter.

An *Exponential Moving Average* is computed according to the following formula:

$$\text{Today's MA} = \text{Yesterday's MA} + \frac{2}{N+1}(\text{Current Price} - \text{Yesterday's MA})$$

Exponential MA Days = (Wilder MA Days X 2) - 1

Forex Carry
An adjustment for the costs of borrowing and lending the two currencies involved in a Forex transaction.

Locked-In Profits
That portion of Open Equity that would be realized if all the current positions were exited at their current stop values.

Moving Average
An average measure of prices that is computed using a fixed number of days. For example, a 20 day moving average is computed using the last 20 days. There are two common types of moving averages: *Simple Moving Average*, and *Exponential Moving Average*.

N
A measure of volatility, N is a concept used by the Turtles (student's of Richard Dennis and Bill Eckhardt) to represent the underlying volatility of a particular market.

N is simply a 20-day *Average True Range*. Conceptually, N represents the average range in price movement that a particular market makes in a single day, accounting for opening gaps. N was measured in the same points as the underlying contract.

See the Original Turtle Rules Document for more info on N as used by the Turtles.

Open Equity
A measure of the value of positions that have been entered but not closed out. In Trading Blox, *Open Equity* is the current value of each position measured at the close of the previous day.

For a given long position:

Position Open Equity = (Current Price - Entry Price) X Contracts or Shares X Dollars per Point

For a given short position:

Position Open Equity = (Entry Price - Current Price) X Contracts or Shares X Dollars per Point

Percent Profit Factor
A measure of the relative profitability of a system. Like Profit Factor below except it is calculated using:

Percent Profit Factor = Average Win Percent / Average Loss Percent

Profit Factor
A measure of the relative profitability of a system. Profit Factor is calculated using:

Profit Factor = Total Winning Trade Profits / Total Losing Trade Losses

R-Multiple™
The concept of an R-Multiple™ was pioneered in 1993 by private trader Chuck Branscomb. R-Multiple™ is short for "Risk Multiple".

An R-Multiple™ is simply the profit or loss for a given trade divided by the entry risk. The entry risk is defined as the difference between the entry price and the stop price as the time of the entry.

To learn about the origin of R-Multiple™s, see the forum topic Improve Risk/ Reward by using Multiple Timeframes on the Trader's Roundtable forum.

R Squared
A statistical measure of how close an equity curve is to a straight line plotted on a logarithmic graph. A value of 0 indicates a jagged line and a value of 1.0 represents a straight line. A fixed percentage investment that compounded and paid daily would have a straight line equity curve and an R Squared value of 1.0.

Set File
A Set file is an ASCII Text file which defines a portfolio which will appear in the Portfolio manager section of each system. See: Portfolios for more information.

Simple Moving Average
A simple, or arithmetic moving average is calculated by adding the closing price of the security or futures market for a number of time periods, then dividing this total by the number of time periods. In other words, it's the average stock price over a period of time. Unlike the *Exponential Moving Average*, this type of moving average gives equal weight to each daily price.

Standard Deviation
In trading, a Standard Deviation of price data is often used as a measure of volatility. In this context it is usually measured over a fixed number of days rather than over the entire data set.

More generally, the standard deviation (SD) of a given data set is derived from its variance, i.e. the arithmetic mean of all the squared distances between the data values and their arithmetic mean:

$$\text{Variance} = \frac{1}{\text{N}} \sum_{i=1}^{N} (x_i - \overline{x}_{ari})^2$$

The standard deviation is defined as the square root of the variance:

$$\text{Standard Deviation} = \sqrt{\text{Variance}}$$

For more info see: http://mathworld.wolfram.com/StandardDeviation.html

Total Equity
Total Equity is a measure of equity that takes into account all closed out trades and open position equity (or Open Equity).

