คู่มือการใช้งานเบื้องต้น AmiBroker 6.0

ฉบับแปลภาษาไทย

โดย **SiamQuant**

จุดเริ่มต้นของการลงทุนอย่างเป็นระบบ

คู่มือการใช้งาน Amibroker 6.00 เบื้องต้น V.1.0

ฉบับแปลภาษาไทย โดย

SiamQuant : จุดเริ่มต้นของการลงทุนอย่างเป็นระบบ

วิธีการใช้งานคู่มือฉบับแปลไทย **12**

การทำงานพื้นฐาน **14**

เพิ่มชื่อหลักทรัพย์ใหม่ (Adding a new symbol) 14

ลบชื่อหลักทรัพย์ (Removing a symbol) 14

แตกหุ้น(Splitting a stock) 14

ลบจำนวนหุ้น (Deleting quotation) 15

เพิ่ม/ลด ชื่อหลักทรัพย์จากรายการโปรด (Adding/removing symbol from favourites) 15

รวมจำนวนหุ้นจาก หลักทรัพย์ 2 ตัว (Merging quotations of two symbols) 16

คู่มือการใช้ชาร์ทเบื้องต้น **17**

เกริ่นนำ (Introduction) 17

การทำงานพื้นฐาน (Basic operations) 19

การเลื่อน (Scrolling) 19

การซูม (Zooming) 19

ย่อ, ขยาย และขยับสเกลของแกน Y (Shrinking, expanding and moving Y-axis scale) 19

เปลี่ยนไทม์เฟรมของแท่ง (Changing bar interval (periodicity)) 20

Selecing a quote 21

เลือกช่วง (Marking range) 22

เพิ่ม / ปิดหน้าต่างชาร์ท (Adding / closing chart panes) 22

เชื่อมโยงและล็อคชาร์ท (Linking and locking chart) 22

การใช้เครื่องมือวาด (Using drawing tools) 23

วิธีใช้งานการลากและวางชาร์ท(Indicator) **27**

เกริ่นนำ (Introduction) 27

จะใส่อินดิเคเตอร์ใหม่ได้อย่างไร (How to insert a new indicator) 27

จะทำยังไงให้อินดิเคเตอร์อันนึงวางซ้อนบนอีกอันนึงได้ (How to overlay one indicator on another indicator) 28

จะลบอินดิเคเตอร์อย่างไร (How to delete the indicator) 29

จะลบอินดิเคเตอร์ในหน้าต่างนั้นออกไปอย่างไร (How to remove the indicator plot from the pane) 30
จะเปลี่ยนค่าพารามิเตอร์ สี และสไตล์ของอินดิเคเตอร์อย่างไร (How to change parameters/colors/styles of indicators) 31
จะใส่อินดิเคเตอร์ที่มีสเกลต่างกันซ้อนกันได้อย่างไร (How to overlay indicators with different scales) 31
จะสร้างอินดิเคเตอร์ใหม่โดยอ้างอิงบนอินดิเคเตอร์อื่นอย่างไร(How to create an indicator based on another indicator) 32
การใช้คำสั่ง Param(), ParamColor(), ParamToggle(), ParamStyle() (Using Param(), ParamColor(), ParamToggle(), ParamStyle() functions) 33
คำถามที่พบบ่อยเกี่ยวกับฟังก์ชั่นลากและวาง (Frequently Asked Questions about drag & drop functionality) 39

ธีมของชาร์ท 43

1. ธีมพื้นฐาน (Basic Theme) 44
2. ธีมแบบง่าย(Nature Simple Theme) 45
3. ธีมแบบมีสี (Nature Gradient Theme) 46
4. ธีมสีเทา (Gray Theme) 47
5. ธีมสีเทาเข้ม (Dark Gray Theme) 48
6. ธีมสีดำ (Black Theme) 49

การปรับแต่งหน้าจอผู้ใช้ 50

การเอากลุ่มบางอันออก / การซ่อนแท็บ (Advanced nested docking / tear-off tabs) 50
การซ่อนหน้าต่างอัตโนมัติ (Sliding Auto-hide panes) 51
การปรับแต่งแถบเครื่องมือ เมนู และปุ่มลัดขั้นสูง (Advanced customizable toolbars, menus and keyboard shortcuts) 53
ธีมที่ปรากฏ (Themed appearance) 55
แท็บ MDI (multiple document interface) (MDI (multiple document interface) tabs) 56

การใช้งานหน้าชาร์ทและหน้าต่าง Layout 59

หน้าชาร์ทและรูปแบบ (Chart sheets and templates) 59
ชื่อหลักทรัพย์และการเชื่อมโยงภายใน (Symbol and Interval Linking) 63
การแยกหน้าต่าง (Floating windows) 64
โครงสร้างของหน้าต่าง (Window layouts) 65

การใช้งาน Layers 67
Layers คืออะไร (What layers are) 67
Layers ทำงานอย่างไร(How to work with layers) 67
เมนูย่อย (Context menu) 68

การใช้งานหน้าต่างการวิเคราะห์แบบใหม่ 70
เกริ่นนำ (Introduction) 70
หน้าจอผู้ใช้ (User interface) 71
การทำงานพื้นฐาน (Basic operations) 73
ระบุตัวกรอง (Defining Filter) 74
กำหนดช่วงวันและเวลา (Defining date/time Range) 74
การดูผลรายงาน / เปิดหน้า Report Explorer (Viewing reports / Running Report Explorer) 74
เปลี่ยนการตั้งค่า / ตัวเลือก (Changing settings / options) 75
ใช้การเทสแบบ Walk Forward (Running Walk forward Test) 75
แสดงผลการ Optimize แบบชาร์ท 3 มิติ (Displaying 3D optimization chart) 76
ส่งออกและนำเข้าผลลัพธ์ (Exporting and importing result list) 76

การจัดการ Watch List 78
การ เพิ่ม/ลบ Watchlist (Adding / removing watch lists) 78
การนำ tickers เข้าสู่ watch lists (Adding tickers to watch lists) 79
การเรียงลำดับ tickers in a watch list (Sorting tickers in a watch list) 80
การลบ tickers ออกจาก watch list(Removing tickers from watch lists) 80
การลบ watch list หรือ ล้างรายชื่อที่อยู่ใน watch list ของคุณ(Erasing watch lists) 81
ซ่อน/ปิดการซ่อน watch list ต่างๆที่ว่างเปล่า (Hiding/Unhiding empty watch lists) 81

การใช้ watch lists ในหน้าต่าง Automatic analysis (Using watch lists in Automatic analysis window) 81
จะ นำเข้า/นำออก watch list จาก/ไปยัง ไฟล์ ได้อย่างไร 82
การนำเข้า watch list จากไฟล์ใดๆ (IMPORT WATCH LIST FROM FILE) 82
นำออก WATCHLIST ไปยังไฟล์ (EXPORT WATCHLIST TO FILE) 83
จะ นำเข้า watch list จากฐานข้อมูลภายนอก หรือ จะนำออก watch list ไปสู่ฐานข้อมูลภายนอกได้อย่างไร (How to import/export watch list from/to external database) 84
การนำเข้า Family จาก Fasttrack (IMPORT FAMILY FROM FASTTRACK) 84
การนำออก Family ไปยัง Fasttrack (EXPORT WATCHLIST TO FASTTRACK FAMILY) 86

ทำความเข้าใจว่า AFL นั้นทำงานอย่างไร 87

บทนำ (Introduction) 87
อาร์เรย์ คือ อะไร? (What is an Array?) 88
การทำงานของอาร์เรย์ - ทำไม AFL ถึงทำงานได้รวดเร็วมากนัก(Processing arrays - why is AFL so fast?) 88
Moving averages เงื่อนไขของคำสั่ง (Moving averages, conditional statements) 89
ลองเพิ่มความซับซ้อนเข้าไปอีกหน่อย (Getting little bit more complex) 90
IIF(condition, truepart, falsepart) ฟังก์ชัน 92
AMA(array, factor) ฟังก์ชัน 92
การสร้างลูปใหม่ (New looping) 93

สร้างอินดิเคเตอร์ด้วยตัวคุณเอง 94

การใช้ปรับใช้ styles, colors, titles และ parameters ในกราฟ Indicator 100

Plot() function 100
การกำหนดค่าของสี (Color Constants) 101
การกำหนดรูปแบบของกราฟ (Style) (Style Constants) 104
X-shift feature 107
PlotForeign() function 107

การพล็อตกราฟหลายๆอันด้วยสเกลที่ต่างกัน (Multiple plots using different scaling) 108
การใช้การกำหนดพารามิเตอร์ด้วยตนเอง (Using custom defined parameters) 110
การพล็อตข้อความตัวหนังสือลงบนกราฟ (Plotting texts at arbitrary positions on the chart) 111
การใส่สีแบบไล่ระดับลงบนพื้นหลัง (Gradient fill of the background) 112
การไล่ระดับสีบนพื้นที่ของกราฟ (Gradient fill area charts) 113
การทำให้กราฟมีความหนา (Super thick charts) 115
จิปาถะ (Miscellaneous) 115

คุณจะสร้างตัว Exploration ด้วยตัวเองได้อย่างไร 118

กราฟ Scatter (X-Y) พล็อต ใน Exploration (Scatter (X-Y) charts in Exploration) 122
เคล็ดลับส่งท้าย (Final tip) 125

วิธีการเขียนคอมเม้นท์บนชาร์ท 125

การเขียนอักษรธรรมดา (Writing static texts) 126
สีและไสตล์ (Colors and styles) 127
ข้อความที่แปรไปตามการประมวลผล (Dynamic content) 127
เงื่อนไขต่างๆของผลลัพธ์ที่มีลักษณะเป็นตัวหนังสือ (Conditional text output) 129

การใช้งานเครื่องมือวาดบนสตร AFL 132

การวาดเส้นเทรนด์ไลน์ (Drawing trend line) 132
กำหนด Study ID (Define study ID) 132
การเขียนโค้ดเพื่อระบุการเบรคของเส้นเทรนด์ไลน์ (Write the formula that checks trend line break) 134

การ Backtest ไอเดียการเทรดของคุณ 135

เกริ่นนำ (Introduction) 135
การเขียนกฎการลงทุน (Writing your trading rules) 135
การ Back Test (Back testing) 136
การวิเคราะห์ผลลัพธ์ (Analysing results) 136
การเปลี่ยน Setting (Changing your back testing settings) 137

คำสั่ง Default (Reserved variable names) 137
คอนเซประดับสูง (Advanced concepts) 139
การชอร์ท (Short trade support) 139
การกำหนดราคาซื้อ (Controlling trade price) 140
การตั้ง Stop (Profit target stops) 143
"Exit at stop" 143
Trailing stops 144
Dynamic stops 145
การเขียนโค้ดสร้างจุด Stop (Coding your own custom stop types) 146
Position sizing 147
จำนวนหน่วยการลงทุน และ การเคลื่อนไหวของราคา (Round lot size and tick size) 149
Round lot size 149
Tick size 149
บัญชี Margin (Margin account) 150
Setting อื่นๆ (Additional settings) 150

Portfolio-level backtesting 152

ขั้นตอนการติดตั้ง (HOW TO SET IT UP ?) 152
การตั้งจำนวน position สูงสุดในการเทรดหนึ่งครั้ง (SETTING UP MAXIMUM NUMBER OF SIMULTANEOUSLY OPEN TRADES) 152
การตั้งขนาดการเทรดต่อหนึ่งการเทรด (SETTING UP POSITION SIZE) 153
การใช้ Position Score (USING POSITION SCORE) 154
BACKTEST MODES 155
ROTATIONAL TRADING 158
HOLDMINBARS และ EARLY EXIT FEES (HOLDMINBARS AND EARLY EXIT FEES) 158
การแก้ไขปัญหาการเข้าซื้อพร้อมๆกัน (RESOLVING SAME BAR, SAME SYMBOL SIGNAL CONFLICTS) 161
กรณี 1. Only one signal per symbol is taken at any bar (1 Bar 1 Signal) 162

กรณีที่ 2 สัญญาณในการเข้า และ สัญญาณในการออกเกิดขึ้นพร้อมๆกัน (Both entry and exit signals are used and entry signal precedes exit signal) 163
กรณีที่ 3. สัญญษณทั้งสองเกิดขึ้นพร้อมๆกัน แต่สัญญษณในการเข้ามาทีหลังสัญญาณในการออก (Both signals are used and entry signal comes after exit signal) 164
ใช้งานในรูปแบบของพอร์ต (How does it work in portfolio case?) 165
กลยุทธ์แบบ market-neutral, long-short balanced strategies (Support for market-neutral, long-short balanced strategies) 165
SeparateLongShortRank backtester option 166
การอ่านผลการ Backtest 168
Exposure % 169
Net Risk Adjusted Return % 169
Annual Return % 169
Risk Adjusted Return % 169
Avg. Profit/Loss 169
Avg. Profit/Loss % 170
Avg. Bars Held 170
Max. trade drawdown 170
Max. trade % drawdown 170
Max. system drawdown 170
Max. system % drawdown 170
Recovery Factor 170
CAR/MaxDD 170
RAR/MaxDD 171
Profit Factor 171
Payoff Ratio 171
Standard Error 171
Risk-Reward Ratio 171

Ulcer Index 171
Ulcer Performance Index 171
Sharpe Ratio of trades 171
K-Ratio 172
การใส่สีเข้าไปในผลการ backtest (ใหม่ในเวอร์ชั่น 5.60) (Color-coding in the backtest report) 172

วิธีการ Optimize 173

เกริ่นนำ (Introduction) 174
โค้ด AFL formula (Writing AFL formula) 174
แสดงผล optimization แบบสามมิติ (Displaying 3D animated optimization charts) 177
การควบคุมกราฟสามมิติ (3D chart viewer controls) 179
โหมด Smart (non-exhaustive) optimization (Smart (non-exhaustive) optimization) 180
เกริ่นนำ (Introduction) 180
Quick Start 180
ขั้นตอนเป็นอย่างไร (How does it work ?) 181
SPSO - Standard Particle Swarm Optimizer 183
TRIBES - Adaptive Parameter-less Particle Swarm Optimizer 184
CMA-ES - Covariance Matrix Adaptation Evolutionary Strategy optimizer 185
Multi-threaded individual optimization 187

Walk-forward testing 189

การสร้างเส้น IN-SAMPLE และ OUT-OF-SAMPLE (IN-SAMPLE and OUT-OF-SAMPLE combined equity) 193
บันทึกสรุปผลลัพธ์โดยรวมจากข้อมูล OUT-OF-SAMPLE (ใหม่สำหรับเวอร์ชั่น 5.60) (OUT-OF-SAMPLE summary report (new in 5.60)) 194

การจำลองด้วยวิธี Monte Carlo 196

บทนำ (Introduction) 196
แล้วมันทำงานอย่างไรบน AmiBroker? (How does it work in AmiBroker?) 197
การตั้งค่า (Settings) 198

Enable Monte Carlo simulation 199
Number of runs 199
Position sizing 199
Enable MC equity curves (Min/Max/Avg) 200
การอ่านและตีความผลลัพธ์จากการทดสอบแบบ Monte Carlo (Interpreting the results) 201
จะควบคุมมัน จากการใส่สูตร/โค้ด (formula) ได้อย่างไร? (How to control it from the formula level?) 204
จะเพิ่มตัวบ่งชี้ที่กำหนดขึ้นเองในการทำ Monte Carlo ใน backtest Report ได้อย่างไร? (How to add custom metric based on MC test distribution(s) to the backtest report ?) 205
จะใช้การสุ่มแบบ Monte Carlo แทนการใช้การทดสอบแบบ bootstrap ได้อย่างไร? (How about Monte Carlo randomization instead of bootstrap test?) 207

Pyramiding (scaling in/out) และ การปรับใช้อัตรตราแลกเปลี่ยนค่าเงินในการ backtest พอร์ทการลงทุน 209

Pyramiding / Scaling 209
ตัวอย่าง อย่างง่าย (Easy examples) 211
ตัวอย่างที่ 1: dollar-cost averaging (แต่ละเดือนซื้อหุ้นด้วยจำนวนเงินที่เท่าๆกัน) (Example 1: dollar-cost averaging (each month you buy stocks for fixed dollar amount)) 211
ตัวอย่างที่ 2: dollar-cost averaging (เป็นโค้ดอย่างง่าย เนื่องจาก amibroker ถือว่าหากมีสัญญาณครั้งแรกให้ซื้ออยู่แล้ว) (Example 2: dollar-cost averaging (simplified formula because AB treats first sigScaleIn as buy anyway)) 211
ตัวอย่างที่ 3: เพิ่มขนาดของ position เมื่อเทรดแล้วได้กำไร โดยปราศจากการทำ Pyramiding เพิ่มขนาดเมื่อกำไรมากกว่า 5% และลดขนาดเมื่อ ขาดทุนมากกว่า -5% (Example 3: increasing position when profit generated by trade without pyramiding becomes greater than 5% and decreasing position when loss is greater than -5%) 212
ตัวอย่างที่ 4: การออกบางส่วน (scaling out) เมื่อถึงเป้ากำไรที่เรากำหนด (Example 4: partial exit (scaling out) on profit target stops) 213
Mulitple Currency Support 215

การเขียนโค้ดเพื่อตั้งการแจ้งเตือน 217

บทนำ(Introduction) 217
การตั้งค่าต่างๆ (Settings) 217
ฟังก์ชัน AlertIF (AlertIF function) 219
หมายเหตุ (Notes) 221

การเปิดใช้หน้าต่างการตีความ 223

การสนับสนุนการใช้ Multiple Time Frame ด้วย AFL 225
มันทำงานได้อย่างไร ? (How does it work internally ?) 231

ฟังก์ชั่นการจัดอันดับ 235
StaticVarGenerateRanks ฟังก์ชัน (StaticVarGenerateRanks function) 237
จะสามารถใช้ StaticVarGenerateRanks ใน Analysis window ได้อย่างไร (How to use StaticVarGenerateRanks in Analysis window) 242

คีย์ลัดในโค้ด AFL 244
การกำหนดคีย์ลัดของตัวคุณเอง (DEFINING YOUR OWN SNIPPETS) 245
TECHNICAL INFO (สำหรับผู้ใช้ขั้นสูง) (TECHNICAL INFO (advanced users only)) 248

คู่มือการใช้งานแบบวิดีโอ (on-line) 250

วิธีการใช้งานคู่มือฉบับแปลไทย

การทำความเข้าใจตัวอย่าง Coding ในคู่มือ

เพื่อให้ผู้อ่านมีความเข้าใจมากขึ้น ผู้แปลแนะนำในผู้อ่านลองนำโค้ดในคู่มือ ไปลองเขียนใส่ไว้ในโปรแกรม Amibroker แล้วของรัน หรือ ลองให้มันแสดงผลดูนะครับ

วิธีการ เชื่อมโยง/อ้างอิง เนื้อหาในคู่มือฉบับนี้ กับตัวต้นฉบับ

ในแต่ละหัวข้อไม่ว่าจะเป็นหัวข้อใหญ่ หรือ หัวข้อย่อย ทางผู้แปลได้ทำการแนบชื่อภาษาอังกฤษของหัวข้อนั้นๆเพื่อให้ผู้อ่านมีความสะดวกสบายในการใช้งานเรียบร้อยแล้วครับ วิธีใช้ง่ายๆดังนี้

หัวข้อใหญ่ หน้าตาจะเป็นแบบนี้ครับ :

ชื่อหัวข้อใหญ่นั้นๆ

(ตามด้วยวงเล็บสีน้ำเงินที่เป็นชื่อภาษาอังกฤษของหัวข้อนั้นๆ)

ผู้ใช้สามารถที่จะนำชื่อในวงเล็บสีน้ำเงิน + คำว่า Amibroker ไปค้นหาใน google ได้เลยครับ เช่น หากผู้อ่านต้องการเชื่อมโยงหัวข้อ “การทำงานพื้นฐาน” ก็ให้ผู้อ่านใช้ Keyword นี้ในการค้นหาใน google ครับ

Basic operations Amibroker

จากใจทีมงาน SiamQuant

คู่มือการใช้งานเบื้องต้น Amibroker ฉบับนี้ เกิดขึ้นด้วยความประสงค์ที่ต้องการจะร่วมถวายเป็นพระราชกุศลเนื่องในโอกาสวันคล้ายวันพระบรมราชสมภพ พระบาทสมเด็จพระปรมินทรมหาภูมิพลอดุลยเดช ปีที่ ๘๙ วันที่ ๕ ธันวาคม ๒๕๕๙ โดยมีจุดมุ่งหมายให้เป็นแหล่งความรู้ในการใช้โปรแกรม Amibroker ให้กับนักลงทุนไทยทุกๆคน

พวกเราหวังเป็นอย่างยิ่งว่าคู่มือการใช้งาน Amibroker ฉบับนี้จะเป็นประโยชน์แก่เพื่อนๆนักลงทุนที่มีความสนใจในการลงทุนอย่างเป็นระบบเชิง Quantitative and Systematic Trading ไม่มากก็น้อย โดยหากคู่มือเล่มนี้มีข้อผิดพลาดประการใด พวกเราต้องขออภัยเพื่อนๆนักลงทุนทุกท่านไว้ ณ ที่นี้ด้วย

โดยหากว่าท่านมีข้อสงสัยหรือข้อแนะนำประการใดเกี่ยวกับคู่มือฉบับนี้ กรุณาสอบถามพูดคุยกับพวกเราได้ที่กล่อง Comment ด้านล่างของ www.siamquant.com/thai-amibroker-manul ตามอัธยาศัยครับ

แหล่งที่มา

คู่มือการใช้งานเบื้องต้น Amibroker ฉบับภาษาไทยนี้แปลจาก
https://www.amibroker.com/bin/UsersGuide.pdf (ไฟล์ PDF)
http://www.amibroker.com/guide/tutorial.html (ไฟล์ Online)
โดยคุณ มด แมงเม่าคลับ (www.mangmaoclub.com) ได้รับอณุญาตอย่างถูกต้องจากคุณ Tomasz Janeczko เรียบร้อยแล้ว

การทำงานพื้นฐาน

(Basic operations)

เพิ่มชื่อหลักทรัพย์ใหม่ (Adding a new symbol)

ในการที่จะเพิ่มชื่อหลักทรัพย์ใหม่เข้าไปในฐานข้อมูล ให้คุณเข้าไปที่ *Symbol > New* หรือ คลิกปุ่ม Add symbol ในแถบ Toolbar หลังจากที่ทำตามขั้นตอนข้างต้นแล้ว คุณจะได้รับการแจ้งว่ามีชื่อหลักทรัพย์ใหม่เพิ่มเข้ามา โดยที่ความยาวสูงสุดของตัวอักษรนั้น จะอยู่ที่ 48 ตัวอักษร ตัวอักษรที่เหมาะสมสำหรับการนำเข้าข้อมูลนั้นจะต้องเป็นตัวพิมพ์ใหญ่

ลบชื่อหลักทรัพย์ (Removing a symbol)

สำหรับการลบชื่อหลักทรัพย์ที่มีอยู่ในฐานข้อมูลนั้นให้เข้าไปที่ *Symbol > Remove* ในแถบเมนู หรือจะคลิกที่ปุ่ม Remove Symbol ในแถบ Toolbar ก็ได้ เมื่อกดแล้วจะมีการถามเพื่อยืนยันอีกครั้งว่าต้องการจะลบหรือไม่ (โปรดจำไว้ว่าขั้นตอนนี้ไม่สามารถย้อนกลับได้ !) การลบหลายๆ ชื่อหลักทรัพย์พร้อมกันสามารถทำได้ โดยใช้ Assignment Organizer

แตกหุ้น(Splitting a stock)

ในการที่จะแตกหุ้น สามารถเข้าไปได้ที่เมนู *Symbol > Split* หรือ กดที่ปุ่ม Split ในแถบ Toolbar ก็ได้ Amibroker ได้นำเสนอวิธีการที่ง่าย ในการที่จะแตกหุ้นเองได้ โดยโปรแกรมจะพยายามคาดเดาวันที่แตกหุ้น และ คาดคะเนอัตราส่วนต่างๆ โดยการวิเคราะห์จำนวนหุ้น หากมีเพียงแค่หุ้นเดียวหลังจากแตกหุ้นแสดงว่ามันได้ผล แต่ถ้าไม่ได้ คุณควรค้นหาวันที่แตกหุ้น และ สัดส่วนในการแตกหุ้นในครั้งนั้นๆ (จากผู้แปล : แต่ถ้าคุณใช้ Data จากทาง Siamquant คุณก็ไม่จำเป็นต้องกังวลเกี่ยวกับเรื่องนี้) โปรดจำไว้ว่าขั้นตอนนี้ไม่สามารถย้อนกลับได้ !