Total Equity = Closed Equity + Open Equity

Wilder Moving Average Formula
Welles Wilder's indicators, including Average True Range do not use the standard exponential moving average formula. The Wilder formula for computing an N-Day Exponential Moving Average is:

$$\text{Today's MA} = \text{Today's Price} \times \frac{1}{\text{N}} + \text{Yesterday's MA} \times \frac{\text{N - 1}}{\text{N}}$$

You can determine the equivalent non-Wilder *Exponential Moving Average* using the following formula:

Exponential MA Days = (Wilder MA Days X 2) - 1

Section 2 – Folders

The main Trading Blox folder "TradingBlox" contains the following sub folders:

Folder	**Contains**
Blox	contains the blox for the systems
Backups	contains the backup zip files that are created each time the application starts
Data	contains the sample data and dictionaries folder
Export	exported systems are placed here
Forex Sets	the forex portfolios (or Forex Set files)
Futures Sets	the futures portfolios (or Futures Set files)
Images	images used for HTML-based reporting
Log Files	normal.log file is here.
Import	systems to be imported on startup
Orders	the orders generated by the Suite | Generate Orders menu command
Results	The test results and reports. Contains information from every test run. Over time, this folder can grow fairly large -- you may want to empty this folder occasionally.
Service Reports	created service reports are here
Sounds	the sound files used by Trading Blox
Stock Sets	the stock portfolios (or Stock Set files)
Systems	contains the Blox systems
Simulation Suites	the Simulation Suite files

Section 3 – Technical Support

Trading Blox has a technical support team dedicated to helping you resolve any technical or functional issues you may experience. Technical support personnel can be reached by email at any time, or via our Internet discussion forum.

If your issue can be resolved on the spot, it will be. If necessary, someone from our support team will investigate the issue, and contact you directly with an answer as soon as one is available.

Current email, phone, and address here: http://www.tradingblox.com/tradingblox/contact.htm

Forum: http://www.tradingblox.com/forum/viewforum.php?f=58

If you can't access the above forum, please be sure you have registered and are logged onto the forum. If you still can't access this forum, you will need to update your customer profile in the Customer Login http://www.tradingblox.com/tradingblox/updateLogin.php section of the website. Here when you update your forum username, it will add you to the customer group. Then try the above link again.

Be sure to keep your customer profile update, so we have your latest address, phone, and email in case we need to contact you about new versions and updates.

PDF Manual Files

Please visit the Documentation webpage for the most recent PDF, Browser based, and Help File based Documentation for both Trading Blox and the Blox Builder.

Create Service Report

In many cases it is helpful for us to have a copy of your environment. We have a utility called Create Service Report, that you can access from the Help menu. When you have created the report, email us the output and we will be able to trouble shoot your issue more easily and quickly.

To create, view, and send the service report:

- Use the Help / Create Service Report menu item. This will create a zip file of your blox and systems. It will open the folder containing this file.
- Email this file to customer support. Check the website for the current customer support email address.

Section 4 – Risk and Return

The ideal investment is one that offers the greatest return, with the least amount of risk.

Thus, given the choice between two investments with identical returns, the one with the least fluctuation, or variability of returns, is generally deemed the superior investment. But as Jack Schwager points out in Managed Trading, Myths & Truths, "Too many investors take the mistake of focusing solely on return...It is also critical to incorporate some measure of risk as part of the evaluation process.

Adjusting for Risk

Opportunity involves risk. It is important, therefore, to understand the risks we take, in order to evaluate whether we are being justly rewarded. Measures of risk-adjusted return (such as the Sharpe and the MAR Ratios, do just that: They literally divide an investment's return by the risk required to achieve it.

They are all, in one form or another, ratios of gain-to- pain.

Between risk and return, return is by far the more tractable concept. Looking at an arithmetic average or a compounded rate of return will tell you just about everything you need to know about reward.

The evaluation of risk, on the other hand-trying to determine what risk is, and how to measure it-is a more daunting task, and one that is complicated by the fact that the human perception of risk, and our tolerance for it, varies greatly with circumstances such as age, health, and wealth (to name just a few), along with a host of psychological and emotional factors.

If you manage money for others, for instance, your tolerance for risk will be influenced strongly by your clients' risk preferences, and will be entirely different from that of an individual trader whose goal may be to shoot for the highest possible returns without blowing out.

Drawdown

Drawdown is measured as a percentage retracement from a previous equity peak (or account balance high). This downside risk statistic is sampled peak-to-valley, and typically uses marked-to-the-market daily or monthly data points.

Maximum Drawdown is the largest such negative excursion during the life of the simulation, and conveys the maximum pain an investor would have had to endure in order to achieve the resulting return. Maximum Total Equity Drawdown is the denominator of the MAR Ratio (which uses daily data points), and the Calmar Ratio (monthly data points).