แต่เวอร์ชั่น 2.0 ขึ้นไป ฟังก์ชั่นการแตกหุ้นสามารถทำได้มากกว่านั้น คุณสามารถใช้สัดส่วนแบบเก่า หรือ คุณสามารถระบุการแตกหุ้นแบบเฉพาะเจาะจงได้ด้วยคำสั่งนี้

x->y

คำสั่งนี้หมายความว่า หุ้นจำนวน x หุ้นก่อนแตกหุ้น เปลี่ยนมาเป็น y หุ้นในเวลาต่อมา ยกตัวอย่างเช่น 2->3 ก็หมายความว่า จาก 2 หุ้นเป็น 3 หุ้น ดังนั้นถ้าต้องการแตกหุ้นเป็น 5 หุ้น ก็ใช้คำสั่ง 1->5

คุณน่าจะพอเดาได้ว่าการแตกหุ้นสามารถย้อนกลับได้ เช่น การใช้ 2->1 ด้วยคำสั่งนี้แปลว่าหุ้น 2 หุ้นจะถูกรวมเข้าด้วยกันเป็นหุ้นเดียว

ลบจำนวนหุ้น (Deleting quotation)

การที่จะลบจำนวนหุ้นนั้นสามารถทำได้ง่ายๆ ด้วยการเลือกชื่อหุ้นที่คุณต้องการโดยคลิกที่ชาร์ทหุ้น (เส้นแนวตั้งที่จะปรากฏขึ้นเมื่อเลือกวันที่หรือหุ้น) จากนั้นเลือก *Edit > Delete*

สำหรับการลบจำนวนหุ้นจากหุ้นทุกตัวภายในระยะเวลาที่กำหนดคุณต้องเลือก *Edit > Delete session*

และ ใช้ Quote Editor สำหรับลบหุ้น

เพิ่ม/ลด ชื่อหลักทรัพย์จากรายการโปรด (Adding/removing symbol from favourites)

สำหรับการเพิ่มสัญลักษณ์เข้าไปในรายการโปรด คุณต้องติ๊กเลือกช่อง Favourite ในหน้าต่าง Information ส่วนการลบมันออกจากรายการโปรด ก็เพียงแค่เอาเครื่องหมายติ๊กถูกออกไป หรือ อีกทางเลือกคือ คุณสามารถคลิกขวาที่แถบเมนูได้ และ เลือก “Add to favourites” และ “Remove from favourites” ตามที่คุณต้องการ

รวมจำนวนหุ้นจาก หลักทรัพย์ 2 ตัว (Merging quotations of two symbols)

มันเป็นไปได้ที่บางครั้ง Ticker ของหลักทรัพย์อาจจะเปลี่ยนไปเมื่อคุณมี Ticker 2 ตัว ในฐานข้อมูลเดียวกัน ตัวหนึ่งอยู่ในราคาเก่าและอีกตัวอยู่ในราคาอันใหม่ (หลังจากที่เปลี่ยนชื่อแล้ว) เพื่อที่จะเอาราคาทั้งหมดใส่ไว้ใน Ticker ตัวเดียว คุณจำเป็นต้องใช้คำสั่ง Symbol > Merge คุณจำเป็นต้องเลือกเฉพาะ Ticker ตัวใหม่เท่านั้น (หลังจากที่เปลี่ยนชื่อแล้ว) และ เลือก Symbol > Merge เมื่อเสร็จแล้วคุณจะต้องเลือก Ticker อันเก่า (“merge with”) และเลือกตรวจสอบรายการต่อไปนี้

- ทำซ้ำและเขียนทับ - การเลือกรายการนี้จะมีการเขียนทับราคาที่อยู่ใน Ticker อันใหม่ด้วยการแสดงผลเป็น Ticker ตัวเก่า (ตามจริงแล้วไม่ควรเป็นแบบนี้ แต่ก็มีโอกาสเกิดได้)

- ลบหลังจากใช้ Merge with - เมื่อเลือกรายการนี้จะเป็นการลบ Ticker ตัวเก่าหลังจากรวมเข้าด้วยกันแล้ว
- ตั้งชื่อเป็นนามแฝง - ตัวเลือกนี้จะทำการคัดลอก Ticker ตัวเก่าเป็นชื่อแฝง และจะกลายเป็น Ticker ตัวใหม่

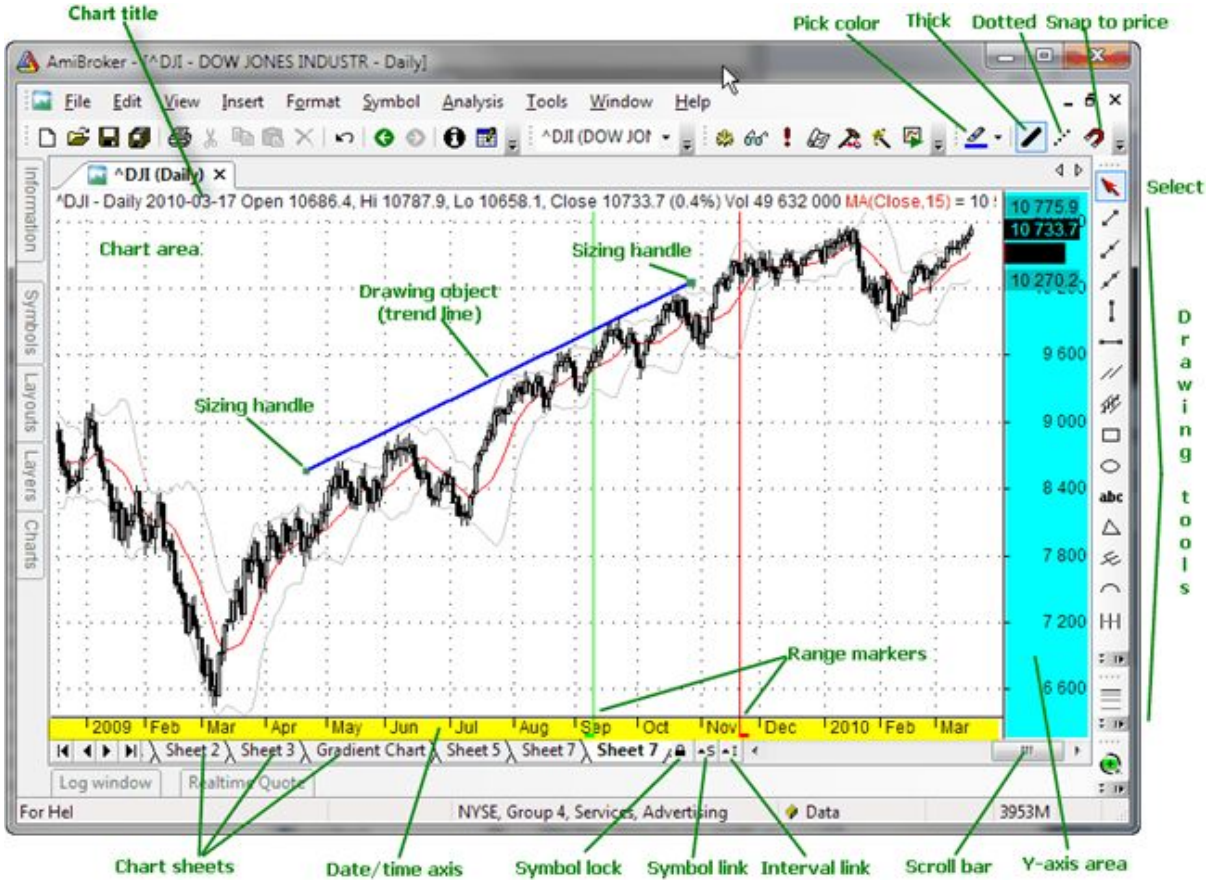
คู่มือการใช้ชาร์ทเบื้องต้น

(Beginners' charting guide)

เกริ่นนำ (**Introduction**)

เครื่องมือชาร์ทของ Amibroker นั้นจะมีเครื่องมือด้านการวาดต่างๆมากมาย พร้อมทั้งคุณสามารถย้าย, ย่อหรือขยาย, ตัด, คัดลอก, วาง และ ลบวัตถุได้โดยง่าย บทความนี้จะแนะนำคุณ ถึงเครื่องมือที่สำคัญต่างๆ สำหรับการใช้งาน

เรามาดูกันที่ User Interface กันก่อนดีกว่า



อย่างที่คุณเห็น ตรงกลางจะเป็นพื้นที่ของชาร์ท ซึ่งเป็นชาร์ทของราคาประกอบกับเส้นค่าเฉลี่ย และ Bollinger Band (คุณสามารถควบคุมการแสดงผลของชาร์ทได้จาก Tools > Preferences)

ด้านล่างของชาร์ท คุณจะเห็นแกน X ที่เป็นแกนของเวลา(วันที่) และ ด้านใต้จะเป็น Scroll Bar ซึ่งคุณสามารถที่จะเลื่อน Scroll Bar เพื่อดูราคาย้อนหลังได้ ในขณะที่แท็บของชีทๆ ใช้เพื่อดูชาร์ทที่แตกต่างกันในแต่ละหน้า (ขึ้นอยู่กับว่าคุณปรับแต่งชีทต่างๆของคุณอย่างไร)

ด้านขวา คุณจะเห็นแกน Y ที่แสดงถึงระดับของราคา โดยที่ราคาล่าสุดนั้นจะมีการเติมสีไว้ (ถ้าในตัวอย่างจะเห็นเป็นแถบสีดำ) เพื่อให้มองเห็นได้อย่างเด่นชัด บริเวณแกน Y นั้นยังสามารถใช้เพื่อย่อ หรือขยายชาร์ทในแนวตั้งได้ด้วย

ถัดมาอีกทางขวามือจะเป็นเครื่องมือที่ใช้ในการวาด ซึ่งคุณสามารถเลือกได้ว่าจะใช้เครื่องมือตัวไหน (มีแต่เครื่องมือยอดนิยมเท่านั้นที่จะโชว์ในแถบเมนูด้านขวานี้ ถ้าต้องการดูเครื่องมือทั้งหมดให้เลือกที่เมนู Insert)

"Select" (ลูกศรสีแดง) ใช้เพื่อเลือก, ย้าย, ย่อหรือ ขยาย วัตถุที่วาดไว้เรียบร้อยแล้ว หรือไว้เลือกแท่งราคาในชาร์ท

ในส่วนของแถบด้านบนคุณจะเห็นแถบเมนูการจัดรูปแบบที่ใช้เพื่อให้คุณสามารถปรับสี, สไตล์ (เช่น แบบหนา/แบบจุด) และ โหมด (snap to price หรือการวาดรูปโดยที่ให้ปลายจุดเข้ากับราคาได้อย่างง่าย) ได้

จากในภาพ คุณจะเห็นได้เลยว่าการตีเส้นเทรนด์ไลน์นั้น ทำได้โดยการลากจากจุดที่คุณต้องการไปสู่อีกจุดหนึ่ง การควบคุมเครื่องมือวาดเหลือนี้ ทำได้โดยการลาก หรือ การย่อขนาด ซึ่งเราจะอธิบายต่อจากนี้

การทำงานพื้นฐาน (Basic operations)

การเลื่อน (Scrolling)

การเลื่อนชาร์ทไปข้างหน้า/ข้างหลัง เพียงแค่เลื่อนปุ่ม Scroll Bar หรือใช้ปุ่มที่เป็นลูกศร "<" และ ">" ซึ่งอยู่ด้านซ้ายและขวาของ Scroll Bar โปรดทราบว่าการใช้ "< >" จะเป็นการเลื่อนชาร์ททีละหนึ่งแท่ง และการเลื่อนชาร์ทสามารถทำได้ด้วยการใช้ปุ่ม Wheel ที่อยู่บนเมาส์

การซูม (Zooming)

สำหรับการขยายชาร์ท (เพิ่มหรือลดแท่งที่แสดงบนหน้าจอ) คุณสามารถเข้าไปที่ View > Zoom หรือใช้ปุ่ม Zoom ที่อยู่แถบเมนู Tool Bar หรือจะใช้ปุ่ม Wheel ที่อยู่บนเมาส์ก็ได้ (ที่เป็น icon สีเหลือง และสีเขียวมีลักษณะเป็นวงกลม)

คุณยังสามารถซูมได้โดยการลากขอบซ้ายหรือขอบขวาของแท่ง Scroll Bar ถ้าทำตามนี้ การขยายชาร์ทจะเป็นการลดจำนวนแท่งที่แสดงผลบนหน้าจอ, แต่ถ้าย่อชาร์ท จะเป็นการเพิ่มจำนวนแท่งที่แสดงผล, การขยายภาพทั้งหมด (Zoom-all)จะปรากฏแท่งทั้งหมดตามที่มีข้อมูล, ขยายแบบธรรมดา (Zoom-normal) จะรีเซ็ตให้แสดงผลของแท่งข้อมูลตามที่ได้ตั้งค่าไว้ใน Tools > Preferences > Charting การซูมเข้า และออกนั้นสามารถทำได้โดยตรงผ่านการกดปุ่มในแถบ Tool Bar (ดูตัวอย่างได้จากรูปด้านล่าง) ส่วนการซูมด้วย

wheel ของเมาส์นั้นจะต้องกดปุ่ม CTRL ค้างไว้ด้วยถึงจะทำได้ คุณยังสามารถซูมได้ด้วยการเลือกช่วงที่ต้องการบนชาร์ท (จะกล่าวถึงภายหลังในหัวข้อ Marking Range)

ย่อ, ขยาย และขยับสเกลของแกน Y (Shrinking, expanding and moving Y-axis scale)

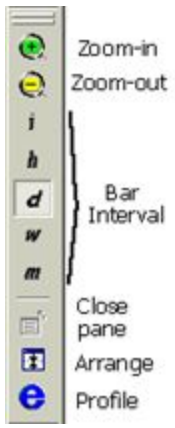
การที่จะ **ขยับ** สเกลของแกน Y ทำได้ด้วยการนำเมาส์ไปวางบริเวณแกน Y (พื้นที่สีฟ้าๆ ดังรูปภาพด้านบนอันแรก) และ จากนั้นคุณจะเห็นว่าลูกศรของเมาส์มันได้เปลี่ยนไปเป็นลูกศรขึ้น และ ลงเรียบร้อยแล้ว จากนั้นคุณก็เพียงแค่ คลิกและลากขึ้นลากลงตามที่ต้องการแล้วปล่อยในตำแหน่งที่คุณต้องการ

ย่อ / ขยาย แกน Y : กดปุ่ม SHIFT ค้างไว้และคลิกเมาส์ในบริเวณพื้นที่ของแกน Y จากนั้นก็เลื่อนเมาส์ขึ้นลง และปล่อย เมื่อคุณทำการเลื่อนเสร็จเรียบร้อย

ส่วนการ **Reset** สเกลของแกน Y ทำได้ง่ายๆ เพียงแค่ดับเบิ้ลคลิกบริเวณพื้นที่แกน Y เท่านั้นเอง

เปลี่ยนไทม์เฟรมของแท่ง (Changing bar interval (periodicity))

คุณสามารถสลับไทม์เฟรมอย่าง Daily / Weekly / Monthly และ Intraday ได้ง่ายๆ ด้วยการเลือกไทม์เฟรมที่ต้องการจากในเมนู View (ดูตัวอย่างได้ด้านล่าง)



ใน Toolbar จะใช้เครื่องหมายตามรูปภาพแทนที่ไทม์เฟรมแต่ละแบบ i - intraday, h - hourly, d - daily, w - weekly, m - monthly ตัว i นั้นเป็นตัวแทนของ Intraday ตาม “ฐานข้อมูล” ที่ถูกอ้างอิงไว้ใน File > Database Settings กราฟไทม์เฟรม Intraday จะยังคงมีอยู่ในเมนู View > Intraday อีกด้วย

การตั้งค่าไทม์เฟรมนั้นจะมีผลต่อหน้าต่างปัจจุบันเท่านั้น ดังนั้นแต่ละหน้าต่างสามารถใช้ไทม์เฟรมที่แตกต่างกันได้

หมายเหตุ : กราฟ Intraday นั้นจะไม่สามารถใช้ได้หากฐานข้อมูลของคุณเป็นแบบ End-of-Day โหมด Intraday จะใช้ได้เฉพาะฐานข้อมูลซึ่งมี “Base Time Interval” ที่อยู่ใน File > Database Settings ซึ่งจะถูกตั้งค่าไว้ด้วยข้อมูลที่เป็นภาพเล็กกว่า End-of-Day ตัวอย่างเช่น ถ้า Base Time Interval ถูกตั้งค่าไว้ที่ 5 นาที นั่นแปลชาร์ททั้งหมดจะสามารถใช้ได้เฉาะไทม์เฟรมที่มีขนาดใหญ่กว่า 5 นาทีขึ้นไป

ไทม์เฟรมที่มีให้จะมีดังต่อไปนี้

- Daily
- Weekly
- Monthly
- Hourly (intraday)
- 15-minute (intraday)
- 1-minute (intraday)
- 15-second (intraday RT only)
- 5-second (intraday RT only)
- Tick (intraday RT only)

คุณสามารถกำหนดแท่งแบบรายนาทีได้ทั้งหมด 5 แบบที่คุณต้องการว่าต้องการแท่งละกี่นาที และ ยังสามารถปรับกราฟแบบ Intraday ตามที่คุณต้องการได้อีก 5 แบบเช่นกัน ว่าต้องการแท่งละกี่ Tick เพียงเข้าไปที่ Tools > Preferences > Intraday ตัวเลือกที่ปรับแล้วจะเห็นได้เฉพาะผ่านทางการเข้า View > Intraday เท่านั้น

Selecing a quote

มันง่ายมากๆ ในการที่จะดูราคา หรือ ค่าของอินดิเคเตอร์ โดยการใช้โหมด Select ขั้นแรกให้ย้อนไปดูข้อมูลในอดีต และเลือกโหมดเป็น Select (ลูกศรสีแดงในแถบ Toolbar) จากนั้นคลิกพื้นที่บริเวณชาร์ท (แต่ต้องไม่โดนวัตถุที่วาดเอาไว้) เส้นแนวตั้งจะโชว์ขึ้นมาตรงที่ลูกศรเมาส์วางอยู่ ซึ่งตรงส่วนหัวข้อของชาร์ทนั้นจะโชว์ข้อมูลของแท่งที่เราเลือก ในหน้าต่างอินดิเคเตอร์ก็จะโชว์ค่าของมันในตำแหน่งเดียวกัน การเลื่อนทีละแท่งคุณสามารถทำได้ทั้งเลื่อนไปข้างหน้าหรือข้างหลัง ด้วยการใช้ปุ่มที่เป็นลูกศร “<” และ “>”

เมื่อต้องการจะเปลี่ยนโหมดการใช้งาน คุณสามารถคลิกที่เส้นแนวตั้งนั้นอีกที หรือคลิกที่แกนของวันที่ก็ได้ (มาร์กด้วยจุดสีแดงในรูปภาพก่อนหน้า) หรือคลิกตรงพื้นที่ว่างด้านขวา เมื่อทำดังนี้ จะเป็นการปิดโหมดนี้ และ ค่าที่แสดงอยู่ในส่วนหัวของชาร์ทจะเป็นค่าล่าสุดของแท่งข้อมูลล่าสุดแทน

*เลือกช่วง (**Marking range**)*

สำหรับการโชว์ส่วนที่ต้องการ เพียงแค่กดดับเบิ้ลคลิกบริเวณชาร์ทที่ต้องการให้เป็นจุดเริ่มต้น และดับเบิ้ลคลิกอีกครั้งในจุดที่ต้องการให้เป็นจุดสิ้นสุดของช่วง หรือคุณสามารถใช้ปุ่ม F12 ประกอบกับโหมด Select (ที่อธิบายอยู่ด้านบน) เพียงแค่เลือกแท่งของข้อมูลที่ต้องการและกด F12 เพื่อเลือกจุดเริ่มต้น และกดปุ่ม SHIFT+F12 เพื่อเลือกเป็นจุดสิ้นสุด คุณสามารถลบการเลือกช่วงที่ทำไว้ด้วยการกด CTRL+F12 หรือดับเบิ้ลคลิกในที่เดิมที่เรามาร์กไว้

การเลือกช่วงนั้นสามารถใช้เพื่อซูมดูช่วงที่เราเลือกได้ (View > Zoom > Range) และ ยังใช้เพื่อการคำนวณค่าที่เราเลือกด้วยคำสั่ง BeginValue และ EndValue ในฟังก์ชั่นของ AFL ได้อีกด้วย

*เพิ่ม / ปิดหน้าต่างชาร์ท (**Adding / closing chart panes**)*

ในแต่ละหน้าจะประกอบด้วยหน้าต่างหลายๆอัน ซึ่งแสดงชาร์ทหรืออินดิเคเตอร์ที่ต่างกันไป ในการที่จะใส่อินดิเคเตอร์ตัวใหม่เข้าไปในหน้าต่างของชาร์ท ก็เพียงแค่เลือกอินดิเคเตอร์ใน Chart list (ใช้เมนู **Window > Charts**) และดับเบิ้ลคลิกที่ชื่อของอินดิเคเตอร์ที่ต้องการ สำหรับข้อมูลเพิ่มเติม สามารถดูได้ในส่วนของ *การลากและวางชาร์ท*

ส่วนการจะปิดหน้าต่างชาร์ท ให้คลิกที่หน้าต่างนั้น จากนั้นเข้า **View > Pane > Close** ผ่านเมนูหลัก หรือ คลิกที่หน้าต่างนั้นๆ ด้วยคลิกขวา และเลือก **Close** ก็ทำได้เช่นกัน

เชื่อมโยงและล็อคชาร์ท (Linking and locking chart)

ชาร์ทหลายๆ หน้าต่าง (สามารถเปิดได้โดย File > New > Default Chart หรือ File > New > Blank Chart) สามารถเชื่อมโยงกันได้ Symbol-Linked จะแทนด้วยตัว s พิมพ์เล็ก และ Interval-Linked จะแทนด้วยตัว i พิมพ์เล็ก และ ทั้งสองปุ่มนี้อยู่ทางด้านซ้ายของแถบ Scroll Bar (ด่านล่าง) เมื่อคุณคลิกที่ปุ่มจะมีเมนูโชว์ขึ้นมาเป็นสีๆ เลือกสีที่ต้องการในหน้าต่างแรก และอีกหน้าต่างหนึ่งก็ให้เลือกสีเดียวกัน การเชื่อมโยงความหมายก็คือ ถ้าเราใช้ Symbol-Linked เมื่อเปลี่ยนชื่อสัญลักษณ์ในหน้าต่างชาร์ทแรก ชาร์ทที่สองก็จะเปลี่ยนชื่อตามด้วย ส่วน Interval-Linked เมื่อคุณคลิกที่แท่งข้อมูลไหนในหน้าต่างแรก หน้าต่างที่สองก็จะคลิกที่แท่งข้อมูลเดียวกัน

คุณสามารถป้องกันไม่ให้สัญลักษณ์เปลี่ยนชื่อไปตามอีกหน้าต่างที่เราเปลี่ยนชื่อได้ ด้วยการกดที่เครื่องหมายแม่กุญแจอันเล็กๆ ที่ฝั่งซ้ายของ Scroll Bar (เลือก Symbol Lock) เมื่อกดแล้วชื่อสัญลักษณ์ของชาร์ทหน้าต่างนี้จะไม่เปลี่ยนตามชาร์ทอันแรก จนกว่าคุณจะกดปุ่ม Symbol Lock อีกครั้ง