Duration of the Longest Drawdown, like Maximum Drawdown, is a one-time event, and is measured from previous equity peak, to new equity peak. It is also an important performance statistic psychologically not only for individual traders, but for fund managers, who are prone to losing clients during extended underwater periods.

One reason why various measure of drawdown and its duration are such important statistics to many traders is that even good systems typically spend far more time in periods of drawdown than they do making new equity highs.
But while Maximum Drawdown and Period of Longest Drawdown represent unique, extreme downside risk events, they convey almost no information about overall volatility.

Standard Deviation

The overall variability of investment returns is best described by Standard Deviation. It is typically calculated based on monthly, or annual percentage changes (rather than dollar fluctuations of the equity curve series itself).
Standard Deviation measures both upside and downside volatility, and is literally the square root of variance, which describes the dispersion of data points around an average. Standard deviation is the denominator of the Sharpe Ratio, a classic measure of risk vs. reward.

Standard Deviation is an easy concept to convey without resorting to mathematical formulae, and the following example demonstrates why averages with large standard deviations can be misleading, and why two investments with similar returns can have dramatically different risk characteristics:

> *The U.S. Virgin Islands have an average annual temperature of approximately 80ºF. This is paradise, and it is habitable year-round.*
>
> *Let's compare this with an actual location in Southern California, not too far inland from the Pacific Ocean. It has a year-round average temperature of 76ºF, which is a little cooler than paradise. So far, so good.*
>
> *Research shows that the average daily high throughout the year is 90ºF, and the average daily low is 62ºF. Still habitable, but it's beginning to sound a little hot. Maybe not, though; the Virgin Islands has an average daily high/low of 86º / 74ºF (http://www.orcasailing.com/bvi-details.html)*
>
> *Our pleasant California locale also has very mild winters. More research, though, shows that while the average daily low for the summer is 77°F, the average daily summer high is...105°F. So it's apparent that, even thought its average annual temperature is slightly lower than the Virgin Islands, the range, or variance of its temperature is significantly higher, and may even be cause for alarm, where habitability is concerned.*
>
> *Further digging turns up the fact that 1996 was the hottest summer on record: There were 40 days over 120°F, and 103 days over 110°F. The summer of 1974 was no more hospitable, setting a record of 134 consecutive days with a maximum temperature of over 100°F. In 1913, the temperature reached 129°F or above, five days in a row (a world record at the time), and the hottest temperature ever recorded was 134°F on July 10 of that same year. (http://wrgis.wr.usgs.gov/docs/parks/deva/weather.html)*
>
> *Our pleasant-sounding climate, with a year-round average temperature lower than the Virgin Islands, now sounds more like an inferno. And it is: It's Furnace Creek in Death Valley, CA, the hottest, driest place in North America.*
>
> *So the vast difference in habitability between these two locales-which share similar year-round average temperatures-lies solely in the standard deviation of their respective temperatures.*

There is no doubt that, by treating upside and downside volatility the same, Standard Deviation does a very good job of describing the overall variability of investment returns. However, there is a school of thought which maintains that Standard Deviation is an inappropriate risk measure under certain circumstances because it unfairly penalizes the high upside volatility often experienced by trend-following Commodity Trading Advisors.

Trend-following CTAs typically employ rigid stop-loss strategies that produce a return profile characterized by a small number of well-contained losses, and an even smaller number of large winning trades. So the argument that standard deviation unfairly penalizes their risk-adjusted returns may well have merit.

In response, some risk-adjusted metrics-such as the Sortino Ratio-look only at downside Standard Deviation (also called semi-deviation). While similar to the Sharpe Ratio in general form, the denominator of the Sortino Ratio calculates the standard deviation of the negative data points only, against the mean of the population (of all data points).

Determining Your Own Measures

The various risk-adjusted return statistics all look at the same picture from slightly different angles, and there simply isn't a single metric that will satisfy every investor, under all circumstances. But understanding the components of these various statistics will allow you to determine which combination is right for you, and Trading Blox offers a number of risk-adjusted metrics that help you to do just that.

As a parting thought, it is worth noting that the distribution of returns from the world's financial markets are non-normally distributed, which means simply that extreme events are likely to occur more frequently that randomly (normally) distributed returns would dictate. In addition, there is evidence that the volatility of financial markets increases steadily with time. (Bernstein, Peter L. - Capital Ideas; The Improbable Origins of Modern Wall Street, pps. 21-22. The Free Press, 1992.)