การใช้เครื่องมือวาด (***Using drawing tools***)

Amibroker นั้นมีเครื่องมือในการวาดที่หลากหลายมาก ดังนี้

- trend line
- ray (ตั้งแต่เวอร์ชั่น 4.20)
- extended line (ตั้งแต่เวอร์ชั่น 4.20)
- vertical line
- horizontal line
- parallel lines (ตั้งแต่เวอร์ชั่น 4.20)
- Regression channels: Raff, standard deviation, standard error (ตั้งแต่เวอร์ชั่น 4.20)
- Fibonacci Retracement study (ตั้งแต่เวอร์ชั่น 4.20)
- Fibonacci Time zones study
- Fibonacci Fan
- Fibonacci arc

- Gann Square (ตั้งแต่เวอร์ชั่น 4.20)
- Gann Fan (ตั้งแต่เวอร์ชั่น 4.20)
- Ellipse tool
- Arc tool
- Rectangle
- text box tool

ทั้งหมดนี้จะปรากฏในเมนู Insert และเมนู Draw ใน Toolbar แต่ละวัตถุที่วาดแล้วจะสามารถย้าย, ปรับขนาด, คัดลอก, ลบและปรับได้หลังจากที่วาดแล้ว

ในการวาดวัตถุบนชาร์ท ก่อนอื่นคุณต้องเลือกเครื่องมือที่ต้องการจะวาด และ เริ่มวาดด้วยการใช้เมาส์คลิกตรงจุดที่ต้องการให้เริ่มต้น คลิกซ้ายค้างไว้ ทันทีที่ลากเมาส์ เครื่องมือที่เราเลือกจะปรากฏให้เห็น เมื่อต้องการลากถึงตรงไหนก็ให้ปล่อยตรงจุดนั้น คุณสามารถยกเลิกการวาดได้ด้วยการกดปุ่ม ESC

ถ้าคุณลากเมาส์ไปบริเวณวัตถุที่คุณวาด คุณจะเห็นลูกศรเมาส์เปลี่ยนไปเป็นรูปอื่น ซึ่งความหมายของรูปแต่ละแบบก็ คือ

ถ้าลูกศรไปอยู่ใกล้ขอบของวัตถุ จะเปลี่ยนเป็นรูปนี้ เรียกว่า Sizing Pointer เอาไว้ย่อขยายวัตถุได้

ถ้าลูกศรไปอยู่ในบริเวณวัตถุ จะเปลี่ยนเป็นรูปนี้ เรียกว่า Moving Pointer เอาไว้เคลื่อนย้ายวัตถุได้ ทุกๆ วัตถุที่วาดจะสามารถ ย้าย ปรับขนาด ลบ และคัดลอกได้

การเลือกวัตถุใดๆ นั้นสามารถทำได้ง่ายๆ โดยเพียงแค่ใช้เมาส์เลื่อนไปบนวัตถุ เมื่อ Moving Pointer ปรากฏขึ้นให้คลิกหนึ่งครั้ง วัตถุนี้นั้นก็จะถูกเลือก และ ปุ่มที่ใช้สำหรับปรับขนาดก็จะปรากฏขึ้นมาด้วย (ลักษณะเป็นปุ่มสีเหลี่ยมเล็กๆ)

ส่วนวิธียกเลิกการเลือกวัตถุนั้น ก็เพียงคลิกที่พื้นที่ว่างของชาร์ท

การย้ายวัตถุก็ให้คลิกเลือกส่วนใดก็ได้ของวัตถุนั้น กดค้างไว้ แล้วย้ายไปในจุดที่ต้องการ

การลบวัตถุ ให้เลือกวัตถุที่ต้องการแล้วกดปุ่ม DEL ที่อยู่บนคีย์บอร์ดหรือใช้ Edit > Delete ในเมนู หรือจะใช้ปุ่ม Delete ที่อยู่ใน Toolbar ก็ได้

การคัดลอกวัตถุ เลือกวัตถุที่ต้องการแล้วกด Ctrl+C หรือใช้ Edit > Copy หรือใช้ปุ่ม Copy ใน Toolbar

การตัดวัตถุ เช่นเคยให้เลือกวัตถุที่ต้องการจะตัด แล้วกด Ctrl+X หรือใช้เมนู Edit > Cut หรือใช้ปุ่ม Cut ใน Toolbar

การวางวัตถุ ทำได้โดยการกด Ctrl+V หรือใช้เมนู Edit > Paste หรือ จะใช้ปุ่มวางใน Toolbar ก็ได้ เมื่อวางวัตถุแล้ว วัตถุชิ้นใหม่จะวางทับชิ้นเก่าอย่างพอดิบพอดี และ ถูกเลือกไว้โดยอัตโนมัติ ดังนั้น คุณจึงสามารถย้ายวัตถุชิ้นใหม่ไปตรงไหนก็ได้

การนำสีหรือรูปแบบไปใช้ กับวัตถุทีเราเลือกไว้ สามารถทำได้ผ่านเมนู Format หรือใช้ปุ่ม Format ใน Toolbar เพื่อเปลี่ยนสี, ความหนา, จุดหรือปรับสไตล์ของข้อมูลราคาได้ เพิ่มเติมก็คือคุณสามารถเลือกสีหรือ สไตล์ก่อนที่จะวาดวัตถุชิ้นใหม่

การปรับแต่งคุณสมบัติ ของวัตถุชิ้นใดๆ คุณสามารถใช้วิธีดับเบิ้ลคลิกหรือใช้เมนู **Edit > Properties** หรือ ใช้ปุ่ม **Alt+ENTER** ก็ได้

การลบทั้งหมด ทำได้ด้วยการกดเมนู **Edit > Delete All**

ข้อมูลเพิ่มเติม

ถ้าอยากรู้รายละเอียดเพิ่มเติมเกี่ยวกับเครื่องมือการวาด โปรดอ่านในบทของ Drawing tools reference

วิธีใช้งานการลากและวางชาร์ท(Indicator)

(How to use drag-and-drop charting interface)

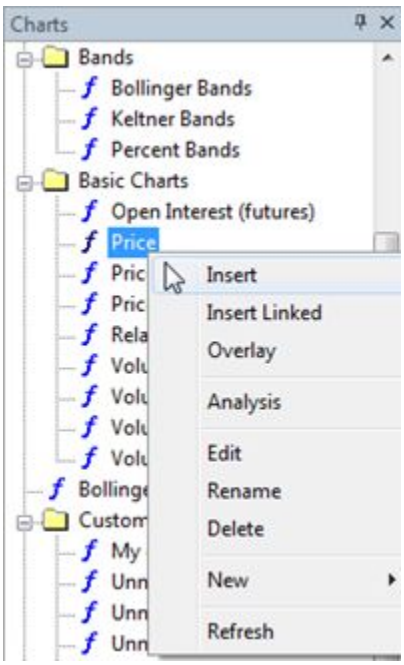
เกริ่นนำ (Introduction)

Amibroker ให้คุณสามารถสร้างและปรับอินดิเคเตอร์โดยเพียงแค่การคลิกเมาส์ไม่กี่ครั้ง ในตอนนี้คุณสามารถสร้างอินดิเคเตอร์ที่ซับซ้อนได้โดยไม่ต้องมีความรู้เรื่องการโปรแกรมมิ่งเลย ซึ่งอินดิเคเตอร์ทั้งหมดที่พร้อมใช้งานจะถูกบันทึกไว้ในแท็บ Charts ที่อยู่ในส่วนของหน้าต่าง Workspace

วิดีโอสาธิตการใช้งานจะอยู่ในลิงค์นี้ http://www.amibroker.net/video/dragdrop1.html ที่จะสอนเกี่ยวกับการใช้งานเบื้องต้นของฟังก์ชั่นลากและวาง

จะใส่อินดิเคเตอร์ใหม่ได้อย่างไร (How to insert a new indicator)

การที่จะใส่อินดิเคเตอร์ตัวใหม่แยกอีกหน้าต่าง สามารถทำได้โดยการเลือกอินดิเคเตอร์ในแท็บ Chart (ใช้ Window > Charts) และ ดับเบิ้ลคลิกที่ชื่อของอินดิเคเตอร์นั้นๆ



หรือ อีกทางเลือกหนึ่ง คุณสามารถคลิกขวาแล้วเลือก Insert จากเมนูที่ปรากฏขึ้นมาก็ได้ เมื่อทำแล้วอินดิเคเตอร์หน้าต่างใหม่จะถูกสร้างขึ้นมา และ ค่าต่างๆก็จะแสดงขึ้นมาด้วย คุณสามารถปรับค่าของอินดิเคเตอร์ได้เช่นกัน (เช่น สีหรือระยะเวลาที่ใช้คำนวณ) เมื่อปรับแต่งเป็นที่เรียบร้อยแล้วให้กด OK (คุณจะเจอรายละเอียดเพิ่มเติมเกี่ยวกับหน้าต่างพารามิเตอร์ได้ที่ด้านล่าง)

ตัวอย่าง: สมมติต้องการเพิ่มหน้าต่าง RSI เข้าไป ให้หาชื่ออินดิเคเตอร์ RSI จากในลิสต์ ดับเบิ้ลคลิกที่ชื่อ เลือกระยะเวลาที่ต้องการให้คำนวณและเลือกสี จากนั้นกดโอเค เป็นอันเสร็จ

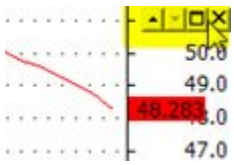
จะทำยังไงให้อินดิเคเตอร์อันนึงวางซ้อนบนอีกอันนึงได้ (How to overlay one indicator on another indicator)

ในการที่จะวางอินดิเคเตอร์ซ้อนกัน ให้กดคลิกซ้ายค้างไว้บนชื่ออินดิเคเตอร์ที่เราต้องการ แล้วลากมันมาใส่ในหน้าต่างอินดิเคเตอร์อีกอัน จากนั้นก็ปล่อย

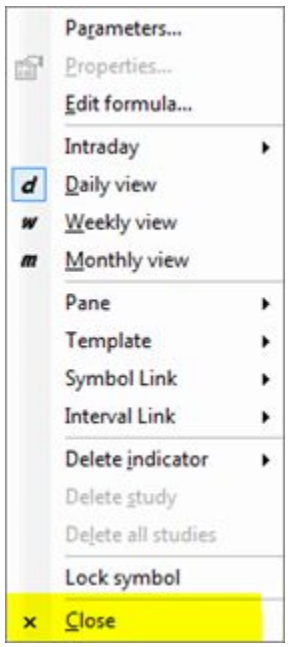
ตัวอย่าง: จากตัวอย่างเดิม เราต้องการใส่ RSI เข้าไปอีกเส้น (ด้วยระยะเวลาที่ต่างจากเดิม) ไว้ในหน้าต่างอันแรก สิ่งที่ต้องทำก็คือลากชื่อ RSI เข้าไปใส่ในหน้าต่างที่เราสร้างก่อนหน้านี้ เปลี่ยนระยะเวลาที่ใช้คำนวณในหน้าต่างพารามิเตอร์ แล้วกด OK หรือ อีกทางเลือกคือ คลิกขวาที่ชื่ออินดิเคเตอร์นั้น แล้วเลือก Overlay

จะลบอินดิเคเตอร์อย่างไร (How to delete the indicator)

เมื่อต้องการเอาอินดิเคเตอร์ออก ให้กดปุ่ม ปิด จากเมนูที่อยู่ตรงมุมขวาบนของหน้าต่างอินดิเคเตอร์นั้น (เมนูจะปรากฏขึ้นเมื่อคุณวางเมาส์ไว้บริเวณนั้น) เมนูนี้จะยอมให้ย้ายหน้าต่างขึ้น ลง หรือขยายให้ใหญ่สุดก็ได้ (เพื่อความเข้าใจให้ดูภาพด้านล่างประกอบ)

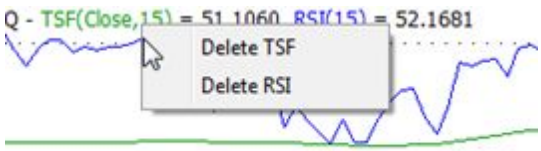


คุณยังสามารถใช้คำสั่งปิดจากเมนูที่จะปรากฏขึ้นเมื่อคุณคลิกขวาที่หน้าต่างของชาร์ทได้อีกด้วย



จะลบอินดิเคเตอร์ในหน้าต่างนั้นออกไปอย่างไร (How to remove the indicator plot from the pane)

หากต้องการลบอินดิเคเตอร์ตัวใดตัวหนึ่งออกจากหน้าต่างนั้น ให้คลิกขวาตรงที่ชื่อของหน้าต่าง (ใกล้กับบนสุดของหน้าต่างชาร์ท) และเลือกอินดิเคเตอร์ที่คุณต้องการจะลบ



คุณยังสามารถลบอินดิเคเตอร์ได้ด้วยการเลือก Delete Indicator ในเมนูที่ปรากฏขึ้นเมื่อคลิกขวาในหน้าต่างชาร์ท

จะเปลี่ยนค่าพารามิเตอร์ สี และสไตล์ของอินดิเคเตอร์อย่างไร (How to change parameters/colors/styles of indicators)

หน้าต่างพารามิเตอร์นั้นสามารถเปลี่ยนค่าต่างๆ รวมถึงสี และสไตล์ของอินดิเคเตอร์ของคุณได้ หน้าต่างพารามิเตอร์จะปรากฏขึ้นมาเมื่อคุณแทรกอินดิเคเตอร์ใหม่เข้าไป คุณยังสามารถเข้าไปปรับแต่งได้ด้วยการคลิกขวาที่หน้าต่างชาร์ทและเลือก Parameters จากเมนูที่เด้งขึ้นมา หน้าต่างพารามิเตอร์จะแสดงค่าทั้งหมดที่ประกอบอยู่ใน AFL โค้ด (ดังนั้นผู้ใช้จึงสามารถปรับค่าได้) ถึงแม้ว่าอินดิเคเตอร์แต่ละตัวจะมีรายละเอียดต่างกัน แต่ส่วนใหญ่แล้วสิ่งที่คุณจะเห็นก็มีดังต่อไปนี้

ฟิลด์ราคา - คือข้อมูลที่ใช้เพื่อคำนวณในการสร้างอินดิเคเตอร์ ถ้าฟิลด์ราคาเป็น Close แปลว่าอินดิเคเตอร์ตัวนั้นจะใช้การคำนวณจากราคาปิด ฟิลด์ราคาจะไม่รองรับในทุกอินดิเคเตอร์ เนื่องจากไม่ใช่เครื่องมือทุกตัวที่จะยอมให้คุณเปลี่ยนฟิลด์ราคาได้ (เช่น ADLine)

ระยะเวลา - ใช้ระบุว่าต้องการให้ระยะเวลาในการคำนวณเป็นเท่าไหร่

สี - เอาไว้ระบุและเปลี่ยนสีของอินดิเคเตอร์

สไตล์ - อนุญาตให้คุณเลือกสไตล์ของการแสดงผลได้ (รายละเอียดของสไตล์แบบต่างๆ จะอยู่ในส่วนของ วิธีการใช้สไตล์กราฟและสี)

จะใส่อินดิเคเตอร์ที่มีสเกลต่างกันซ้อนกันได้อย่างไร (How to overlay indicators with different scales)

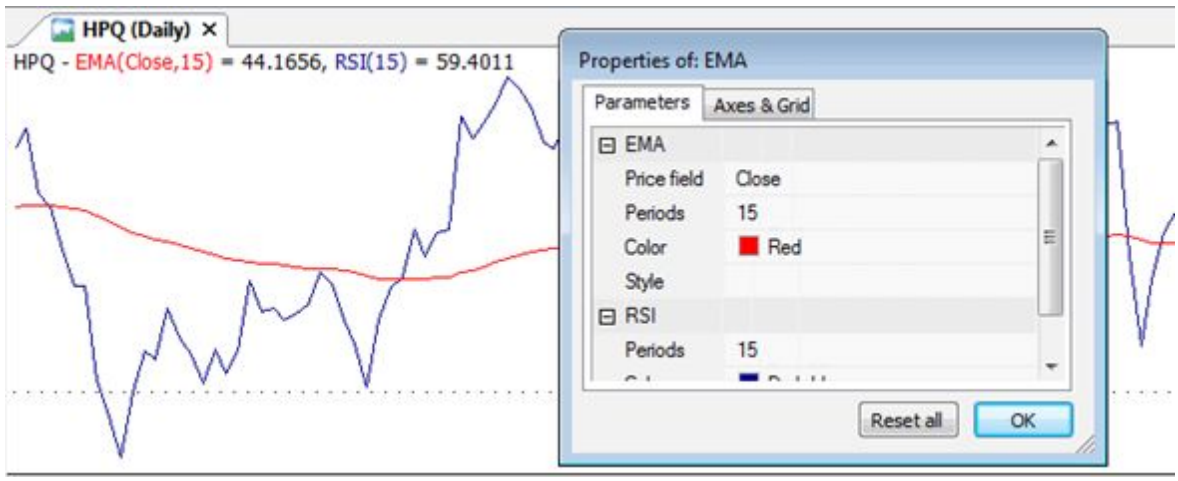
หากในหนึ่งหน้าต่างจะมีอินดิเคเตอร์หลายตัวที่มีสเกลต่างกัน ให้ลากอินดิเคเตอร์ตัวที่สองไปยังอินดิเคเตอร์ตัวแรก ในหน้าต่างพารามิเตอร์ ในส่วนของ Style นั้นให้ติ๊กถูกที่คำว่า StyleOwnScale

ตัวอย่าง: ลาก OBV (On Balance Volume) เข้าไปใส่ในหน้าต่าง RSI เมื่อเราระบุให้มันแสดงผลแบบ StyleOwnScale ผลลัพธ์ที่ได้ก็คืออินดิเคเตอร์ที่แสดงผลออกมาอย่างถูกต้อง

จะสร้างอินดิเคเตอร์ใหม่โดยอ้างอิงบนอินดิเคเตอร์อื่นอย่างไร(How to create an indicator based on another indicator)

AmiBroker สามารถให้คุณสร้างอินดิเคเตอร์ที่อ้างอิงค่าจากอินดิเคเตอร์ตัวอื่นได้โดยง่าย ที่คุณต้องทำทั้งหมดก็คือ คลิกซ้ายค้างไว้บนชื่ออินดิเคเตอร์ที่ต้องการ ลากมันมาใส่ใน หน้าต่างอินดิเคเตอร์ปลายทาง แล้วปล่อย จากนั้นอินดิเคเตอร์ตัวใหม่ก็จะปรากฏในหน้าต่างอินดิเคเตอร์อันเดิม ในหน้าต่างพารามิเตอร์ ในส่วนของ ฟิลด์ราคา (Price field) จะเลือกค่าของอินดิเคเตอร์อันเดิมมาใช้ในการคำนวณ

ตัวอย่าง: ในการคำนวณเส้น Simple Moving Average จากข้อมูลในอดีตของ RSI ก็เพียงแค่ลากอินดิเคเตอร์ MA เข้าไปใส่ในหน้าต่างของ RSI ในส่วนของฟิลด์ราคาก็จะถูกกำหนดไว้ดังเดิม ดังนั้น เส้นค่าเฉลี่ยที่เราเพิ่งเพิ่มเข้าไป มันก็จะคำนวณมาจากค่าของเส้น RSI(15) ดังรูป



หมายเหตุ : เนื้อหาในส่วนด้านล่างจะประกอบไปด้วยข้อมูลเชิงเทคนิคสำหรับผู้ใช้แบบขั้นสูง ผู้ที่เพิ่งเริ่มต้นอาจข้ามส่วนนี้ไปก็ได้

การใช้คำสั่ง Param(), ParamColor(), ParamToggle(), ParamStyle() (Using Param(), ParamColor(), ParamToggle(), ParamStyle() functions)

ฟังก์ชั่นเหล่านี้เมื่อใช้งานจะสามารถทำให้เปลี่ยนการตั้งค่าของอินดิเคเตอร์ได้โดยตรง แทนที่การตั้งค่าผ่านหน้าต่างพารามิเตอร์

Param("name", defvalue, min = 0, max = 100, step = 1, sincr = 0)

วิธีการเพิ่มหรือปรับค่าพารามิเตอร์ต่างๆ สามารถเข้าถึงได้ด้วยขั้นตอนต่อไปนี้

"name" - คือ พารามิเตอร์ที่ให้เรากำหนดได้ว่าอยากให้ชื่อที่แสดงเป็นชื่ออะไร

defvalue - กำหนดค่าพื้นฐานของพารามิเตอร์นั้น

min, max - กำหนดค่าต่ำสุด และ สูงสุดของพารามิเตอร์

step - ระบุได้ว่าต้องการให้พารามิเตอร์ขยับเพิ่มทีละเท่าไหร่จากการเลื่อนแต่ละครั้ง

sincr - เป็นการระบุว่าต้องการให้อินดิเคเตอร์อีกอันหนึ่งที่เพิ่มเข้ามาตอนหลัง มีค่าเพิ่มขึ้นอีกเท่าไหร่จากอินดิเคเตอร์อันแรกซึ่งเป็นชนิดเดียวกัน ตัวอย่างเช่น ถ้าคุณใส่เส้นค่าเฉลี่ยสองเส้นไว้ในหน้าต่างเดียวกัน ถ้าเส้นแรกใช้ Period อยู่ที่ 15 อีกเส้นหนึ่งก็จะมีค่าอยู่ที่ 25 (defvalue = 15 + sincr = 10)

ParamColor("name", defaultcolor)

"name" - คือพารามิเตอร์ที่ให้เรากำหนดได้ว่าอยากให้ชื่อที่แสดงเป็นชื่ออะไร

defaultcolor - กำหนดสีที่ต้องการให้ปรากฏ

ฟังก์ชั่น ParamColor อนุญาตให้คุณใช้ ColorCycle เป็นสีพื้นฐานได้ เมื่อคุณใช้ ColorCycle สีพื้นฐานนั้นจะไล่สีไปตั้งแต่ แดง, เขียว, เขียวขุ่น, ทอง, ม่วง, เขียวอ่อน, ดำเข้ม ตามลำดับ ของอินดิเคเตอร์ที่คุณใส่เข้าไปในหน้าต่างเดียวกัน

ParamStyle("name", defaultval = styleLine, mask = maskDefault)

เป็นฟังก์ชั่นที่อนุญาตให้คุณเลือกรูปแบบการพล๊อตกราฟได้ นอกเหนือไปจากเวอร์ชั่นก่อนหน้าแล้ว ยังมีอีกสองสไตล์ที่เพิ่มเข้ามา ได้แก่

styleHidden - เป็นการผสมระหว่าง styleNoDraw และ styleNoRescale
styleDashed - เส้นประ

ส่วนลิสต์ต่อไปนี้คือสไตล์ที่จะแสดงในหน้าต่างของพารามิเตอร์ ซึ่งขึ้นอยู่กับค่า mask ที่คุณกำหนด

maskDefault - จะโชว์แบบหนา, แบบเส้นประ, ซ่อน และ แบบที่ใช้ขนาดของตัวเอง (OwnScale) (นี่เป็นค่ามาตรฐานที่ถูกตั้งค่าไว้ของ ParamStyle)
maskAll - แสดงสไตล์ทั้งหมด
maskPrice - แสดงแบบหนา, แบบซ่อน, ขนาดของตัวเอง, แท่งเทียน, และแบบแท่ง
maskHistogram - แสดงฮิสโทแกรม, แบบเส้นหนา, ซ่อน, ขนาดของตัวเอง, แบบพื้นที่

ParamField("name", field = 3) - คำสั่งนี้จะยินยอมให้คุณสามารถเลือกฟิลด์ของราคาอันใดก็ตามมาคำนวณในอินดิเคเตอร์ได้ (ฟิลด์จะถูกใช้ในการคำนวณค่าของอินดิเคเตอร์) ฟังก์ชั่นนี้จะเป็นการระบุข้อมูลที่เราต้องการใช้ ซึ่งค่า Default value = 3 จะเป็นการใช้ราคาปิดมาคำนวณ และ ส่วนใหญ่ค่าของฟิลด์จะเป็นดังนี้

-1 - ตัว ParamField จะส่งค่าของอินดิเคเตอร์ตัวแรกกลับมา หรือ เป็นราคาปิดแทนหากไม่มีอินดิเคเตอร์ก่อนหน้า
0 - ส่งกลับ **ราคาเปิด**
1 - ส่งกลับ **ราคาสูงสุด**
2 - ส่งกลับ **ราคาต่ำสุด**
3 - ส่งกลับ **ราคาปิด** (ค่ามาตรฐาน)
4 - ส่งกลับ **ราคาเฉลี่ย** = (ราคาสูงสุด + ราคาต่ำสุด + ราคาปิด) / 3
5 - ส่งกลับ **ปริมาณการซื้อขาย**
6 - ส่งกลับ **สถานะคงค้าง**

7, 8, 9, ... - ส่งกลับค่าของอินดิเคเตอร์ที่ใส่ไว้ในหน้าต่างนั้น

ParamToggle("name", "values", defaultval = 0) - ฟังก์ชั่นนี้จะยินยอมให้คุณใช้คำสั่งบูลีน (Yes/No) ในพารามิเตอร์ได้

"name" - คือชื่อของพารามิเตอร์

"values" - ค่าของพารามิเตอร์ (คั่นด้วยตัว | ยกตัวอย่างเช่น "No|Yes" - ข้อความแรกจะเป็นค่า False และข้อความสองจะเป็นค่า True)

defaultval - ค่ามาตรฐานของพารามิเตอร์

ยกตัวอย่าง : อินดิเคเตอร์ที่อยู่ด้านล่างนี้ จะแสดงให้คุณเห็นว่าโค้ดที่เขียนขึ้นมานั้น จะทำให้พารามิเตอร์ทำงานอย่างไร คุณสามารถเปลี่ยนการตั้งค่าได้จากหน้าต่างของพารามิเตอร์

```
Buy = Cross(MACD(), Signal() );
Sell = Cross(Signal(), MACD() );

pricefield = ParamField("Price Field", 2);
Color = ParamColor("color",colorRed);
style = ParamStyle("style",styleLine,maskAll);
arrows = ParamToggle("Display arrows", "No|Yes",0);
Plot(pricefield,"My Indicator",Color,style);

if(arrows)
{
	PlotShapes(Buy*shapeUpArrow+Sell*shapeDownArrow,IIf(Buy,colorGreen,color
Red) );
}
```

คำสั่งพิเศษ : อธิบายคำสั่ง SECTION_BEGIN, _SECTION_END, _SECTION_NAME, _DEFAULT_NAME, _PARAM_VALUES (สำหรับผู้ใช้ขั้นสูงเท่านั้น)

เมื่อคุณลากสูตรวางในหน้าต่างชาร์ท ตัว AmiBroker เองจะใส่โค้ด _SECTION_BEGIN("name") และ _SECTION_END() ไว้ที่หัวและท้ายของโค้ดตามลำดับโดยอัตโนมัติ

ดังนั้น จากโค้ดเดิมที่เป็นแบบนี้

```
P = ParamField("Price field",-1);
Periods = Param("Periods", 15, 2, 200, 1, 10 );
Plot( MA( P, Periods ), _DEFAULT_NAME(), ParamColor( "Color",colorCycle ),
ParamStyle("Style") );
```

ก็จะถูกเปลี่ยนโดย AmiBroker ให้เป็นแบบนี้

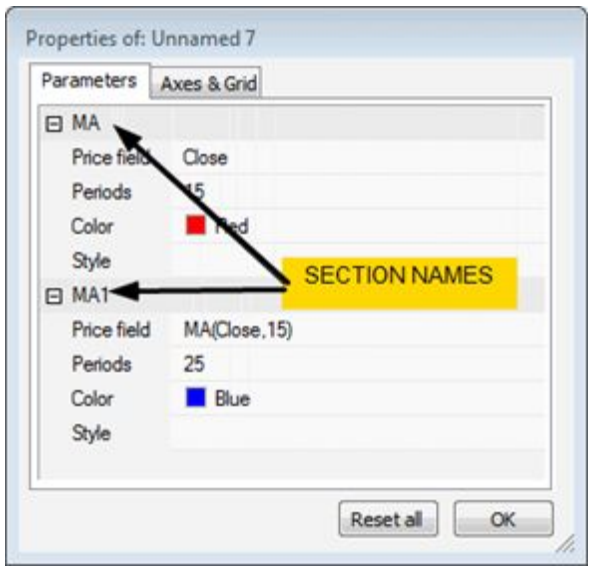
```
_SECTION_BEGIN("MA");
P = ParamField("Price field",-1);
Periods = Param("Periods", 15, 2, 200, 1, 10 );
Plot( MA( P, Periods ), _DEFAULT_NAME(), ParamColor("Color", colorCycle ),
ParamStyle("Style") );
_SECTION_END();
```

AmiBroker อนุญาตให้คุณแยก _SECTION_BEGIN/_SECTION_END ในแต่ละคำสั่ง และ ปรับแต่งมันในภายหลังได้

นอกจากนี้ โปรดแน่ใจว่าคุณใส่มันอย่างถูกต้อง การใช้ชื่อเดียวกัน ในพารามิเตอร์นั้นจะส่งผลให้โค้ดบางส่วนถูกรบกวนซึ่งกันและกันได้ ยกตัวอย่างเช่น ถ้าคุณลากเส้นค่าเฉลี่ยใส่ลงไปสองเส้น โค้ดจะเป็นดังนี้

```
_SECTION_BEGIN("MA");
P = ParamField("Price field",-1);
Periods = Param("Periods", 15, 2, 200, 1, 10 );
Plot( MA( P, Periods ), _DEFAULT_NAME(), ParamColor( "Color", colorCycle ),
ParamStyle("Style") );
_SECTION_END();

_SECTION_BEGIN("MA1");
P = ParamField("Price field",-1);
Periods = Param("Periods", 15, 2, 200, 1, 10 );
Plot( MA( P, Periods ), _DEFAULT_NAME(), ParamColor( "Color", colorCycle ),
ParamStyle("Style") );
_SECTION_END();
```

เพิ่มเติม ชื่อพารามิเตอร์ในสองส่วนนั้นสามารถตั้งให้เหมือนกันได้ โดยที่มันจะไม่ทำงานรบกวนกัน ต้องขอบคุณชื่อของแต่ละส่วนที่ทำให้ไม่เกิดปัญหานี้ นั่นเพราะว่า AmiBroker จะระบุแต่ละส่วนด้วยชื่อของมัน ดังนั้นแล้ว หากชื่อของแต่ละส่วนไม่เหมือนกัน พารามิเตอร์ที่อยู่ในส่วนนั้นก็จะไม่ซ้ำกับที่อยู่ในส่วนอื่น เมื่อคุณลากอินดิเคเตอร์ตัวใหม่เข้ามาใส่ AmiBroker จะตรวจสอบโดยอัตโนมัติ และ จะกำหนดหมายเลขให้หากชื่อในส่วนทั้งสองนั้นซ้ำกัน ชื่อของส่วนนั้นจะปรากฏดังรูป



สุดท้ายแต่ไม่ท้ายสุด : คุณไม่ควรลบ _SECTION_BEGIN / _SECTION_END ออกจากสูตร ถ้าคุณทำแบบนั้น AmiBroker จะไม่สามารถรับรู้ถึงพารามิเตอร์ และ สิ่งอื่นที่อยู่ในส่วนนั้นได้ และจะทำให้รบกวนการทำงานของส่วนอื่นๆ ด้วย

_SECTION_NAME คือฟังก์ชั่นที่ใช้กำหนดชื่อของฟังก์ชั่น (ใส่ต่อจากคำสั่ง _SECTION_BEGIN)

_DEFAULT_NAME เป็นคำสั่งที่ใช้กำหนดชื่อเริ่มต้นของรูปแบบการพล๊อต ชื่อเริ่มต้นนั้นประกอบด้วยชื่อของส่วนต่างๆ และคั่นด้วยเครื่องหมายจุลภาค ยกตัวอย่างดังโค้ดนี้

```
_SECTION_BEGIN("MA1");
P = ParamField("Price field");
Periods = Param("Periods", 15, 2, 200, 1, 10 );
Plot( MA( P, Periods ), _DEFAULT_NAME(), ParamColor( "Color", colorCycle ),
ParamStyle("Style") );
_SECTION_END();
```

_DEFAULT_NAME จะถูกกำหนดเป็น "MA1(Close,15)"

_PARAM_VALUES ทำงานเหมือนกับ _DEFAULT_NAME ยกเว้นเพียงแต่จะไม่รวมชื่อเข้าไปด้วย (มีเพียงแต่ค่าของพารามิเตอร์เท่านั้น) ดังนั้นแล้วจากตัวอย่างข้างบน _PARAM_VALUES จะถูกกำหนดค่าเป็น "(Close, 15)"

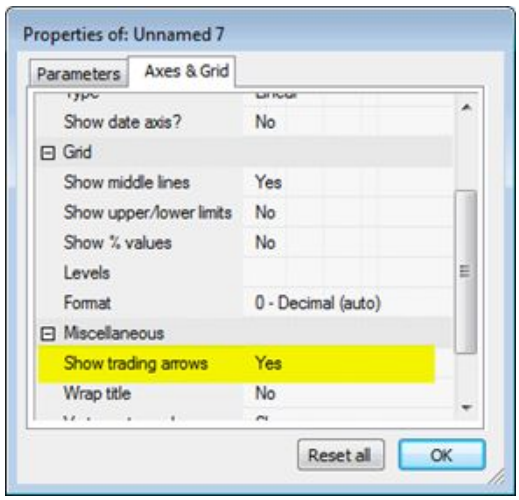
คำถามที่พบบ่อยเกี่ยวกับฟังก์ชั่นลากและวาง (Frequently Asked Questions about drag & drop functionality)

คำถาม : อะไรคือความต่างระหว่างคำสั่งแทรกและแทรกแบบเชื่อมโยงจากตัวเลือกในเมนูชาร์ท?

คำตอบ : คำสั่ง แทรก (Insert) จะเป็นการทำสำเนาจากโค้ดต้นฉบับแล้ววางไว้ในโฟลเดอร์ drag-drop ซึ่งถูกซ่อนเอาไว้ และจะไม่ได้รับผลกระทบจากการเปลี่ยนแปลงใดๆ ที่ทำบนหน้า Overlay ของชาร์ท การดับเบิ้ลคลิกบนชื่อสูตรที่อยู่ในหน้าต่างด้านซ้ายจะเท่ากับการเลือกคำสั่ง แทรก จากในเมนู ในอีกด้านหนึ่ง คำสั่ง แทรกแบบเชื่อมโยง (Insert Linked) จะไม่สร้างสำเนาของสูตรไว้ การแทรกมันเข้าไปใน หน้าต่างชาร์ทจะเป็นการเชื่อมโยงกับสูตรต้นฉบับโดยตรง ด้วยวิธีนี้เมื่อมีการแก้ไขค่าต่างๆ ในหน้า Overlay จะทำให้โค้ดต้นฉบับเปลี่ยนไปด้วย

คำถาม : ฉันไม่เห็นลูกศร ซื้อ/ขาย ในระบบของฉันเลย

คำตอบ : ลูกศรซื้อขายจะปรากฏในหน้าต่างชาร์ททุกหน้า (ยกเว้นชาร์ทราคาในตัว) อย่างไรก็ตาม โดยมาตรฐานแล้วลูกศรจะถูกปิดเอาไว้ไม่ให้แสดงผล ในการเปิดใช้งานคุณต้องเปิดผ่านหน้าต่างพารามิเตอร์ เลือกที่แท็บ “Axes & Grid” และเลือกเมนู “Show trading arrows” แล้วเปลี่ยนตัวเลือกเป็น “Yes”



คำถาม : ข้อความที่ว่า "Automatic Analysis formula window is now drag&drop target too (you can drag formulas and AFL files onto it)" หมายความว่าอย่างไร?

คำตอบ : หมายความว่าคุณสามารถลากสูตรจากหน้าต่างด้านซ้ายก็ได้ หรือเปิดจากไฟล์ .AFL ผ่าน Windows Explorer และ ลากมาใส่ในหน้าต่างการวิเคราะห์แบบอัตโนมัติ (Automatic Analysis : AA) ซึ่งจะถูกโหลดใส่ไว้ในหน้าต่าง AA นี่จะทำให้คุณสามารถโหลดสูตรผ่านการกดปุ่ม "Load" ในหน้าต่าง AA ได้อีกทางหนึ่ง

คำถาม : ฉันสามารถสร้างทางลัดในหน้าต่างสูตรได้หรือไม่?

คำตอบ : ทำไม่ได้ คุณสามารถทำได้เพียงการลากและวางไฟล์โดยส่วนขยายของ .AFL เท่านั้น (เมนูลัดใน Windows มีส่วนขยาย)

คำถาม : ฉันสามารถเพิ่มสูตรของตัวเองในเมนูที่อยู่หน้าต่างด้านซ้ายได้หรือไม่?

คำตอบ : สามารถทำได้ง่ายๆ เพียงแค่เซฟไฟล์ .AFL ของคุณใส่ในโฟลเดอร์ Formulas ที่อยู่ในโฟลเดอร์รวมของ AmiBroker ซึ่งจะปรากฏอยู่ในเมนูด้านซ้าย (กด View > Refresh All ด้วย เพราะบางทีเมื่อคุณเซฟไฟล์แล้วอาจจะต้องอัพเดตให้ระบบอ่านอีกครั้ง)

คำถาม : ฉันเพิ่มไฟล์ใหม่เข้าไปในโฟลเดอร์ Formulas แต่มันไม่แสดงไฟล์ใหม่นั้น ในเมนูเลย นอกจากจะปิดโปรแกรมแล้วเปิดใหม่ มีวิธีอื่นอีกไหมในการรีเฟรชเมนู?

คำตอบ : คุณสามารถรีเฟรชเมนูได้โดยการเลือก View > Refresh All

คำถาม : ถ้าฉันปรับแต่งสูตรที่มาพร้อมกับ AmiBroker ตั้งแต่แรก มันจะถูกเขียนทับอีกครั้งหรือไม่ ในการอัพเกรดครั้งต่อไป?

คำตอบ : ใช่แล้ว มันจะถูกเขียนทับ ถ้าคุณต้องการปรับแต่งโค้ดแบบดั้งเดิมซึ่งมาพร้อมกับ AmiBroker ให้เซฟไฟล์ในชื่อใหม่หรือจะดีกว่าถ้าเซฟไฟล์ไว้ในโฟลเดอร์ส่วนตัว

คำถาม : ฉันเห็นว่ามันมีปุ่ม Reset All ในหน้าต่างพารามิเตอร์ที่จะรีเซ็ตค่า ของพารามิเตอร์ทั้งหมดให้เป็นค่าเริ่มต้น มันพอมีวิธีบ้างไหม หากต้องการรีเซ็ตเพียงแค่ ค่าของพารามิเตอร์ตัวใดตัวหนึ่งเท่านั้น?

คำตอบ : ตอนนี้ยังไม่มี แต่ในอนาคตเราจะเพิ่มมันในรุ่นทดลอง

คำถาม : ฉันลาก RSI ใส่ในหน้าต่างราคาและเห็นเส้นสีแดงอยู่ด้านล่างสุดของหน้าต่าง มีอะไรผิดปกติหรือ?

คำตอบ : ทันทีที่คุณลากอินดิเตเตอร์สองตัวมาใส่รวมกัน ซึ่งทั้งสองตัวมีค่าที่แตกต่างกันอย่างสิ้นเชิง คุณจำเป็นต้องปรับตัว Style ให้เป็นแบบ OwnScale สำหรับอินดิเคเตอร์ตัวใดตัวหนึ่ง คุณสามารถเลือก Style ได้ผ่านหน้าต่างพารามิเตอร์ เมื่อทำแล้วจะมั่นใจได้เลยว่าอินดิเคเตอร์แต่ละตัวนั้นสามารถเห็นได้อย่างชัดเจน ไม่อย่างนั้นแล้วกราฟที่แสดงผลก็จะเป็นแบบแบนอย่างที่คุณได้พบ

คำถาม : คำสั่ง AFL อย่างเช่น _SECTION_BEGIN และอื่นๆ นั้นมันมีสีเทาอ่อนซึ่งมองยากมาก เพราะพื้นหลังของฉันนั้นก็เป็นสีเทา ฉันสามารถเปลี่ยนสีของคำสั่งเหล่านี้ได้หรือไม่?

คำตอบ : ในตอนนี้ยังไม่ได้ แต่พวกเราจะใส่การตั้งค่าให้สามารถเปลี่ยนสีได้ในเวอร์ชั่นถัดไป

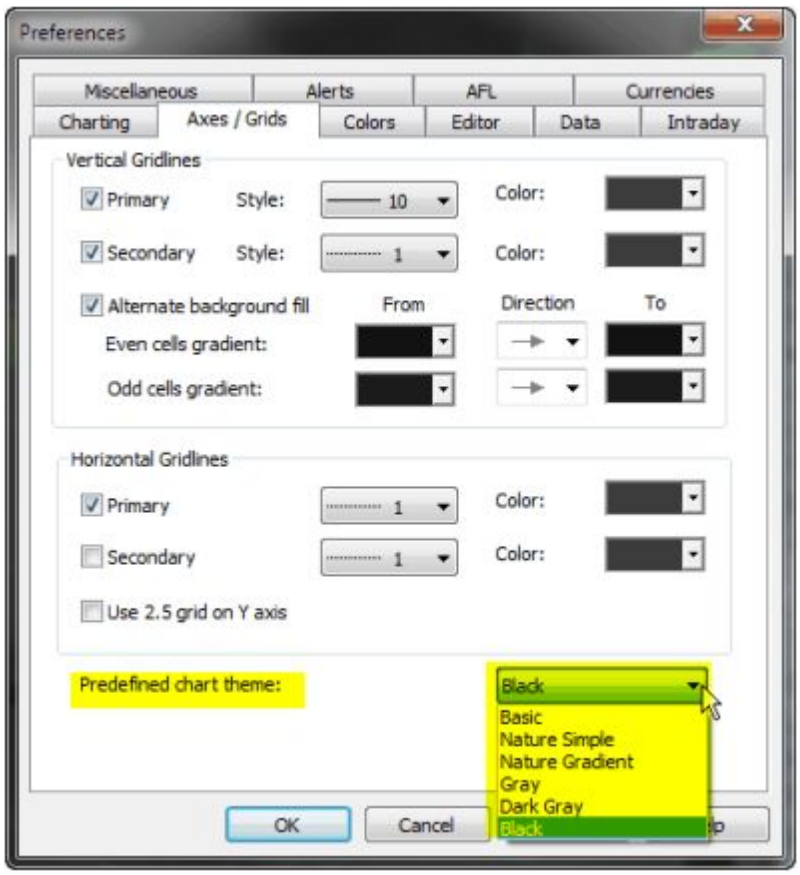
คำถาม : เมื่อฉันลากอินดิเคเตอร์มาใส่แล้ว หน้าต่างพารามิเตอร์กลับไม่โชว์ค่าพารามิเตอร์ของตัวอื่น ฉันทำถูกหรือเปล่า?

คำตอบ : ถูกแล้ว มันเป็นแบบนั้นล่ะครับ ไอเดียที่ซ่อนอยู่นั้นมันง่ายมาก เมื่อคุณลากอินดิเคเตอร์ตัวใหม่เข้ามาใส่ AmiBroker จะปรากฏหน้าต่างพารามิเตอร์เพียงแค่อินดิเคเตอร์ล่าสุดเท่านั้น นั่นทำให้แน่ใจได้ว่าพารามิเตอร์ที่ปรากฏนั้นเป็นของอินดิเคเตอร์ตัวล่าสุด และ ผู้ใช้ใหม่จะได้ไม่ต้องงมเข็มกับพารามิเตอร์เป็นสิบๆ ตัวจากอินดิเคเตอร์ที่สร้างมาก่อนหน้านี้ แต่ในอีกมุมหนึ่ง หากคุณคลิกเลือก “Parameters” จากเมนูผ่านการคลิกขวาที่หน้าต่างชาร์ท พารามิเตอร์ทั้งหมดจะโชว์ขึ้นมา และ คุณสามารถปรับแต่งมันทั้งหมดได้ในภายหลัง

ธีมของชาร์ท

(Chart themes)

AmiBroker 5.52 ได้นำเสนอธีม 6 แบบที่สามารถเปลี่ยนได้จาก



1. ธีมพื้นฐาน (**Basic Theme**)

2. ธีมแบบง่าย(**Nature Simple Theme**)

3. ธีมแบบมีสี (**Nature Gradient Theme**)

4. ธีมสีเทา (**Gray Theme**)

5. ธีมสีเทาเข้ม (**Dark Gray Theme**)

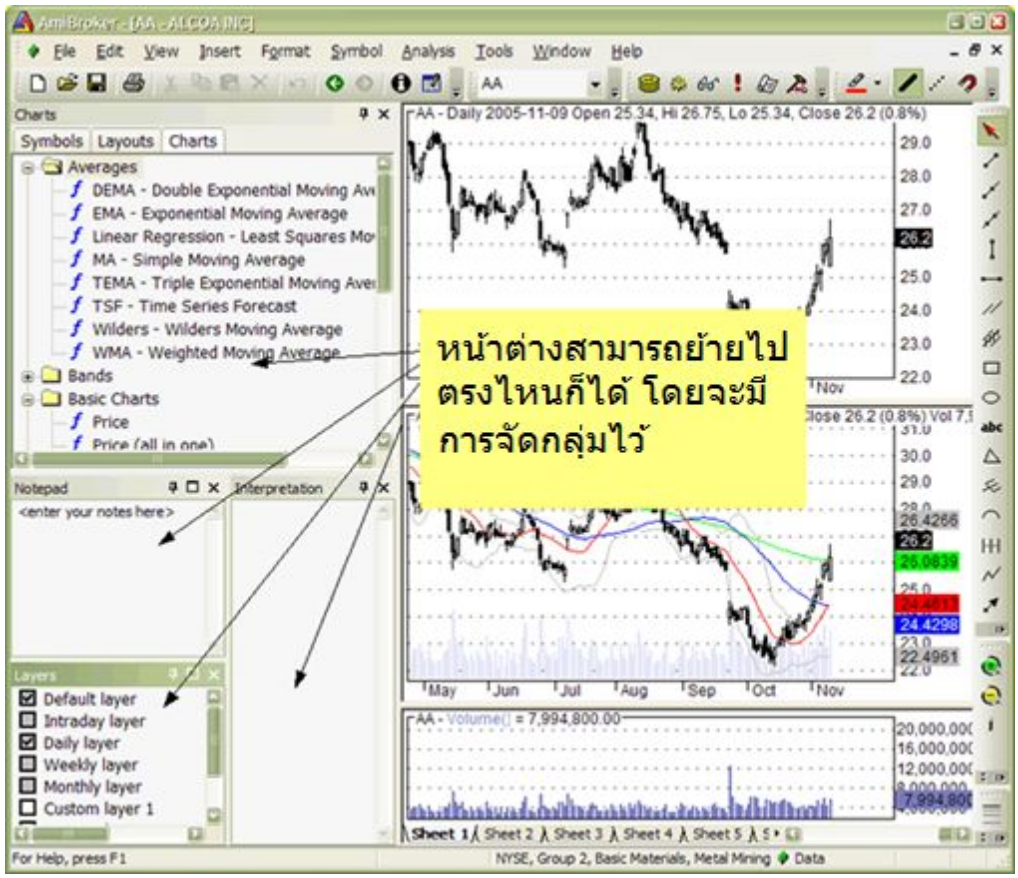
6. ธีมสีดำ (**Black Theme**)

การปรับแต่งหน้าจอผู้ใช้

(User interface customization)

ในเวอร์ชั่นใหม่นี้เรามีการปรับแต่งหน้าจอผู้ใช้หลายอย่างที่น่าประทับใจและควบคุมมันได้อย่างสมบูรณ์มากกว่ารูปลักษณ์ของ AmiBroker แบบเดิมๆ