Therefore, it is imprudent to think that the maximum risk statistics derived from any amount of historical testing will not be exceeded in the future. They most likely will be. Good traders count on it, and plan for it.

Section 5 – Data Providers

Below is a partial list of a few Data Providers, and their contact information.

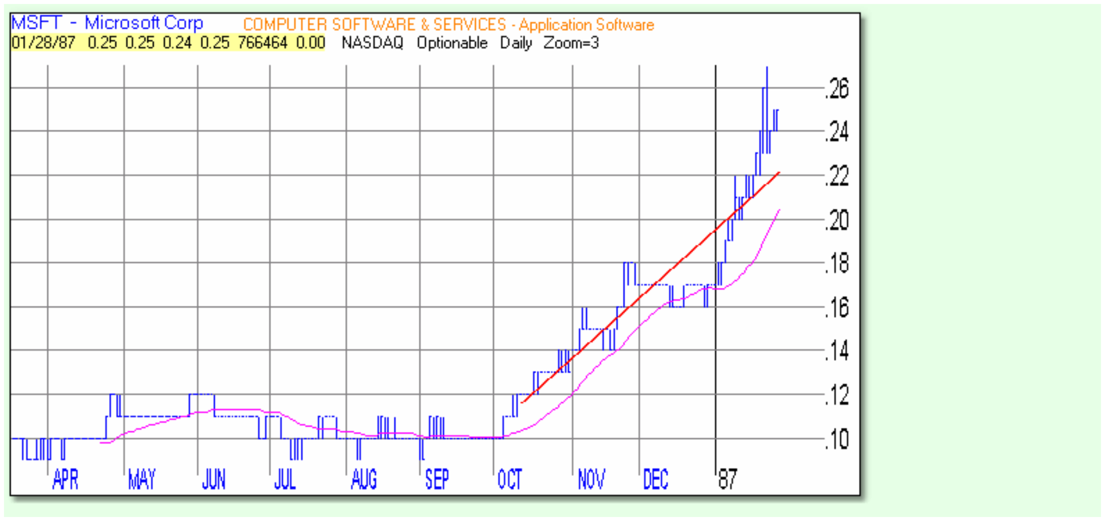
CSI
Commodity Systems, Inc., known in the industry as CSI, offers data in MetaStock and ASCII Text formats for many different U.S. and World futures and stock markets. CSI is the preferred provider of futures data and is the supplier of the sample data we provide.

Trading Blox includes specific Futures Dictionary files for CSI data for their default back-adjusted futures contracts in ASCII text formats. The default rolling algorithm is to use open interest and roll on the first trigger. This back-adjustment algorithm will result in symbols that have "`0_I0B`" appended to them. For example, the full CSI symbol for corn "C" using the default back-adjustment algorithm would be "`C__0_I0B`". The ASCII text file would be named "`C__0_I0B.txt`" and this symbol would be the symbol used inside the MetaStock MASTER file for that data.

Visit http://www.tradingblox.com/tradingblox/howto-ua.htm for a tutorial explaining setting up Trading Blox with data from CSI.

Worden Brothers

Their historical data is not completely accurate for stocks that have split many times because the way they have handled the stock splits has not retained enough precision. This has the effect of causing the resolution in the data to decrease the more splits have taken place. After many splits, the data becomes less than optimal for historical testing. Here's a sample graph showing this phenomenon:



Note how the prices for each day appear to have at most one or two prices. This is because the adjustments for splits have been constrained to $0.01 which is not enough resolution to show the actual pricing relationships between the different days.

Visit http://www.tradingblox.com/tradingblox/howto-worden.htm for a tutorial explaining setting up Trading Blox with data from Wordens.

Section 6 – Forex Carry Calculations

NOTE: This section assumes a familiarity with Forex trading. It is not intended as a tutorial. See http://fxtrade.oanda.com/currency_trading/intro_currency_exchange.shtml for an excellent tutorial of Forex concepts.

Here is an interest rate carry cost calculator:
http://www.oanda.com/products/fxmath/interest.shtml

Trading Blox accounts for the costs of Forex Carry by performing the following tasks each day after the close for each Forex position held:

1. Determine the quantity of both Base and Quote currency based on the price at entry. For long trades one buys the base currency and sells the quote currency, so the carry cost is determined by lending the base currency and borrowing the quote, and for shorts it lends the quote and borrows the base.
2. Determine the applicable historical lend and borrow interest rates for each currency for the day. The file contains the lend, then the borrow, rates. The calculations then assumes you borrow (pay) interest at the borrow interest rate from the history file, and that you lend (receive) interest at the lend interest rate from the file. The rate used for each day is from the current day.
3. Calculate the current days interest based on the quantity of each currency, the applicable rate for the currency, and the position direction (long or short). The base currency bought/sold is the Forex Trade Size from the global parameters. The quote currency bought/sold is this trade size times the current closing price of the forex pair.
4. Convert each of the resulting interest payments back into dollars at the then current exchange rates for the Base and Quote currencies. Rates used are current day rates.
5. Subtract what you need to pay from what you receive to get the net interest -- lending income minus borrowing payments.
6. Divide the total calculated yearly interest by 365.25 to determine the daily rate.
7. Interest is accrued/paid starting on the entry date, and includes weekends and holidays. Interest is not accrued/paid on the exit date.

Required Data Files for Forex Testing
To test a given forex instrument, you need:

- Price history for the instrument.
- If the forex pair is a cross, a USD to Base or USD to Quote price history so the entry and exit prices can be converted to dollars.
- If you turn on carry computation you also need an interest rate history file and USD conversion file for both the Base and Quote currencies. These can be sparsely populated to include only days when the interest rates change. For example, a USD history could be done containing only those days when the Fed changes the rates.

Here is a good source of interest rate data:
https://fx1.oanda.com/user/interestrate.html

More reading on the carry cost subject available here:
http://fxtrade.oanda.com/fxtrade/interest_calculation.shtml#

Section 7 – Determining Big Point Value

Simply put, the Big Point Value is the amount of money you will make if you buy one contract and it goes up by one full point. That is one number to the left of the decimal. So Gold going from 401 to 402, or Soybeans from 4 to 5. Be sure to look at your data to see how the prices are represented.

Different data vendors store the data for the same futures markets in different units.

For example, Data Provider A might store the same line of data for the British Pound in dollars like:

```
20020703, 1.4886, 1.4944, 1.4882, 1.4898, 4356, 41414
```

While Data Provider B might store a line of price data for the British Pound in cents like:

```
20020703, 148.86, 149.44, 148.82, 148.98,4 356, 41414
```

The Big Point Value Mapping reconciles this unitary difference (148.86 vs. 1.4886) for these two vendors The Big Point Value is multiplied by the prices stored in the price data file to determine the dollar value of a one point move in the underlying.

Using the above example, the British Pound entry in the Futures Dictionary for Data Vendor A would be:

```
62500
```

While the entry for British Pound for Data Vendor B would be:

```
625
```

These values, and all Big Point Values, derive from the contract size. According to the Chicago Mercantile Exchange, the BP contract size is "62,500 pounds sterling (British pounds)". Therefore if a data vendor lists the price for the BP contract in full dollars, i.e. like Vendor A above 1.4886, since there are 62,500 pounds in a contract. The Big Point Value is 62,500. If the vendor lists the price in cents like Vendor B above at 148.86, the Big Point Value is 625.

Using Vendor A's Pricing the price of a British Pound goes form 1.4886 to 2.4886, this will result in a gain of $62,500 since there are 62,500 British Pounds in a contract. Using Vendor B's pricing, this corresponds to a price movement of 100, from 148.86 to 248.86 which will also result in a gain of $62,500. Thus the Big Point Value for Vendor A is 100 times the size of the Big Point Value for Vendor B because the price units are 1/100th as large.

For a given price, the contract value should be the same. The contract value can be determined by multiplying the price by the Big Point Value for the market. Since the Big Point Value adjusts for unitary differences the price for a data vendor multiplied by the Big Point Value for that vendor should be the same as that of any other data vendor. Thus:

```
148.86 x 625 = 1.4886 x 62500
```

The Big Point Value should be *expressed directly in the currency in which the futures contract is priced*. For example, the SOFFEX Swiss Government bond contract is priced in Swiss Francs (ISO currency code CHF) so its Big Point Value should be expressed directly in terms of CHF.

Section 8 – Batch Job

You can generate orders from the command line.

You can use windows scheduler for this as well, if you create a .bat file.