การเอากลุ่มบางอันออก / การซ่อนแท็บ (Advanced nested docking / tear-off tabs)



ในการเอาหน้าต่างหรือแท็บออกไปไว้พื้นที่ใดๆของโปรแกรม ทำได้โดยการคลิกตรงแถบบนของหน้าต่างนั้นๆแล้วลากมัน ทันทีที่คุณลากจะมีปุ่มปรากฏขึ้นมาเพื่อให้คุณเลือกว่า จะเอาหน้าต่างนั้นๆไปวางไว้ตรงไหนได้ง่ายขึ้น ดังรูปด้านล่าง

คุณยังสามารถลากมันออกมาแล้วให้หน้าต่างนั้นเป็นหน้าต่างแยกต่างหากได้ ด้วยวิธีนี้คุณสามารถจัดการแยกหน้าต่างทั้งหมดเป็นหน้าต่างแยก หรือ รวมกันให้เป็นหมวดหมู่ได้ คุณยังสามารถลากมันได้อย่างอิสระ และ วางไว้ตรงไหนก็ได้ด้วยการกดปุ่ม Ctrl ค้างเอาไว้

การซ่อนหน้าต่างอัตโนมัติ (Sliding Auto-hide panes)

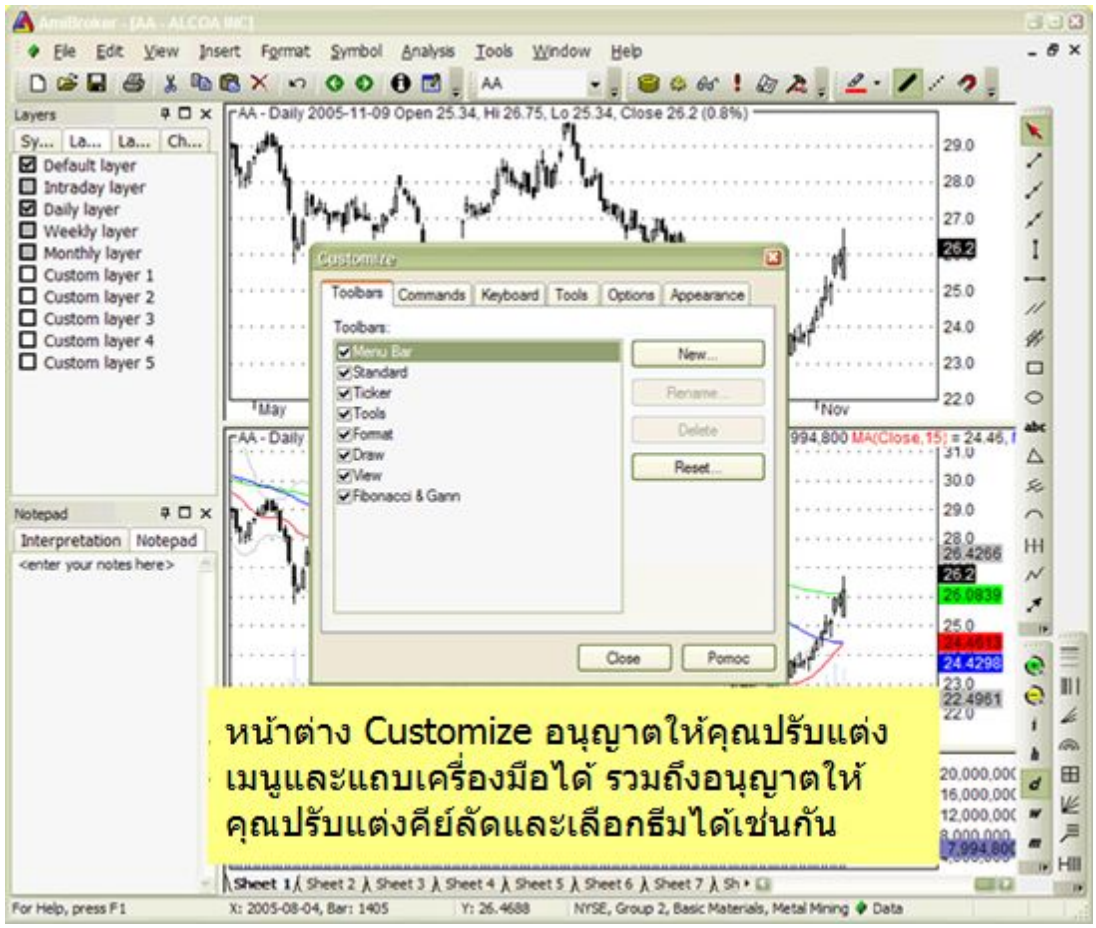
อีกความสามารถที่มีประโยชน์ก็คือการซ่อนหน้าต่างจากหน้าจอได้โดยอัตโนมัติ ในการเปิดปิดคำสั่งนี้จะอยู่ตรงปุ่มเข็มหมุดที่อยู่มุมบนขวาของแต่ละหน้าต่าง ถ้าคุณยกเลิกการปักหมุด หน้าต่างจะถูกซ่อนอัตโนมัติเมื่อเราไม่ได้ใช้งานมัน


ถอนการปักหน้าต่าง
(ซ่อนอัตโนมัติ)
หน้าต่างยังแสดงผลอยู่


หน้าต่างถูกซ่อน
อัตโนมัติแล้ว

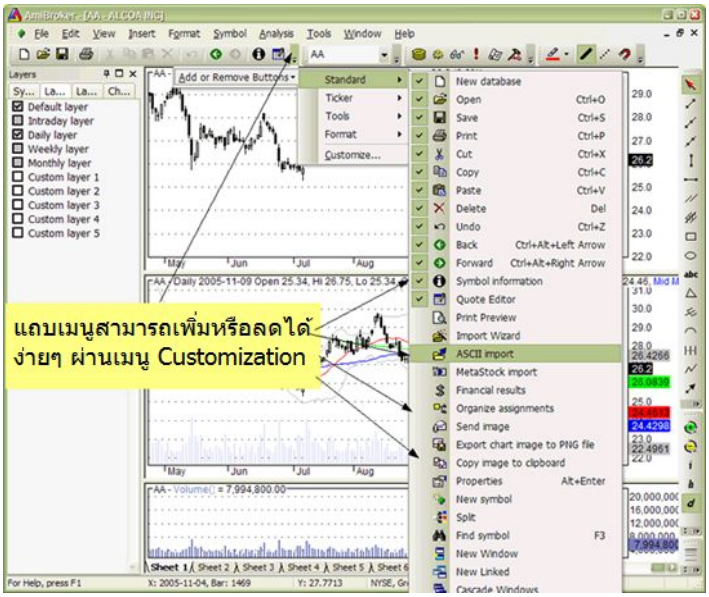
การปรับแต่งแถบเครื่องมือ เมนู และปุ่มลัดขั้นสูง (Advanced customizable toolbars, menus and keyboard shortcuts)

ความสามารถใหม่นี้สามารถทำให้ผู้ใช้สามารถควบคุมลักษณะที่ปรากฏได้ การแสดงผล และ ตำแหน่งต่างๆ ของแถบเครื่องมือทั้งหมด ปุ่ม และเมนู ซึ่งอนุญาตให้คุณเพิ่มปุ่มของคุณเอง ลบและจัดเองใหม่ได้ รวมถึงคุณสามารถกำหนดหรือแก้ปุ่มลัดได้เองด้วย ความสามารถในการปรับแต่งทั้งหมดนี้มีอยู่ในเมนู Tools > Customize หรือจะเข้าจากลูกศรที่อยู่ใต้แท็บก็ได้

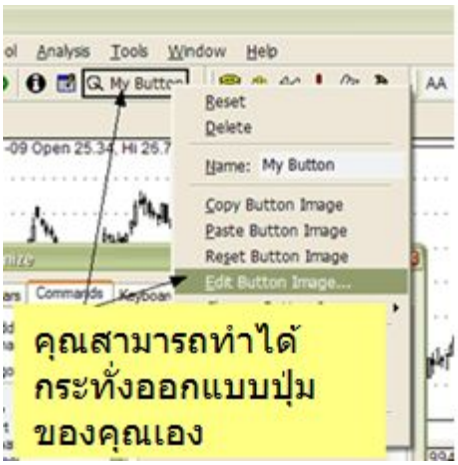


เครื่องหมายลูกศรที่ว่านั้นเป็นลูกศรเล็กๆ ที่อยู่ตรงด้านล่างสุดของขอบแถบเครื่องมือ เจ้าสิ่งนี้ได้ซ่อนเครื่องมืออื่นๆ ของแถบเมนูไว้โดยอัตโนมัติ ตลอดจนคำสั่งในการปรับแต่งแถบเมนู

เมนูย่อย Add or Remove Buttons จะแสดงแถบปุ่มเครื่องมือหรือซ่อนตามที่คุณต้องการในโหมดการปรับแต่ง (เมื่อคุณเข้าโดยไปที่ Tools > Customize) คุณยังสามารถย้ายปุ่มไปรอบๆ เพื่อเปลี่ยนลำดับการแสดงผลได้ และคุณยังสามารถปรับขนาดของส่วนนั้น รวมไปถึงรวมปุ่มให้อยู่ในส่วนเดียวกันได้ (เช่น ส่วนที่ไว้เลือกชื่อหุ้น) ด้วยการคลิกที่รายการที่เราต้องการ และปรับขนาดเมื่อเลือกเสร็จแล้ว

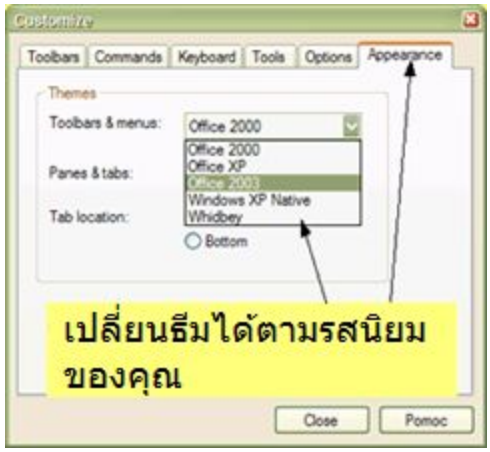


คุณสามารถทำได้แม้กระทั่งการออกแบบปุ่มของคุณเองด้วยโปรแกรมแก้ไขรูปภาพที่มีมาให้



ธีมที่ปรากฏ (**Themed appearance**)

Amibroker อนุญาตให้คุณเปลี่ยนหน้าตาของโปรแกรมที่ปรากฏตามที่รสนิยมของคุณได้เช่นกัน



แท็บ MDI (multiple document interface) (MDI (multiple document interface) tabs)

AmiBroker นั้นมีฟังก์ชั่นการแสดงผลแบบหลายหน้าต่าง (MDI) ความหมายง่ายๆ ก็คือเจ้าสิ่งนี้ทำให้คุณสามารถเปิดและทำงานหลายหน้าต่างได้ในเวลาเดียวกัน หากต้องการข้อมูลเพิ่มเติมว่า MDI คืออะไร คุณอาจจะอยากเข้าดูเนื้อหานี้ http://en.wikipedia.org/wiki/Multiple_document_interface

สำหรับตอนนี้แท็บ MDI (ดังรูป) เป็นอีกทางเลือกในการเปลี่ยนหน้าต่างเพื่อใช้งาน (นอกเหนือจากการเข้าผ่านเมนู Window ที่ซึ่งจะแสดงหน้าต่างที่สามารถใช้งานได้)

เป็นเรื่องสำคัญที่จะต้องเข้าใจว่าแท็บ MDI นั้น “ผู้ใช้ไม่สามารถรเจาะจงแท็บได้” หรืออีกแง่ก็ คือไม่สามารถเปลี่ยนชื่อได้ตามใจต้องการ ไม่เหมือนกับหน้าต่างชาร์ท (ที่สามารถระบุได้) ชื่อของแท็บจะถูกกำหนดโดยชื่อของหน้าต่างหรือเอกสารนั้นเอง สำหรับหน้าต่างชาร์ทนั้นชื่อจะถูกตั้งด้วยรูปแบบ : ตัวย่อ - ชื่อเต็มเสมอ, หน้าต่างสำหรับการเข้าเว็บไซต์จะใช้ HTML Title เป็นชื่อของหน้าต่าง (ซึ่งถูกกำหนดไว้แล้วในโค้ดของ HTML), หน้าต่างจัดการบัญชี จะใช้ชื่อไฟล์บัญชีจริงๆ (ซึ่งคุณสามารถกำหนดได้ต่อเมื่อคุณเซฟ)

แท็บ MDI นั้นไว้ใช้สำหรับการสลับหน้าต่าง (เหมือนกับ Taskbar ด้านล่างของ Windows) และมันถูกควบคุมโดยอัตโนมัติจาก AmiBroker เมื่อคุณเปิดหรือปิดหน้าต่าง และ แน่นอนว่ามันทำงานแบบเดียวกับ Taskbar ของ Windows เลย มาเปรียบเทียบกันดูดีกว่า

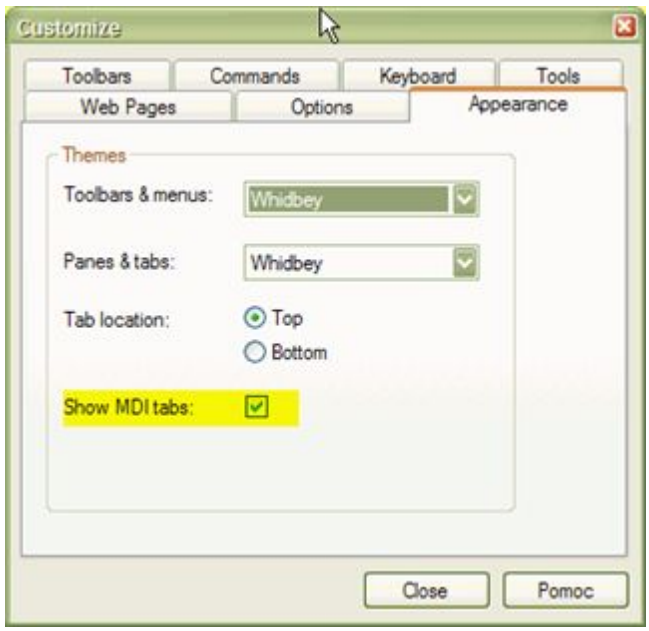
เมื่อคุณใช้ Taskbar ของ Windows

- คุณเปิดโปรแกรม - ปุ่มใหม่ในแถบ Taskbar จะปรากฏขึ้น
- และคุณสามารถสลับหน้าโปรแกรมที่เปิดไว้อยู่ได้ผ่านปุ่มเหล่านั้น
- คุณไม่สามารถเปลี่ยนชื่อปุ่มได้เพราะมันแสดงชื่อของโปรแกรมอยู่
- และคุณจำเป็นต้องระวังเมื่อคุณเปิดหลายโปรแกรมมากไป
  เพราะโปรแกรมที่เปิดขึ้นมาจะดึงทรัพยากรระบบมาใช้ (ให้การกระบวนการทำงานช้าลง)

คราวนี้ เรามาดูแท็บ MDI ของ AmiBroker กันบ้าง

- คุณเปิดเอกสาร (window) > ปุ่มใหม่ปรากฏขึ้น (แท็บ)
- คุณสามารถสลับหน้าต่างด้วยปุ่ม (แท็บ)
- คุณไม่สามารถเปลี่ยนชื่อหุ้นได้เพราะมันแสดงผลชื่อของเอกสาร / หน้าต่างอยู่
- และคุณจำเป็นต้องระวังเมื่อเปิดหลายๆ เอกสาร / หน้าต่าง พร้อมกัน เพราะทุกหน้าที่คุณเปิดจะเป็นการดึงทรัพยากรระบบมาใช้

คุณสามารถปิดแท็บ MDI ได้ด้วยการเอาเครื่องหมายถูกในส่วนของ “Show MDI tabs” ที่ปรากฏอยู่ใน Tools > Customize ออกไป ดังตัวอย่างด้านล่าง



หมายเหตุสำหรับเวอร์ชั่นเก่า : ในเวอร์ชั่นก่อน 4.90 การเปลี่ยนหน้าเอกสารนั้นคุณต้องใช้ผ่านเมนู Window แต่ในตอนนี้มีอีกทางเลือกหนึ่ง ก็คือคุณสามารถใช้แท็บได้ แต่ฟังก์ชั่นนี้สามารถทำได้เพียงแค่อำนวยความสะดวกสบายเท่านั้น สำหรับข้อมูลเพิ่มเติมโปรดดูที่http://en.wikipedia.org/wiki/Tabbed_Document_Interface (โปรดจำไว้ว่าวิกิพีเดียลิ้งก์นี้อธิบายเรื่อง TDI / MDI หรืออะไรบางอย่างที่ล้าสมัยไปแล้ว และ AmiBroker ได้รวมทั้งวิธีการแบบ TDI และ MDI ไว้ด้วย

สำหรับข้อมูลเพิ่มเติม ดูได้ที่เอกสารนำเสนอของการประชุมที่ Houston
http://www.amibroker.com/docs/Houston1.pdf (ไฟล์ PDF)
http://www.amibroker.com/docs/Houston1.html (ไฟล์ Flash)

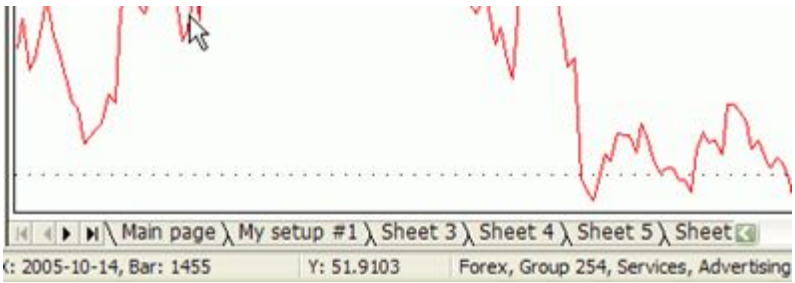
การใช้งานหน้าชาร์ทและหน้าต่าง Layout

(Working with chart sheets and window layouts)

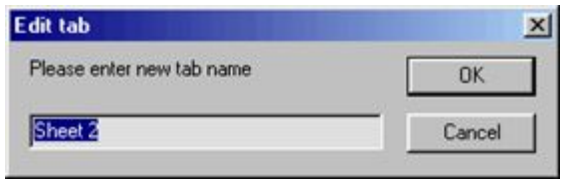
AmiBroker มีการจัดการชาร์ทหลายๆหน้าต่าง และหน้าต่าง Layout หลายๆ หน้าด้วยความสามารถที่รวดเร็วในการโหลดและเซฟ ความสามารถนี้ทำให้คุณสามารถเปลี่ยน หรือ สลับอินดิเคเตอร์ที่คุณตั้งไว้ได้อย่างรวดเร็ว และ ประหยัดเวลาได้อย่างมาก

หน้าชาร์ทและรูปแบบ (Chart sheets and templates)

หน้าชาร์ทจะถูกตั้งไว้อยู่ในชุดของ Chart Pane (พร้อมอินดิเคเตอร์) ซึ่งแสดงผลอยู่ในหน้าเดียว คุณสามารถสลับระหว่างหน้าแต่ละหน้าด้วยการคลิกที่แท็บซึ่งอยู่ด้านล่างสุดของหน้าต่างโปรแกรม AmiBroker ตามรูปภาพด้านล่าง



คุณยังสามารถเปลี่ยนชื่อของแท็บได้ด้วยการคลิกขวาที่แท็บ ดังรูปนี้



คุณสามารถเปลี่ยนชื่อแท็บทั้งสี่ (แต่ละแท็บ) เพื่อรายละเอียดที่มากขึ้นได้ (และเกี่ยวข้องกับเนื้อหาในหน้านั้น)

คุณสามารถเลื่อนแท็บได้ด้วยลูกศรและยังสามารถเลื่อนการจัดเรียงได้ด้วยการลาก (คลิกที่แท็บค้างไว้แล้วลากไปยังจุดที่ต้องการ ลูกศรที่ปรากฏขึ้นจะแสดงตำแหน่งที่ได้วางไว้)

คุณยังสามารถเข้าถึงหน้าใดๆ ด้วยความรวดเร็ว ผ่านการคลิกขวาบนลูกศร เพื่อให้มีเมนูเด้งขึ้นมา และ รายการที่อยู่ในเมนูนั้นคือชื่อแท็บทั้งหมดซึ่งอนุญาตให้คุณเข้าถึงได้แบบทันทีด้วยเพียงแค่คุณคลิก (โดยไม่ต้องเลื่อน)

ขั้นตอนต่อไปคือการตั้งค่าหน้าต่างๆ ให้แสดงผลได้ตามต้องการ เพียงแค่ใช้ในส่วนของ เพิ่ม/ลด จากแต่ละหน้า ด้วยวิธีนี้คุณสามารถตั้งค่าอินดิเคเตอร์ได้สูงสุดถึง 60 ตัว และ สามารถเรียกใช้มันใหม่อีกครั้งได้อย่างรวดเร็วผ่านการคลิกที่แท็บ ส่วนการกำหนดจำนวนหน้าของแท็บนั้นสามารถกำหนดได้ที่ Tools > Preferences > Charting > “Number of chart sheets”

การตั้งค่าแบบสมบูรณ์แล้วเราเรียกมันว่า “เทมเพลต” และ คุณสามารถตั้งให้มันเป็นแบบถาวรได้เพียงแค่คลิกขวาที่ชาร์ทและเลือกเมนูดังนี้ Template >Save, Template, Save as default

เทมเพลตพื้นฐานนั้นใช้เมื่อคุณสร้างหน้าต่างใหม่ขึ้นมา (Window > New)

คุณยังสามารถโหลดเทมเพลตที่เซฟไว้ได้ด้วยการเลือก Template > Load จากเมนูผ่านการคลิกขวาที่หน้าต่างชาร์ท

นอกจากนี้ สมมติว่าเทมเพลตเก่าจะถูกเซฟใหม่ด้วยไฟล์นามสกุล .chart ซึ่งนั่นจะไม่เพียงแค่บันทึกขนาด และ สูตรอ้างอิงของชาร์ทเท่านั้น แต่ยังรวมถึงโค้ดของตัวมันเองด้วย ดังนั้น ทุกอย่างที่คุณต้องการจะอยู่ในไฟล์เดียว (เทมเพลตชาร์ท, ไฟล์ .chart แบบสมบูรณ์) และถ้าคุณคัดลอกไฟล์นี้ไปเปิดที่คอมพิวเตอร์อื่น ชาร์ทจะถูกสร้างใหม่ด้วยสูตรเดิมของมัน

การเซฟชาร์ทใส่รูปแบบใหม่ให้ทำตามขั้นตอนนี้

1. คลิกขวาบนชาร์ทและเลือก Template > Save...

2. ในหน้าต่างโต้ตอบที่เด้งขึ้นมา “File of type” ทั้งคู่ให้เลือก “Chart Template, Complete (*.chart)”

3. ตั้งชื่อไฟล์ แล้วเลือกเซฟ

ส่วนการโหลดชาร์ทก่อนหน้าที่เซฟเสร็จแล้วให้ทำตามนี้

1. คลิกขวาบนชาร์ทแล้วเลือก Template > Load...