Flags supported are:

–suite "suitename1" "suite name 2"	Suites to use
-test	Runs a test for each suite in the list
-orders	Generates orders for each suite in the list
-min	Starts Trading Blox in a minimized window
-hidden	Runs Trading Blox as a hidden window
-noProgress	Hides the progress dialog
-exit	Exits the program when finished. Runs silent mode: no breakpoints and no data loading error messages

To use, create a text file using notepad, with a .bat extension. Place .bat file in the Trading Blox folder.

Here is an example .bat file that opens the MySuite suite, generates orders, and then exits the application.

```
tradingblox.exe -suite "MySuite" -orders -exit
```

To run multiple .bat files in a row, or at the same time, use a Windows Script Host file.
Create a file in notepad with a .vbs extension. The following sets the current directory, and runs several bat files one after another.
The first parameter of the WSHShell.Run function is the bat file name, the second is the window style, and the third is wait on return.

```
Dim WSHShell
Set WSHShell = CreateObject("WScript.Shell")

WSHShell.CurrentDirectory = "C:\Testing\TradingBlox\"

WSHShell.Run "BatchRun1.bat", 0, true
WSHShell.Run "BatchRun2.bat", 0, true
WSHShell.Run "BatchRun3.bat", 0, true
```

The above example runs hidden, and the .bat file also uses the -hidden -noProgress flags, so these can run behind the scenes while the computer is used for other tasks. If the bat files reference different install folders of Trading Blox, then the wait on return parameter could be false, which would launch all these processes at the same time. A multiple simultaneous license of Trading Blox would be required to run multiple instances at the same time.

Reference on Windows Script Host:

http://msdn.microsoft.com/en-us/library/d5fk67ky(v=vs.85).aspx

Section 9 – Ini File

The following rarely used options are available in the tradingblox.ini file. To change, close Trading Blox, open the TradingBlox.ini file in Notepad, modify the setting, save, and then start Trading Blox.

The **Dividend File Suffix** is used as a naming convention for the dividend files, as the suffix to the symbol. The dividend file for IBM is IBM_Div.csv.
The **Graph File Suffix** is used as the suffix for all the summary graphs. Changing to .gif produces a slightly larger but more portable file, and changing to .jpg produces a large file with more defined colors, less compression.
The **Process Weekends** flag controls weekend processing. Normally if a weekend date is found anywhere in the data, then all weekends are included in the test loop. If this flag is set to false, then weekends are excluded from the test loop.
The **Add Non Traded Instruments** flag controls the process whereby instruments with no trades for a particular test are listed in the trade report with a zero trade. This zero trade is there so that you can access the chart and indicators, to explore why the instrument did not trade. But if you have 1000's of these in a large stock test, you might want to set this flag to false.
The **Summary Report Test Maximum** controls how many tests over which the html summary output will be truncated. Too many test results in an html file make the file so large that windows cannot open the file. So this number is usually set to 5000 or less.
The **Risk Free Rate** as used in some sharpe computations (Annual Sharpe but not Modified Sharpe).
The **Automatic Blox Script Parsing** flag controls whether blox scripting will parse the script as each character is pressed. This is the default setting so that you can catch syntax errors immediately. If the blox script is so huge that this parsing process takes a few seconds, you can turn this automatic parsing off. The script is still parsed for syntax errors when the script is saved, on exit or by pressing control-s.
The **Blox Editor Font Size** sets the font size in the Blox Editor
The **Blox Editor Font** sets the font used by the Blox Editor
The **Backup Archive Days** sets the number of days of backups to keep in the Backups folder.
The **Multi Parameter Chart** turns on/off the two multi parameter chart styles: contoured and 3D.
The **Global Heap Size** sets the amount of memory the Trading Blox will allocate to a private memory heap on application startup. Setting to zero uses the default windows settings.
The **Use Jelly Fish Editor** tells the Blox Editor to save the scripts when opened into a .bas file in the Blox/JellyFish folder and open with Jelly Fish Pro. Script editing in the Blox Editor will be disabled.
The **Futures Data SSV** is not used. SSV is Space Separated Values. It was implemented to accommodate Tradestation data years ago, but is no longer used or enabled.
The **Integrity check** enables more strict rules on data checking such as errors when the exchange or currency does not exist.
The **Use Population SD** sets the formula used by Trading Blox for Standard Deviation. By default this is true and uses Population Standard Deviation. If Sample is required, then set to false.
The **Display Last Roll Date** controls the historic roll info on the positions section of the order generation report.
The **Memory Usage Threshold** sets the percent of physical memory to use before raising an error. Default is 90, but this can be set higher or lower as needed. A setting of -1 will disable this check.
The **Stock Margin Short Rate** sets the margin rate for short stock sales. Default is 100%.
The **Stock Margin Long Rate** sets the margin rate for long stock purchases. Default is 100%, but some brokers require more or less margin for short sales.