2. ในหน้าต่างโต้ตอบ เลือกไฟล์ *.chart ที่เซฟไว้ก่อนหน้านี้ แล้วกด “Open”

หมายเหตุ : ขั้นตอนที่ AmiBroker ทำอยู่ภายในจะเป็นดังนี้ : เมื่อคุณเซฟไฟล์ชาร์ทในรูปแบบใหม่ มันจะมีการเซฟไฟล์ XML ไว้ด้วยดังนี้

a) ชื่อของหน้าทั้งหมด หน้าต่าง ขนาด ตำแหน่งและการตั้งค่าอื่นๆ
b) โค้ดทั้งหมดที่เชื่อมโยงกับทุกหน้า
c) โค้ดของตัวมันเอง

เมื่อคุณโหลดชาร์ท AmiBroker ในคอมพิวเตอร์เครื่องใหม่

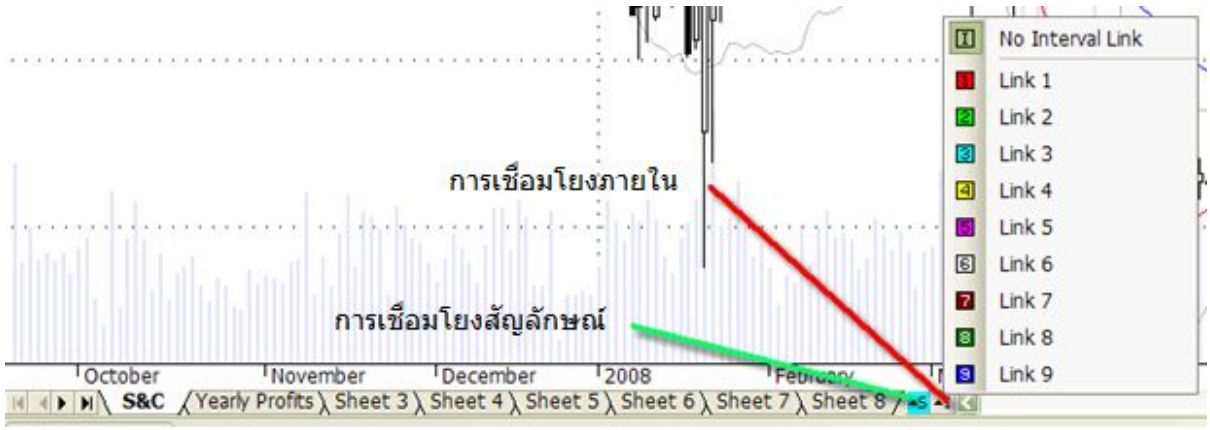
a) ตั้งค่าหน้า / หน้าต่างตามที่ได้กำหนดไว้ในไฟล์
b) สำหรับแต่ละสูตรที่ถูกเก็บไว้ในไฟล์ จะถูกเช็คว่าตรงกับสูตรที่อยู่ในคอมพิวเตอร์อีกเครื่องหรือไม่
    - ถ้าไม่มีไฟล์อยู่ - ระบบจะสร้างขึ้นมาอีกอัน
    - ถ้ามีไฟล์อยู่และเนื้อหาในไฟล์เหมือนกัน ไฟล์ .chart จะไม่ทำอะไร
- ถ้ามีไฟล์อยู่แต่เนื้อหาในไฟล์ต่างกัน ระบบจะสร้างสูตรใหม่ขึ้นมาโดยมี _imported.afl ต่อท้าย (ดังนั้นไฟล์เก่าจะไม่ถูกนำมาใช้) และจะอ้างอิงสูตรกับหน้าต่างด้วยสูตรที่มี _import.afl ต่อท้ายแทน

เรื่องสำคัญ : ถ้าคุณใช้คำสั่ง #include ใดๆ ก็ตาม AmiBroker จะเก็บเนื้อหาของไฟล์นั้นเอาไว้ในไฟล์ชาร์ท และจะพยายามสร้างไฟล์ใหม่ในคอมพิวเตอร์อีกเครื่อง โปรดจำไว้ว่าในกรณีนี้จะมีการเช็คไฟล์ว่าต่างกันหรือไม่ ถ้าทั้งสองไฟล์มีความต่างกัน โปรแกรมจะมีการถามว่าต้องการให้สร้างไฟล์ทับหรือไม่ ถ้าคุณเลือกให้สร้างไฟล์ทับไปเลย ไฟล์จะถูกสร้างทับและถูกสำรองเอาไว้อีกอันหนึ่ง กลับมาที่เรื่องส่วนขยาย ถ้าคุณใช้ไฟล์ใดๆ ก็ตามในฐานะ “standard include files” และดึงพวกมันมาด้วยเครื่องหมาย <> AmiBroker จะสำเนาไฟล์เก็บไว้ในโฟลเดอร์ Standard Include ซึ่งอยู่ในคอมพิวเตอร์เครื่องที่เปิด (แม้ว่าโฟลเดอร์ Standard Include นั้นมีส่วนที่ไม่เหมือนกันกับเครื่องต้นฉบับก็ตาม)

ไฟล์ .chart รูปแบบใหม่นี้มีวัตถุประสงค์เพื่อใช้ระหว่างคอมพิวเตอร์ที่แตกต่างกัน ส่วนการจัดเก็บรูปแบบและเทมเพลตบนคอมพิวเตอร์เครื่องหลักคุณควรจะใช้การจัดรูปแบบอันเก่าซึ่งมันกินความจำน้อยกว่า (พวกมันจะเก็บเพียงแค่การอ้างอิงเท่านั้น ไม่รวมถึงโค้ดของมันเอง) แต่ถึงอย่างนั้นก็ตาม การใช้การจัดรูปแบบอันใหม่เพื่อจุดประสงค์ในการเก็บสูตรและการอ้างอิงทั้งหมดไว้ในไฟล์เดียวก็เป้นสิ่งที่สะดวกมากในการสำรองข้อมูล

ชื่อหลักทรัพย์และการเชื่อมโยงภายใน (Symbol and Interval Linking)

คุณสามารถที่จะเชื่อมโยงหน้าต่างชาร์ทแต่ละอันด้วยสัญลักษณ์ และ/หรือ ไทม์เฟรม ในการเชื่อมโยงชาร์ทนั้นให้ใช้ปุ่ม Linking ซึ่งอยู่ตรงด้านล่างสุดของชาร์ท ดังรูปภาพต่อไปนี้

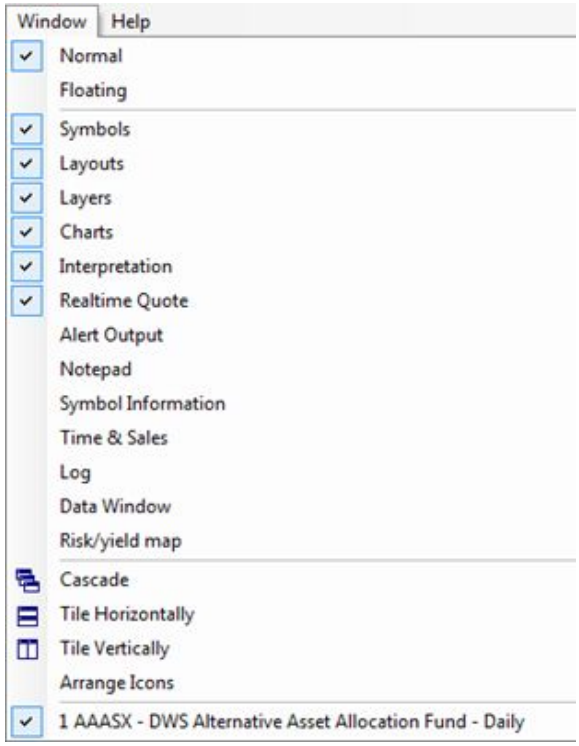


ปุ่ม S และ i ที่เป็นสีเทานั้นหมายถึงไม่มีการเชื่อมโยงใดๆ ถ้าเป็นสีอื่น (แดง, เขียว, ฟ้าอ่อน, เหลือง, ชมพู, ขาว, น้ำตาล, เขียวเข้ม, น้ำเงิน) จะหมายถึงมันมีการเชื่อมโยงระหว่างชาร์ทด้วยการกำหนดสีไว้แล้ว หน้าต่างทุกอันที่มีสีลิ้งก์เหมือนกันจะมีการเปลี่ยนสัญลักษณ์ และ/หรือ การเปลี่ยนแปลงภายในไปพร้อมๆ กัน

การแยกหน้าต่าง (Floating windows)

ถ้าคุณมีหลายหน้าจอ คุณสามารถใช้มันเพื่อแสดงผลชาร์ท AmiBroker ในหลายหน้าจอได้ การทำวิธีนี้ก็ง่ายมาก AmiBroker ตั้งแต่เวอร์ชั่น 5.10 ได้นำเสนอวิธีการที่เรียกว่า “Floating” ซึ่งใช้แยกหน้าต่างชาร์ทออกจากกัน โดยปกติแล้วชาร์ททุกหน้าที่เปิดจะอยู่ในหน้าโปรแกรม AmiBroker ถ้าคุณต้องการแยกชาร์ทออกจากหน้าโปรแกรม คุณก็เพียงดึงมันออกจากหน้าต่างหลัก ดังนั้นคุณสามารถลากมันไปที่อื่นได้ ยกตัวอย่างเช่น ใส่ไว้ในหน้าจออื่น

คุณสามารถสลับระหว่างโหมดปกติและโหมดแยกหน้าต่างได้ด้วยวิธีการดังรูป



วิดีโอต่อไปนี้จะสาธิตการใช้งานว่าจะแยกหน้าต่างออกจากกันและเชื่อมโยงสัญลักษณ์ได้อย่างไร

http://www.amibroker.com/video/FloatAndLink.html

โครงสร้างของหน้าต่าง (Window layouts)

โครงสร้างของหน้าต่างนั้นจะถูกตั้งค่าให้เปิดหลายหน้าต่างโดยที่แต่ละอันจะเป็นชื่อสัญลักษณ์ที่ต่างกันเอาไว้แล้ว ต่างกันทั้งการแสดงผลภายใน ขนาดที่ต่างกัน และการตั้งหน้าชาร์ทที่ต่างกันด้วย

ภาพด้านล่างนี้จะแสดงให้เห็นโครงสร้างของ 4 หน้าต่าง ที่มีความแตกต่างทั้งการตั้งค่า และอินดิเคเตอร์ ด้านซ้ายคุณจะเห็นหน้า “Layouts” ซึ่งเป็นพื้นที่สำหรับแสดงผลชื่อของไฟล์ที่เก็บไว้ใน Local layouts และ Global layouts

ถ้าคุณใช้ AmiBroker ตั้งแต่เวอร์ชั่น 4.20 ขึ้นไป คุณสามารถปรับแต่งได้ไม่จำกัด การใช้งานแม่แบบหลายอันสามารถทำได้ด้วยการสลับ เพียงแค่ดับเบิ้ลคลิ้กชื่อของ Layout ที่ตั้งไว้ซึ่งอยู่ในแท็บ “Layouts”

คุณสามารถ เปิด, บันทึก, ลบ Layout ด้วยการคลิกที่พื้นที่ว่างของหน้า Layout ด้วยการคลิกขวา และ เลือกที่รายการ “Save As” เพื่อบันทึก Layout ปัจจุบันด้วยชื่อใหม่

Local layouts จะอยู่ในแต่ละฐานข้อมูลเท่านั้น ในขณะที่ Global layouts จะสามารถเห็นได้จากทุกฐานข้อมูล

ข้อมูลที่ถูกบันทึกใน Layout จะประกอบไปด้วย : ขนาดของหน้าต่างและการจัดตำแหน่ง, การย่อและ ขยายของหน้าชาร์ทในแต่ละหน้า (ซึ่งแยกกันแต่ละหน้า), บาร์ที่เลือกเอาไว้, สัญลักษณ์ที่เลือกเอาไว้, หน้าชาร์ทที่เลือกเอาไว้

Layout ที่ใช้ล่าสุดจะถูกเซฟเมื่อปิดและฐานข้อมูลจะสลับให้อัตโนมัติ (ตั้งค่าที่ Tools > Preferences > Miscellaneous “Save on exit: Layouts”)

หมายเหตุ : ตั้งแต่เวอร์ชั่น 4.90 เป็นต้นมา การใช้งานหลายหน้าต่างสามารถเปลี่ยนได้ไม่เพียงแต่รูปแบบเก่าที่เข้าผ่านเมนู Window เท่านั้น แต่ยังรวมถึงการใช้แท็บ MDI ด้วย ข้อมูลเพิ่มเติมเกี่ยวกับแท็บ MDI เข้าไปดูได้ที่บทก่อนหน้า

การใช้งาน Layers

(Using layers)

Layers คืออะไร (What layers are)

Layers เหมือนกับชิ้นส่วนของพลาสติกใส คุณสามารถวาดบนเจ้าสิ่งนี้ได้ เลเยอร์ยังสามารถทำให้มองเห็น หรือ มองไม่เห็นได้ด้วย สิ่งนี้อนุญาตให้คุณ แสดง/ซ่อน สิ่งที่วาดบนเลเยอร์ได้โดยจะไม่มีผลกับเลเยอร์อื่นๆ

Layers ทำงานอย่างไร(How to work with layers)

ก่อนอื่น ทำให้แน่ใจว่าพื้นที่ใช้งานนั้นสามารถมองเห็นได้ (Window > Layers) จากนั้น เปลี่ยนไปที่แท็บ “Layers” ในตอนนี้คุณจะเห็นชื่อของเลเยอร์ที่กำหนดไว้แล้ว ช่องทำเครื่องหมายที่อยู่ด้านซ้ายของแต่ละเลเยอร์นั้นไว้ควบคุมการแสดงผล ถ้าช่องนั้นถูกติ๊กไว้แปลว่าเลเยอร์นั้นสามารถมองเห็นได้ แต่ถ้าไม่มีการติ๊กไว้ เลเยอร์นั้นจะมองไม่เห็น ในเบื้องต้นนั้นเลเยอร์ 5 อันแรกจะถูกล็อกไว้

เลเยอร์ที่ว่านี้ประกอบด้วย

Default Layer - จะถูกมองเห็นตลอดเวลา

Intraday Layer - จะเห็นต่อเมื่อดูชาร์ทแบบ Intraday เท่านั้น

Daily Layer - จะเห็นต่อเมื่อดูชาร์ทแบบรายวันเท่านั้น

Weekly Layer - จะเห็นต่อเมื่อดูชาร์ทแบบรายสัปดาห์เท่านั้น

Monthly Layer - จะเห็นต่อเมื่อดูชาร์ทแบบรายเดือนเท่านั้น

เลเยอร์ที่ถูกล็อกเอาไว้จะมองเห็นโดยอัตโนมัติเมื่อคุณเปลี่ยนรายละเอียดภายในและคุณไม่สามารถเปลี่ยนการมองเห็นผ่านการติ๊กช่องด้านซ้ายได้

เลเยอร์ที่เหลืออยู่นั้นไม่สามารถล็อกและพวกมันสามารถกำหนดการมองเห็นได้ผ่านการติ๊กที่ช่องด้านซ้าย

ในการวาดเครื่องมือต่างๆ ลงบนเลเยอร์ สามารถทำได้โดย

a) เลือกเลเยอร์ที่ต้องการ (คลิกที่ชื่อจนขึ้นไฮไลต์)
b) วาดเครื่องมือเหมือนอย่างเคย

เมื่อคุณเลือกเปิดเลเยอร์อื่น ทุกเครื่องมือวาดที่คุณวาดไว้จะอยู่บนเลเยอร์ที่เลือกนั้นด้วย หลังจากที่วาดแล้วคุณสามารถกำหนดมันให้อยู่ในเลเยอร์อื่นได้ผ่านการเลือก Object Properties Box

เมนูย่อย (Context menu)

ถ้าคุณเลือกที่ชื่อเลเยอร์ด้วยการคลิกขวา คุณจะเห็นเมนูย่อยปรากฏขึ้นมาและมีรายการดังนี้

Add layer

Remove layer

Show all layers

Hide all layers

Toggle

Unlock built-in layers

Lock built-in layers

Properties

การเพิ่ม/ลบเลเยอร์สามารถทำได้เอง แต่โปรดจำไว้ว่าคุณไม่สามารถลบเลเยอร์ 5 อันแรกได้ (ที่ถูกตั้งมาไว้แต่แรก)

Show all/ Hide all - แสดงและซ่อนเลเยอร์ทั้งหมดที่ไม่ได้ล็อกไว้

Toggle - สลับการมองเห็นของเลเยอร์ทั้งหมดที่ไม่ได้ล็อกไว้

Unlock/Lock built in layers - อนุญาตให้คุณปลดล็อก/ล็อก 5 เลเยอร์แรกได้ (ที่ถูกติดตั้งมา) แต่ละครั้งที่ล็อก การมองเห็นจะไม่ถูกเปลี่ยนแปลงอัตโนมัติเมื่อเราเปลี่ยนไทม์เฟรม และคุณสามารถแสดง/ซ่อนด้วยตัวคุณเองได้

Properties - การเปิดตัวเลือกนี้อนุญาตให้คุณสามารถเปลี่ยนชื่อเลเยอร์ และตัดสินใจได้ว่าอยากให้เลเยอร์นั้นควรหรือไม่ควรล็อกการแสดงผล

ถ้าคุณติ๊ก “Lock visibility to select interval” เลเยอร์นั้นจะแสดง/ซ่อนอัตโนมัติขึ้นอยู่กับว่ารายละเอียดภายในนั้นกำลังแสดงผลอะไรอยู่ คุณสามารถกำหนดการมองเห็นของแต่ละเลเยอร์ได้ด้วยปุ่ม “Interval” ร่วมกับปุ่ม “Show/hide automatically” โปรดจำไว้ว่ามันเป็นการแยกการมองเห็นในแต่ละอันหน้าต่างคุณสมบัติของเลเยอร์นั้นจะตั้งค่าเป็นการแสดงผลแบบกราฟรายเดือนเสมอ แต่คุณสามารถตั้งค่า และตั้งการปรับการมองเห็นได้ ในการล็อกเลเยอร์นั้นคุณจำเป็นต้องตั้งค่าการมองเห็นสำหรับทุกๆเลเยอร์ใน Interval Combo-box ง่ายๆ เพียงแค่เลือกไทม์เฟรมที่ต้องการถ้าเลเยอร์นั้นควรจะถูกแสดงผล หรือไม่แสดงผล จากนั้นเลือกไทม์เฟรมอันใหม่และทำแบบเดิมอีกครั้ง จนกระทั่งคุณกำนดไทม์เฟรมทั้งหมด

การใช้งานหน้าต่างการวิเคราะห์แบบใหม่

(Using New Analysis window)

เกริ่นนำ (Introduction)

หน้าต่างการวิเคราะห์แบบใหม่ถูกนำมาใช้ในเวอร์ชั่น 5.50 เป็นต้นมา (จริงๆแล้วครั้งแรกมีอยู่ในเวอร์ชั่นทดลอง 5.41.0) มาดูกันดีกว่าว่ามีความสามารถอะไรบ้าง

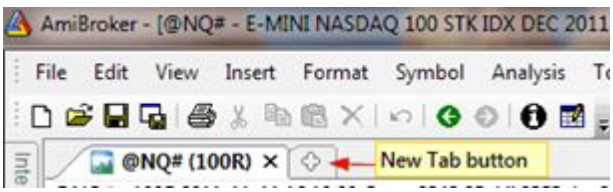
- **การเชื่อมโยงหลายการทำงานอย่างรวดเร็ว** หน้าต่าง Analysis จะใช้ CPU/แกนประมวลผลทั้งหมดที่ว่างอยู่ในการประมวลผลสูตรจากหลายๆการเชื่อมโยงพร้อมกันเพื่อเร่งความเร็วให้เต็มที่ ยกตัวอย่างเช่น ใน CPU Intel i7 แบบ 4 แกนสามารถเร่งได้สูงสุดถึง 8 การเชื่อมต่อ มันสามารถประมวลผลได้เร็วกว่าหน้าต่างการวิเคราะห์แบบเดิมถึง 8 เท่า แน่นอนว่าความเร็วที่เพิ่มขึ้นนั้นจะขึ้นอยู่กับความซับซ้อนของสูตรด้วย (ยิ่งซับซ้อนเท่าไหร่ยิ่งเป็นไปได้ที่จะมีการเร่งความเร็ว) และจำนวนของข้อมูลที่ใช้ในการประมวลผล (การใช้ RAM อาจจะไม่สามารถเพิ่มความเร็วได้เหมือน CPU ดังนั้นมันเป็นไปได้ที่ความเร็วจะถูกจำกัด)
- **ไม่ปิดกั้นการทำงาน** ในตอนนี้คุณสามารถดู เลื่อนและจัดลำดับผลลัพธ์ของการวิเคราะห์ในระหว่างที่มันกำลังประมวลผลอยู่ได้ ดังนั้นแล้วหน้าจอผู้ใช้จะไม่ถูกใช้ไปกับการประมวลผลเพียงอย่างเดียว ชาร์ทและอื่นๆ ยังสามารถใช้งาน และ ตอบสนองได้มากกว่าการประมวลผลเวอร์ชั่นก่อนหน้า
- **วิเคราะห์หลายอย่างพร้อมกัน** คุณสามารถประมวลผลข้อมูลได้มากกว่าข้อมูลเดียวในเวอร์ชั่นใหม่นี้ ดังนั้นคุณสามารถ Scan / Backtest / Explore และ Optimize ได้พร้อมกันโดยที่ไม่ต้องรอให้อย่างใดอย่าง หนึ่งเสร็จก่อนเลย
- **หน้าตาโปรแกรมที่เรียบง่าย** หน้าต่างการวิเคราะห์เวอร์ชั่นใหม่นี้คุณสามารถใช้งานได้เหมือนกับแท็บของเอกสารได้ สามารถย้าย สามารถจัดเรียงปุ่มได้ด้วย ด้วยวิธีนี้คุณจะมีที่ว่าง

สำหรับแสดงผลลัพธ์มากกว่าเดิม ข้อมูลเพิ่มเติมเกี่ยวกับการประมวลผลนั้นจะถูกแสดงในแท็บ “Info” ดังนั้นผลลัพธ์การเทสแบบ Walk-Forward ตอนนี้จะถูกแสดงอยู่ในหน้าวิเคราะห์ซึ่งดูวุ่นวายน้อยลง

หน้าจอผู้ใช้ (User interface)

คุณสามารถเปิดหน้าจอ Analysis ได้ตามขั้นตอนนี้

1. คลิกที่ปุ่มเพิ่มแท็บ (+) และเลือก New Analysis



หรือ

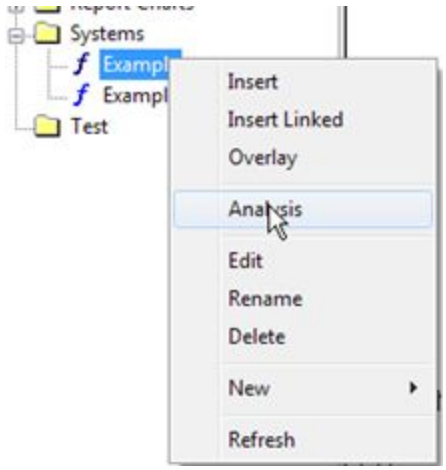
2. เข้าเมนู File > New > New Analysis

หรือ

3. เข้าเมนู Analysis > New Analysis

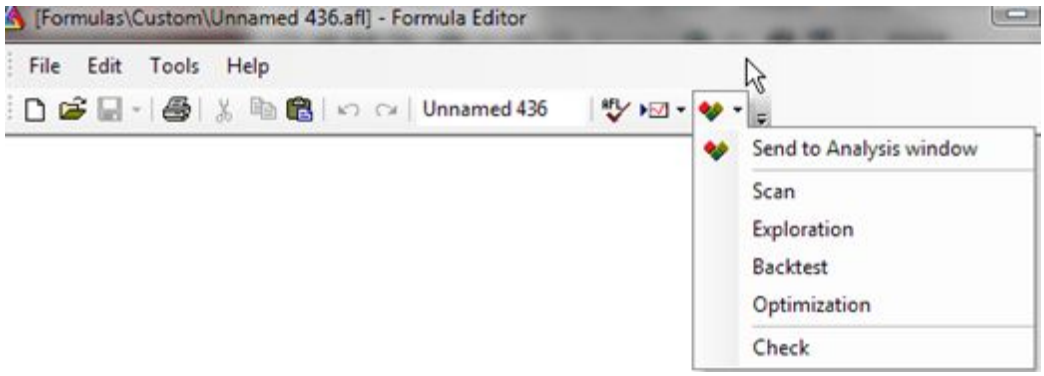
หรือ

4. คลิกขวาที่สูตรในหน้าชาร์ท และเลือก Analysis

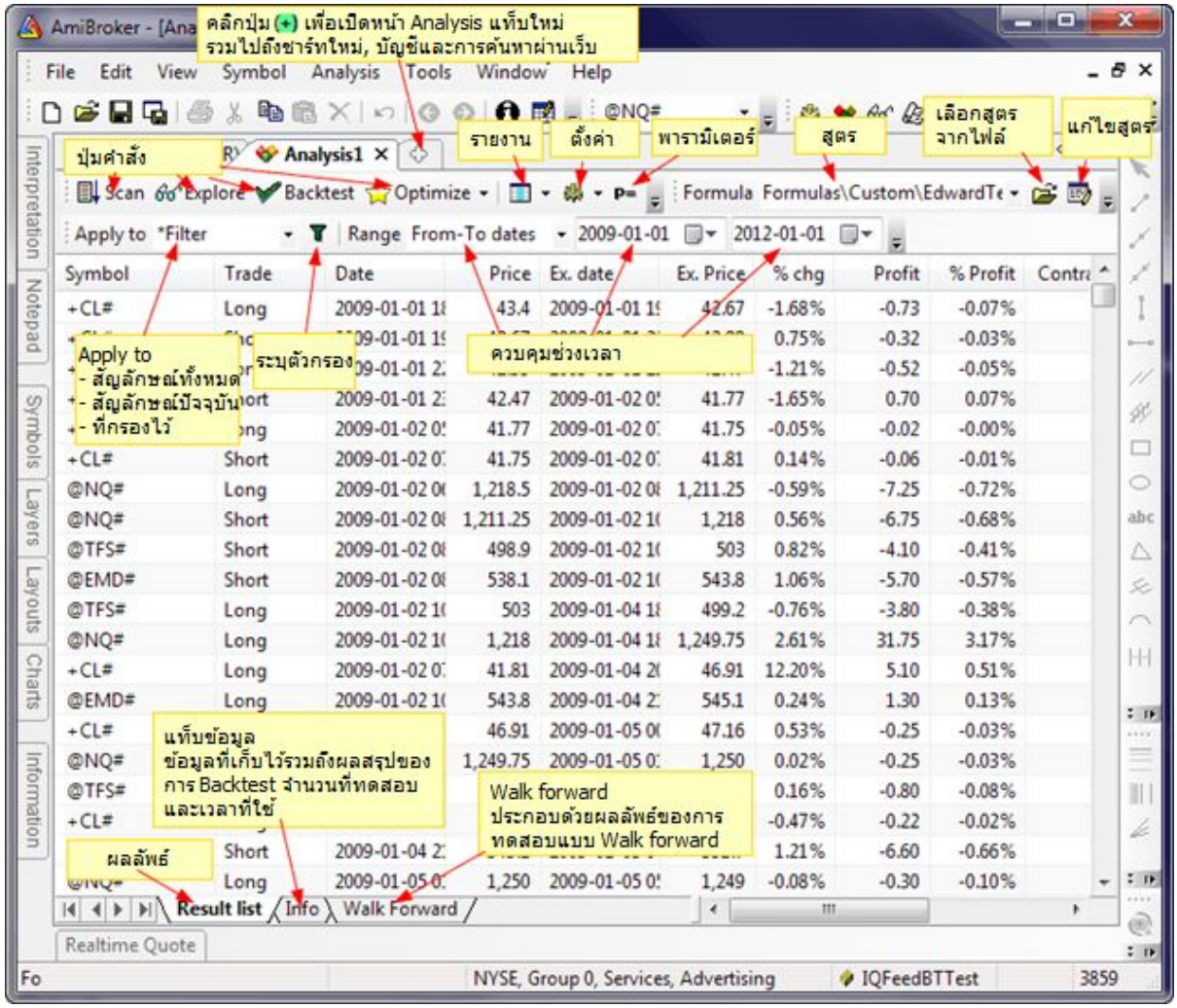


หรือ

5. จากในหน้า Foumular Editor กดที่ปุ่ม Send to Analysis



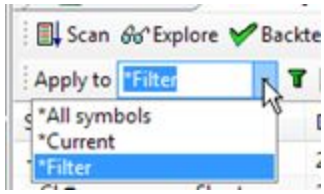
โดยพื้นฐานแล้วหน้าตาโปรแกรมของหน้า Analysis นั้นจะมีการใช้งานหคล้ายกับเวอร์ชั่นเก่าดังนี้



การทำงานพื้นฐาน (Basic operations)

เลือกหลักทรัพย์ที่ต้องการเพื่อใช้ในการวิเคราะห์

เลือกที่เมนูแบบ Drop down จากในส่วนของ Apply to เพื่อเลือกโหมดที่จะใช้ในการทำงาน : All symbols / Current / Filter

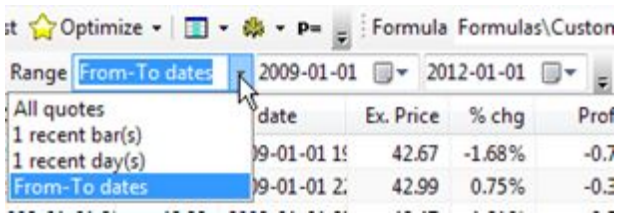


ระบุตัวกรอง (Defining Filter)

ถ้าใน Apply to เราเลือกเป็นที่กรองไว้ หน้าการวิเคราะห์จะทำงานด้วยสัญลักษณ์ที่เรากรองไว้ตามเงื่อนไขของเราจากหน้าต่าง Filter Setting ในการเปิดหน้าต่าง Filter Setting ให้กดปุ่มกรอง

กำหนดช่วงวันและเวลา (Defining date/time Range)

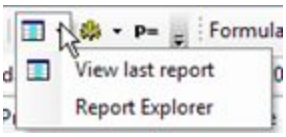
คลิกที่เมนูแบบ Drop down ในส่วนของ Range และเลือกโหมดที่ต้องการ : All quotes / N recent bas(s) / N recent day(s) / ตั้งแต่วันที่กำหนด



N นั้นแสดงถึงตัวเลขใดๆ ก็ตาม ยกตัวอย่างเช่นในการระบุช่วงเวลา 15 วันล่าสุด การเลือก 1 recent day(s) ในขั้นแรกและพิมพ์เลข 15 จากนั้นกด Enter คุณจะเห็นข้อความเปลี่ยนเป็น 15 recent day(s) โดยอัตโนมัติ โปรดจำไว้ว่าคุณไม่จำเป็นต้องพิมพ์อะไรเลย พิมพ์แค่ตัวเลขอย่างเดียวก็เพียงพอแล้ว

การดูผลรายงาน / เปิดหน้า Report Explorer (Viewing reports / Running Report Explorer)

ในการดูรายงานผลจากการทดสอบระบบล่าสุดให้คลิกที่ปุ่ม Report ส่วนการเปิดหน้า Report Explorer ให้คลิกที่เมนูแบบ Drop down ในส่วนของ Report และเมนูจะโชว์ด้านล่าง

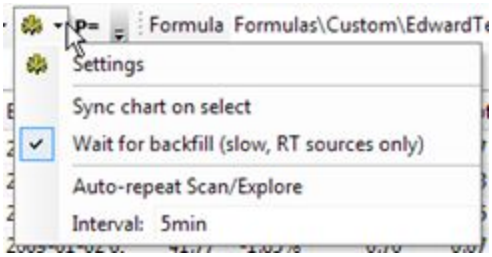


เปลี่ยนการตั้งค่า / ตัวเลือก (Changing settings / options)

การเปลี่ยนตั้งค่าการทดสอบระบบ คลิกที่ปุ่ม Settings เพื่อเปิดปิดตัวเลือกอย่างเช่น

- เชื่อมกับชาร์ทที่เลือก
- รอการทำงาน
- สแกนและค้นหาซ้ำอัตโนมัติ
- ไทม์เฟรมซ้ำอัตโนมัติ

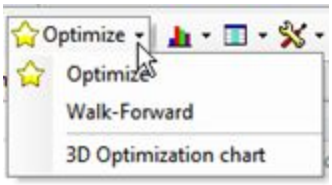
คลิกเลือกที่เมนู Drop down ในหมวด Setting และจะแสดงผลตามภาพด้านล่าง



การตั้งค่าไทม์เฟรมซ้ำสามารถแก้ไขได้จากฟิลด์ Interval โปรดจำไว้ว่าตัวเลขเปล่าๆ (เช่น 5) จะหมาย ถึงมีหน่วยเป็นนาที หากคุณต้องการหน่วยเป็นวินาทีจำเป็นต้องพิมพ์ 5sec หรือ 5s แล้วกด Enter

ใช้การเทสแบบ Walk Forward (Running Walk forward Test)

คลิกที่ลูกศรตรงปุ่ม Optimize เพื่อให้มันแสดงเมนูตามรูปภาพ จากนั้นเลือก Walk-Forward



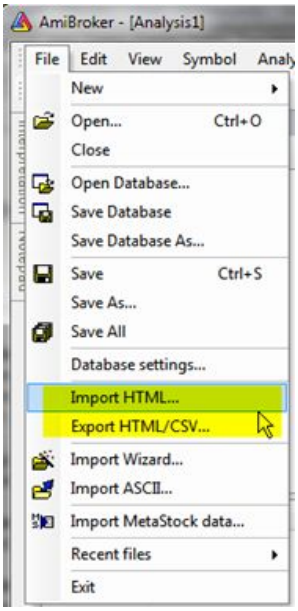
ผลลัพธ์ของการทดสอบ Walk forward จะแสดงผลอยู่ในแท็บ Walk Forward (อยู่ด้านล่างของหน้า Analysis)

แสดงผลการ Optimize แบบชาร์ท 3 มิติ (Displaying 3D optimization chart)

ถ้าต้องการให้แสดงผลการ Optimize แบบสามมิติ ขั้นแรกให้รันการ Optimize ซึ่งแน่นอนว่าจะประกอบไปด้วยพารามิเตอร์สองตัว จากนั้นคลิกที่ลูกศรตรงคำว่า Optimize เพื่อให้แสดงเมนูแบบภาพก่อนหน้า และเลือก 3D Optimization Chart

ส่งออกและนำเข้าผลลัพธ์ (Exporting and importing result list)

ในการส่งออกข้อมูลไปสู่ไฟล์ CSV หรือไฟล์ HTML ให้ใช้เมนู File > Export HTML/CSV (จากเมนูในหน้าหลัก) ส่วนการนำเข้าไฟล์ HTML ที่ส่งออกก่อนหน้านี้ให้ใช้เมนู File > Import HTML... เหมือนกับรูปที่แสดงอยู่ด้านล่าง จำไว้ว่าเมนูเหล่านี้จะปรากฏก็ต่อเมื่อเปิดหน้า New Analysis ไว้อยู่


AmiBroker - [Analysis1]
File
Edit
View
Symbol
New
Open... Ctrl+O
Close
Open Database...
Save Database
Save Database As...
Save Ctrl+S
Save As...
Save All
Database settings...
Import HTML...
Export HTML/CSV...
Import Wizard...
Import ASCII...
Import MetaStock data...
Recent files
Exit

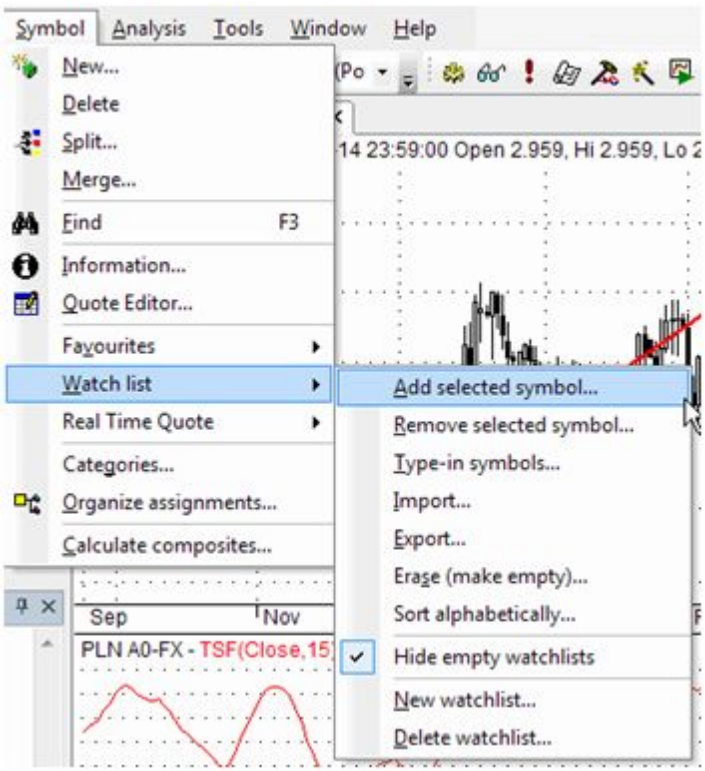
การจัดการ Watch List

(Working with watch lists)

การ เพิ่ม/ลบ Watchlist (Adding / removing watch lists)

คุณสามารถที่จะ เพิ่ม/ลบ watch list ได้โดยการไปที่ Symbol->Watch List->New Watchlist ในกรณีที่คุณต้องการสร้าง watch list ใหม่ หรือไปที่ Symbol->Watch List->Delete Watch list menu หากคุณต้องการลบ watch list

หมายเหตุ : หากคุณได้ทำการปรับแต่ง แถบเมนูใดๆก็ตามลงไปแล้ว สิ่งที่คุณต้องทำเพื่อให้แถบเมนูกลับมาเป็นเหมือนเดิมก็ คือ ไปที่ Tools->Customize และเลือก"Menu Bar" จากนั้นกดปุ่ม "Reset" เพื่อให้แถบเมนูกลับไปเป็นเหมือนเดิม



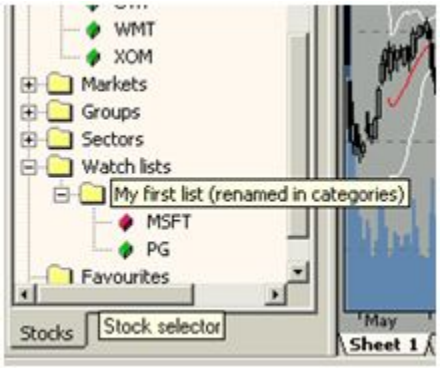
การนำ tickers เข้าสู่ watch lists (Adding tickers to watch lists)

ให้คุณเข้าไปที่ Watch List->Add selected symbol

หลังจากทำตามตัวอย่างข้างต้น หน้าต่างของ รายการwatch list จะปรากฏขึ้นตามรูปข้างล่าง :

ตรงนี้ให้คุณเลือก watch list ที่คุณต้องการจะเพิ่มหลักทรัพย์นั้นๆเข้าไป(เลือก watch list ที่ต้องการเพิ่มชื่อหลักทรัพย์เข้าไป) หมายเหตุ : คุณสามารถที่จะเพิ่มชื่อหลักทรัพย์หนึ่งชื่อ ไปยังหลายๆ watch list

พร้อมๆกันได้ โดยการกดปุ่ม CTRL ค้างไว้และเลือก watch list ที่คุณต้องการจะเพิ่มลงไป หลังจากนั้นกดปุ่ม OK หลักทรัพย์ที่คุณเลือกไว้ก็จะมีรายอยู่ไปแสดงอยู่ใน watch list ที่คุณได้ทำการเลือกไว้ :



นอกจากนี้คุณยังสามารถพิมพ์ชื่อหลักทรพย์เพื่อนำเข้าสู่ watch list ได้โดยตรงโดยใช้ Symbol->Watch list->Type-in หากคุณต้องการเพิ่มชื่อหลักทรัพย์มากกว่าหนึ่งชื่อในครั้งเดียว คุณต้องใส่เครื่องหมาย , ระหว่างชื่อของหลักทรัพย์

การเรียงลำดับ tickers in a watch list (***Sorting tickers in a watch list)***

คุณสามารถจัดเรียงชื่อใน watch list ตามลำดับตัวอักษรได้โดยการ - คลิกขวาใน watch และ เลือก "Sort Alphabetically"

การลบ tickers ออกจาก watch list(***Removing tickers from watch lists)***

การลบรายชื่อหลักทรัพย์ออกจาก watch list นั้นง่ายพอๆกับการเพิ่มรายชื่อหลักทรัพย์เข้าไปใน watch list แค่เพียงคลิกขวาบนรายชื่อหลักทรพย์ที่คุณต้องการจะลบ และเลือก Remove from watch list(s)

การลบ watch list หรือ ล้างรายชื่อที่อยู่ใน watch list ของคุณ(*Erasing watch lists)*

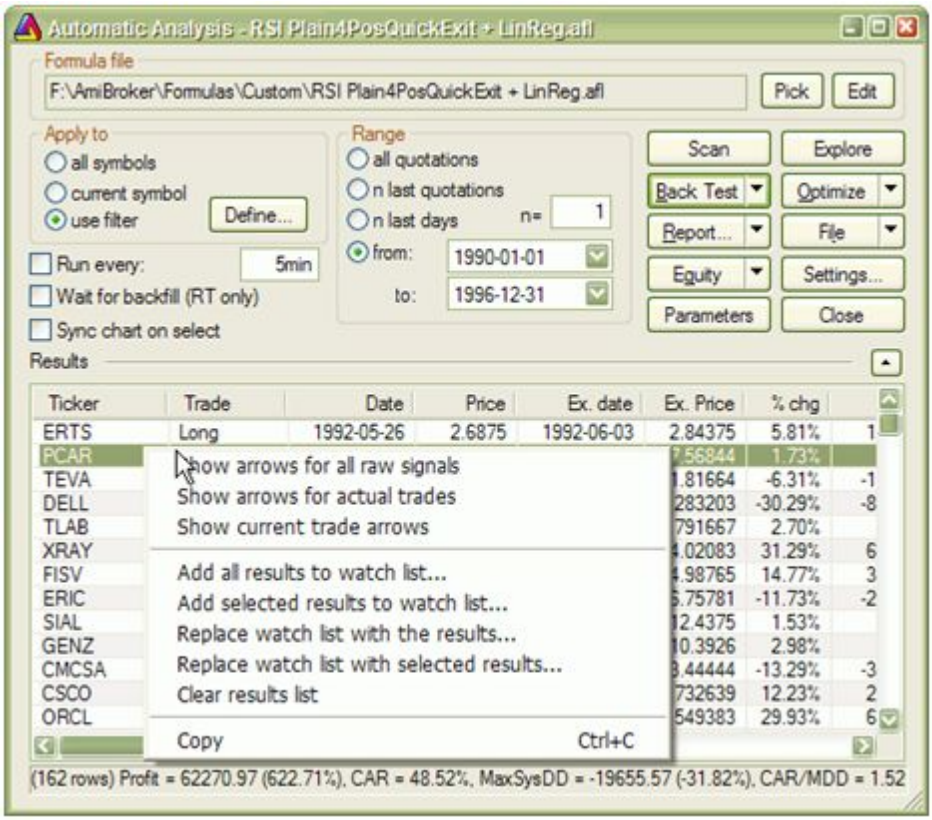
บางครั้งคุณอาจต้องการล้าง (หรือลบ) watch list ของคุณ สิ่งที่คุณต้องทำก็แค่เพียงเลือก Symbol -> Watch list -> Erase (empty)

ซ่อน/ปิดการซ่อน watch list ต่างๆที่ว่างเปล่า (Hiding/Unhiding empty watch lists)

โดย default แล้ว watch lists ที่ว่างเปล่า จะแสดงให้เห็นไว้อยู่แล้ว แต่คุณสามารถทำการซ่อน watch list เหล่านั้นได้โดย การคลิกขวาที่หน้าต่าง watch list และ เลือก"Hide Empty Watchlists" และหากคุณต้องการที่จะปิดการซ่อนของมันคุณก็เพียงแค่คลิกปุ่มเดิมอีกครั้งหนึ่ง

การใช้ watch lists ในหน้าต่าง Automatic analysis (Using watch lists in Automatic analysis window)

Amibroker ช่วยให้คุณทำงานได้ง่ายขึ้น ในการจัดเก็บผลการสแกน ผลการ backt และ ผลการ exploration เก็บไว้ใน watch list ด้วยเพียงแค่คลิกเดียว แค่เพียงคุณรัน AFL formula ตามปกติ และคลิกที่รายการผลลัพธ์ที่ออกมา เพื่อให้เมนูด้านล่างนี้แสดงขึ้นมา :

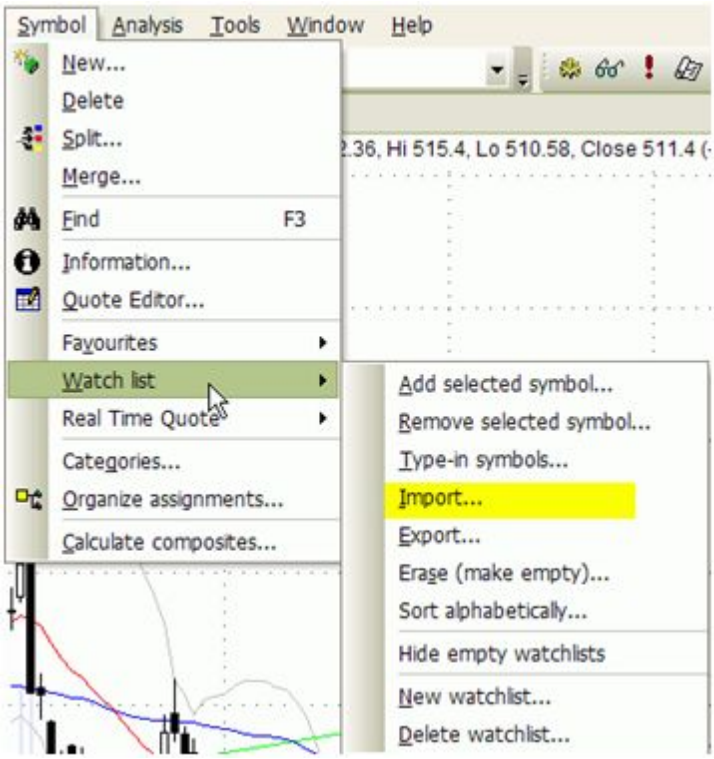


เมื่อคุณเลือก Add all/selected results to watch list watch list ที่คุณเลือกนั้นจะมีรายชื่อของหลักทรัพย์ในหน้าต่างผลลัพธ์ของคุณทั้งหมด(ไม่ว่าจะสแกน หรือ backtest อะไรก็แล้วแต่) นอกจากนี้คุณสามารถที่จะใช้ตัวเลือก Replace watch list with the results/selected results หากคุณใช้ตัวเลือกนี้รายชื่อหลักทรัพย์ที่อยู่ใน watch list นั้นทั้งหมดจะถูกแทนที่ด้วยผลลัพธ์ที่คุณได้

จะ นำเข้า/นำออก watch list จาก/ไปยัง ไฟล์ ได้อย่างไร

การนำเข้า watch list จากไฟล์ใดๆ (IMPORT WATCH LIST FROM FILE)

1. เลือก Symbol->Watch List->Import menu หรือ คลิกขวาที่ watch list และเลือก Import



2. เลือกที่อยู่ของ watch list

3. ในหน้าจอแสดงผลจะแสดงเฉพาะไฟล์ที่ มีนามสกุลดังต่อไปนี้เท่านั้น TLS, .LST, .TXT หรือ .CSV

4. คลิก OK

*นำออก WATCHLIST ไปยังไฟล์ (**EXPORT WATCHLIST TO FILE**)*

1. เลือก Symbol->Watch List->Export menu.