Default Settings:

```
[TradingBlox Directories]
Dividend File Suffix=_Div.csv
```

```
Graph File Suffix=.png

[Data]
Futures Data SSV=FALSE
Integrity Check=FALSE
Process Weekends=TRUE
Global Heap Size=0

[General Reporting]
Add Non Traded Instruments=TRUE
Summary Report Test Maximum=5000
Multi Parameter Chart 1=TRUE
Multi Parameter Chart 2=TRUE
Risk Free Rate=0.03
Use Population SD=TRUE
Display Last Roll Date=TRUE

[Block Basic]
Automatic Blox Script Parsing=TRUE
Blox Editor Font Size=14
Blox Editor Font=Courier New
Use Jelly Fish Editor=FALSE

[Application Settings]
Backup Archive Days=30
Memory Usage Threshold=90

[Backtesting]
Stock Margin Short Rate=100
Stock Margin Long Rate=100
```

Section 10 – Backups

Backups are created as the application starts for the first time each day, and can be created manually from the File/Backup menu. The program will launch the backup.bat file, which has a list of folders and files to backup. These are all compressed into a zip file and date time stamped. They are archived in the Backups folder. By default, backups older than 30 days are deleted. The number of days can be changed in the Trading Blox INI file under Backup Archive Days.

Here is the list of folders and files that are backed up by default. The files and folders to backup can be changed by modifying this file using notepad.

```
7za a -tzip Backups/CurrentBackup.zip "Systems\*"
7za a -tzip Backups/CurrentBackup.zip "Blox\*"
7za a -tzip Backups/CurrentBackup.zip "Extensions\*"
7za a -tzip Backups/CurrentBackup.zip "Import\*"
7za a -tzip Backups/CurrentBackup.zip "Export\*"
7za a -tzip Backups/CurrentBackup.zip "Test Settings.ini"
7za a -tzip Backups/CurrentBackup.zip "TradingBlox.ini"
7za a -tzip Backups/CurrentBackup.zip "Data\Forex\ForexInfo.*"
7za a -tzip Backups/CurrentBackup.zip "Data\Futures\FuturesInfo.*"
7za a -tzip Backups/CurrentBackup.zip "Data\Stocks\StockInfo.*"
7za a -tzip Backups/CurrentBackup.zip "Data\Dictionaries\*"
7za a -tzip Backups/CurrentBackup.zip "Simulation Suites\*"
7za a -tzip Backups/CurrentBackup.zip "Stock Sets\*"
7za a -tzip Backups/CurrentBackup.zip "Futures Sets\*"
7za a -tzip Backups/CurrentBackup.zip "Forex Sets\*"
```

Section 11 – Moving the License

Installation on a new computer

The license agreement and authentication process allows for one user to install on two computers for non simultaneous use. The first two computers you install the application on will be the ones the authentication server will accept. You can always re-install as many times as you need to on these two computers. If you try to install on a third computer, it will not authenticate. You must be connected to the internet to authenticate your installation for the first time on a new computer. The authentication process uses both the computer username and the computer identification, so two users logged into the same computer would be seen as two separate installations.

In the case where you get a new computer and would like to MOVE an install from an old computer to this new computer, you must use the Remove License menu option to first remove the license from your old computer. This will only remove the license, and not the Trading Blox folder. You will then be able to install the application on the new computer, or if you want to save all your changes you can just move the entire Trading Blox folder from the old computer to the new computer. You may need to update your data folder locations and such. Do not just uninstall the application from the old computer, as that will not remove the license from that computer.

If you have a hard drive failure or some other catastrophe that makes it impossible to use the Remove License process, you can reset your license account from the Customer Login section of the website. Be sure to keep your maintenance current so that this online process will be immediate. If your maintenance is not current, you will need to request a manual reset. You can do all of this from the Customer Login section of the website.

Index