หรือ คลิกขวาที่ watch list และ เลือก Export

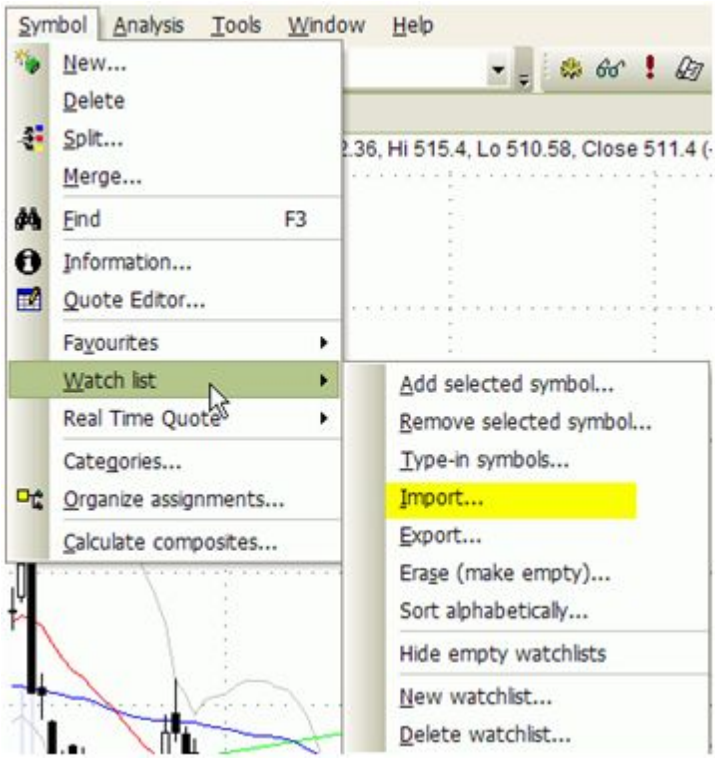
2. เลือก source watch list และเปลี่ยนไปที่ "External data source"

3. ในไฟล์โต้ตอบให้เลือกไฟล์ที่จะทำการ export ไฟล์จะถูกสร้างโดยเป็นไฟล์ ASCII ด้วยหนึ่ง ticker symbol ต่อหนึ่ง line

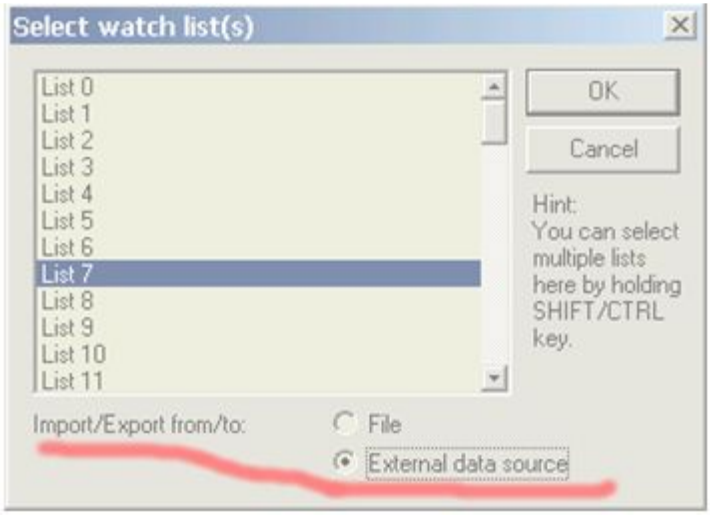
จะ นำเข้า watch list จากฐานข้อมูลภายนอก หรือ จะนำออก watch list ไปสู่ฐานข้อมูลภายนอกได้อย่างไร (How to import/export watch list from/to external database)

การนำเข้า Family จาก Fasttrack (IMPORT FAMILY FROM FASTTRACK)

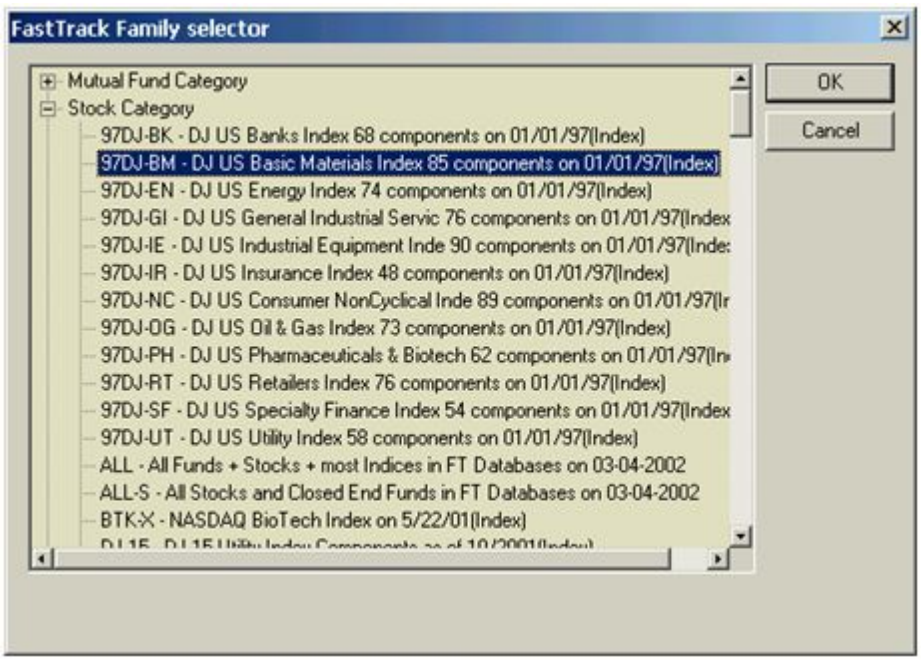
1. เลือก Symbol->Watch List->Import เมนู หรือ คลิกขวาที่ watch หลังจากนั้นเลือก Import



2. เลือกที่ตั้งของ watch list และเปลี่ยนไปที่ "External data source"



3.ในกล่องข้อความโต้ตอบจะปรากฏจะ unfold อยู่หนึ่ง category จากนั้นให้เลือก family จากที่ที่คุณต้องการที่จะนำเข้า symbols

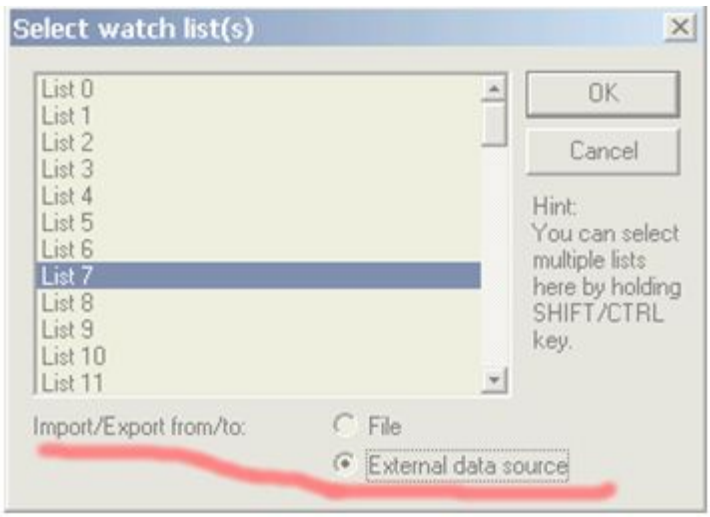


4. คลิก OK

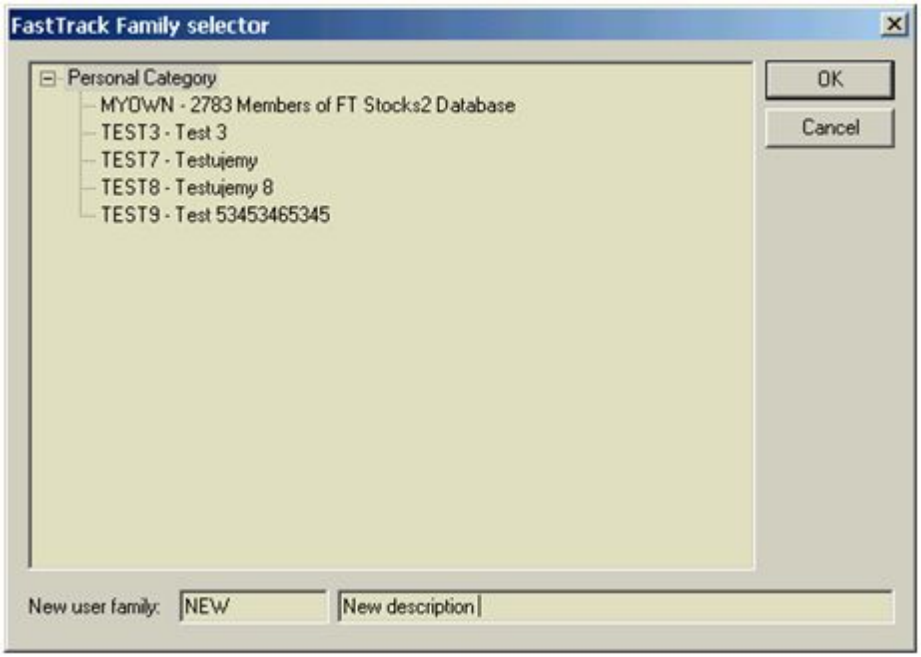
การนำออก Family ไปยัง Fasttrack (EXPORT WATCHLIST TO FASTTRACK FAMILY)

1. เลือก Symbol->Watch List->Export เมนู
หรือคลิกขวาที่ watchหลังจากนั้นเลือก Export

2. เลือก source ของ watch list และ เปลี่ยนไปที่ "External data source"



3. ตอนนี้พิมพ์ชื่อ personal family ที่เราต้องการ ลงใน "New user family" (และใส่คำอธบาย เพิ่มเติมลงในช่องถัดไปด้านขวามือ) หรือ เลือก personal family จาก list ที่มีอยู่เดิม



ทำความเข้าใจว่า AFL นั้นทำงานอย่างไร

(Understanding how AFL works)

บทนำ (Introduction)

หนึ่งในส่วนที่สำคัญที่สุดของ AFL ก็คือ มันเป็นภาษาที่มีการประมวลผลอย่างหลากหลาย มันทำงานอยู่บน อาร์เรย์ (หรือแถว / เวกเตอร์) ของข้อมูล วิธีการทำงานของมันนั้น ค่อนข้างที่จะคล้ายคลึงกับการทำงานของสเปรดชีตที่เป็นที่นิยมต่างๆ (เช่น Microsoft Excel) ถ้าหากคุณคุ้นเคยกับใช้งาน MS Excel อยู่แล้ว คุณก็จะทำความเข้าใจการใช้งาน AFL ได้อย่างรวดเร็ว - ในความเป็นจริงตัวอย่างในบทความนี้ถูกสร้างขึ้นทั้งหมดโดยใช้ MS Excel

อาร์เรย์ คือ อะไร? (What is an Array?)

อาร์เรย์ คือ รายการ/แถว อย่างง่ายของค่าต่างๆ ในหนังสือบางเล่มก็อาจจะเรียกมันว่าเวกเตอร์ ตัวเลขที่อยู่หน้าแถวแต่ละแถวนั้นเป็นตัวแทนของแต่ละอาร์เรย์ Amibroker จะเก็บฐานข้อมูล 6 อาร์เรย์ต่อ symbol หนึ่งอัน ซึ่งใน 6 อาร์เรย์นี้ก็จะประกอบไปด้วย 1.ราคาเปิด 2. ราคาต่ำสุด 3.ราคาสูงสุด 4. ราคาปิด 5.โวลุ่ม 6.จำนวนสถานะคงค้างเขียน หรือจะเขียนเป็นตัวย่อที่เราคุ้นเคยในรูป O, L, H, C, V, OI

	Bar	1	2	3	4	5	6	7	8	9	10
1	Open	1,23	1,24	1,21	1,26	1,24	1,29	1,33	1,32	1,35	1,37

รูปแสดงตัวอย่างของ อาร์เรย์ของราคาเปิด

โดยอาร์เรย์อื่นๆนอกเหนือจากนี้ ก็จะถูกคำนวณจาก 6 อาร์เรย์นี้ โดยใช้สูตรที่สร้างขึ้นใน AFL (อาร์เรย์เหล่านี้จะไม่เก็บไว้ในฐานข้อมูล แต่จะมีคำนวณในกรณีที่จำเป็นต้องใช้)

แต่ละค่าในอาร์เรย์มีวันที่เกี่ยวข้องกับมัน เมื่อคุณย้ายเคอร์เซอร์ไปที่กราฟแท่งเทียนรายวัน (Preferences -> Miscellaneous Tab - > Price data tool tips) สี่เหลี่ยมเล็ก ๆ สีเหลืองปรากฏ AFL จะแสดงค่า Open high low Close volume

การทำงานของอาร์เรย์ - ทำไม AFL ถึงทำงานได้รวดเร็วมากนัก(Processing arrays - why is AFL so fast?)

การทำงานของคำสั่งมีการทำงานดังต่อไปนี้:

```
MyVariable = ( High + Low )/2;
```

เมื่ออยู่ระหว่างการประเมินคำสั่งดังตัวอย่าง (High + Low) / 2 Amibroker ไม่จำเป็นต้องตีความรหัสนี้ในแต่ละ bar จริงๆแล้วมันเรียกใช้อาร์เรย์ High และ อาเรย์ Low แล้วทำการคำนวณเพื่อสร้าง

ตัวแปรใหม่ในขั้นตอนเดียว การทำงานบนอาร์เรย์ในครั้งเดียวนั้นจะมีการดำเนินการด้วยความเร็วเต็มรูปแบบ

ลองมาดูรายละเอียดให้มากขึ้นดีกว่า - ดูที่รูปข้างล่าง .. เมื่อ AFL อ่านค่า (High + Low) / 2 ในครั้งแรก มันจะใช้ค่าของอาร์เรย์ High (1) และ อาร์เรย์ Low (2) จากนั้นจึงประมวลผล (ในขั้นตอนเดียว) เป็น อาร์เรย์ชั่วคราว (3) และ จากนั้นก็จะสร้างอาร์เรย์สุดท้าย (4) ด้วยการหารอาร์เรย์ชั่วคราวด้วยสอง ผลที่ได้นี้จะถูกใส่ไว้ในตัวแปร myVariable

	Bar	1	2	3	4	5	6	7	8	9	10
1	**High** *(built-in array)*	1,24	1,27	1,25	1,29	1,25	1,29	1,35	1,35	1,37	1,29
2	**Low** *(built-in array)*	1,20	1,21	1,19	1,20	1,21	1,24	1,30	1,28	1,31	1,27
3	**High+Low** *(temporary array created during evaluation)*	2,44	2,48	2,44	2,49	2,46	2,53	2,65	2,63	2,68	2,46
4	**(High+Low) /2** *(gets assigned to MyVariable)*	1,22	1,24	1,22	1,245	1,23	1,265	1,325	1,315	1,34	1,23

Moving averages เงื่อนไขของคำสั่ง (Moving averages, conditional statements)

โปรดพิจารณาโค้ด ดังต่อไปนี้:

```
Cond1 = Close > MA( Close, 3 );
Cond2 = Volume > Ref( Volume, -1 );
Buy = Cond1 AND Cond2;
Sell = High > 1.30;
```

โค้ดนี้จะซื้อเมื่อราคาปิดมากกว่า Moving average 3 วัน และ Volume มากกว่าวันที่แล้ว และขายเมื่อราคาสูงสุดในวันสูงกว่า 1.3

ถ้าหากใน AFL โค้ด คุณต้องเห็นเมื่อ ราคาปิดของวันสูงกว่า simple moving average 3 วัน AFL จะทำงานโดยผ่านอาร์เรย์ของราคาปิด โดยสร้างอาร์เรย์ใหม่ที่เรียกว่า MA(close,3) ขึ้นมาเพื่อ symbol จะได้ทำการวิเคราะห์ได้ ในแต่ละเซลล์ของอาร์เรย์ใหม่ สามารที่จะทำการเปรียบเทียบหนึ่งต่อหนึ่งกับอาร์เรย์

ของราคาปิดได้เลย ยกตัวอย่างเช่น อาร์เรย์ที่ชื่อว่า Cond1 จะถูกสร้างโดยวิธีการนี้ ในแต่ละเซลล์ที่ราคาปิดนั้นสูงกว่าอาร์เรย์ MA(close,3) ค่าในเซลล์ ของอาร์เรย์ใหม่ที่ชื่อว่า Cond1 จะมีค่าเท่ากับ 1 และถ้าหากค่าของราคาไม่อยู่สูงกว่า ค่าในเซลล์ ของอาร์เรย์ใหม่ที่ชื่อว่า Cond1 จะมีค่าเท่ากับ 0

AFL ยังสามารถมองไปข้างหน้าหรือถอยหลังจำนวนช่องของเซลล์ในอาร์เรย์ โดยใช้ฟังก์ชั่น Ref (จากภาพด้านล่าง ดูแถว 6 ที่อาร์เรย์ชั่วคราวจะถูกสร้างขึ้นโดยใช้โวลุ่มจากวันก่อนหน้า)

ในแถว 9 อาร์เรย์ใหม่ที่เรียกว่า Cond2 ได้ถูกสร้างขึ้นโดยการเปรียบเทียบค่าของแต่ละเซลล์ในอาร์เรย์ Volume หากตรงกับการตั้งค่าในเซลล์ Cond2 ก็จะแสดง '1' ถ้าเป็นจริงและ '0' ถ้าเป็นเท็จ

แถวที่ 10 แสดงให้เห็นว่าอาเรย์ที่เรียกว่า 'Buy' ที่สร้างขึ้นโดยเปรียบเทียบค่าใน Cond1 และ Cond2 ถ้า Cond1 มีค่า '1' และ สอดคล้องกันกับ Cond2 แล้วอาร์เรย์นี้จะแสดงค่า 1 แต่หากไม่สอดคล้อง หรือสอคล้องกันในกรณี 0 , 0 ก็จะให้ค่าเป็น 0

	Day	1	2	3	4	5	6	7	8	9	10
1	**Open**	1,23	1,24	1,21	1,26	1,24	1,29	1,33	1,32	1,35	1,37
2	**High**	1,24	1,27	1,25	1,29	1,25	1,29	1,35	1,35	1,37	1,29
3	**Low**	1,20	1,21	1,19	1,20	1,21	1,24	1,30	1,28	1,31	1,27
4	**Close**	1,23	1,26	1,24	1,28	1,25	1,25	1,31	1,30	1,32	1,28
5	**Volume**	8310	3021	5325	2834	1432	5666	7847	555	6749	3456
6	**Ref(Volume, -1)** *(temporary array created during eval)*	Null	8310	3021	5325	2834	1432	5666	7847	555	6749
7	**MA(Close, 3)** *(temporary array created during eval)*	Null	Null	1,243	1,260	1,257	1,260	1,270	1,287	1,310	1,300
8	**Cond1 = Close < MA(close,3)** *(gives 1 (or true) if condition met, zero otherwise)*	Null	Null	1	0	1	1	0	0	0	1

ลองเพิ่มความซับซ้อนเข้าไปอีกหน่อย (Getting little bit more complex)

ตัวอย่างข้างต้นนั้นเป็นตัวอย่างอย่างง่าย ตอนนี้เราจะมาอธิบาย 3 สิ่งที่ดูเหมือนจะสร้างความสับสนให้แก่ผู้ใช้ดูกว่า

- **referencing selected values (SelectedValue, BeginValue, EndValue, LastValue)**
- **IIF function**
- **AMA function**

อย่างที่เขียนในในหัวข้อ Tutorial: Basic charting guide แล้ว คุณสามารถที่จะเลือก quote ไหนก็ได้จาก chart และคุณสามารถเลือก From-To range ได้ บาร์ที่ถูกเลือก โดยแส้นแนวตั้งนั้น จะถูกเรียกว่า"selected" bar ในขณะที่ start และ end bar ของช่วงที่ถูกเลือกดังกล่าวจะถูกเรียกว่า "begin" และ "end" bar โดย AFL จะมีฟังก์ชันพิเศษที่สามารถอ้างอิงค่าของอาร์เรย์ที่ถูกเลือกได้ ฟังก์ชันนี้ถูกเรียกว่า SelectedValue, BeginValue และ EndValue นอกจากนี้มันยังมีอีกหนึ่งฟังก์ชันที่เรียกว่า LastValue ซึ่งจะได้ค่าอาร์เรย์ของบาร์สุดท้าย ทั้ง 4 ฟังก์ชันนี้ใช้องค์ประกอบของอาร์เรย์ที่บาร์ที่ถูกเลือก และ return SINGLE NUMBER ของค่าอาร์เรย์ในจุดที่ถูกเลือกนั้น สิ่งนี้จะช่วยในการคำนวณสถิติบางประการเกี่ยวกับจุดที่ถูกเลือกนั้นๆ

ยกตัวอย่างเช่น:

EndValue(Close) - BeginValue(Close)

มันจะให้ค่าที่เปลี่ยนแปลงไปของราคาปิดจากช่วงที่คุณเลือกไว้

เมื่อมีตัวเลขถูกดึงค่า จากฟังก์ชันเหล่านี้ มันจะถูกนำไปเปรียบเทียบกับ array หรือ ตัวเลขใดๆที่เกี่ยวข้องกันทางคณิตศาตร์ และ array จะถูกดำเนินการด้วยตัวมันเอง คล้ายกับ ตัวเลขที่ถูกวัดด้วย องค์ประกอบทั้งหมด ด้านล่างนี้ คือ ตารางตัวอย่าง (แถว 2, 6, 7) สีเขียวหมายถึง "begin" bar และ สีแดงหมายถึง "end" bar และสุดท้ายนี้ bar ที่ถูกเลือกจะเป็นสีน้ำเงิน

	Day	1	2	3	4	5	6	7	8	9	10
1	Open	1,23	1,24	1,21	1,26	1,24	1,29	1,33	1,32	1,35	1,37
2	BeginValue(Open)	1,24	1,24	1,24	1,24	1,24	1,24	1,24	1,24	1,24	1,24
3	EndValue(Open)	1,32	1,32	1,32	1,32	1,32	1,32	1,32	1,32	1,32	1,32
4	SelectedValue(Open)	1,21	1,21	1,21	1,21	1,21	1,21	1,21	1,21	1,21	1,21
5	LastValue(Open)	1,37	1,37	1,37	1,37	1,37	1,37	1,37	1,37	1,37	1,37
6	Close	1,22	1,26	**1,23**	1,28	1,25	1,25	1,31	1,30	1,32	1,28
7	Close <= BeginValue(Open)	1	0	1	0	0	0	0	0	0	0
8	result = IIF(Close <= BeginValue(Open), Close, Open);	1,22	1,24	1,23	1,26	1,24	1,29	1,33	1,32	1,35	1,37
9	Period	2	3	4	2	3	5	2	3	4	2
10	Factor = 2/(Period+1)	0,667	0,500	**0,400**	0,667	0,500	0,333	0,667	0,500	0,400	0,667
11	1 - Factor	0,333	0,500	0,600	0,333	0,500	0,667	0,333	0,500	0,600	0,333
12	AMA(Close, Factor)	0,8125	1,0363	**1,1138**	1,2234	1,2367	1,2399	1,2853	1,2927	1,3036	1,2866

IIF(condition, truepart, falsepart) ฟังก์ชัน

มันจะแสดงผลของค่า truepart หรือไม่ก็ falsepart ตามเงื่อนไขที่กำหนด จากที่เราเห็นในแถวที่ 8 เมื่อเงื่อนไขถูกต้องจะได้ค่าเป็นราคา Close แต่ถ้าเงื่อนไขนั้นผิดจะให้ค่าของราคา Open ในเคสนี้ IIF function จะให้ค่าไม่ Close ก็ Open

หมายเหตุ ทั้งอาร์เรย์ truepart และ falsepart พวกมันจะได้รับการประเมินโดยไม่คำนึงถึงสภาพ (ดังนั้นนี้ไม่ได้เป็น IF-ELSE ธรรมดาทั่วไป แต่มันเป็นฟังก์ชันที่ return ค่าเป็นอาร์เรย์)

AMA(array, factor) ฟังก์ชัน

เป็นฟังก์ชันที่ดูเหมือนว่าจะมีปัญหามากที่สุดในการทำความเข้าใจ แต่ในความเป็นจริงมันเป็นเรื่องง่ายมาก มันทำงานในลักษณะของการเรียกซ้ำ (recursive) ก็หมายความว่ามันจะใช้ค่าก่อนหน้ามาเป็นส่วนหนึ่งในการคำนวณค่าปัจจุบันด้วย เพื่อให้เข้าใจมากขึ้นมาดูตัวอย่างการคำนวณค่าของ AMA ในคอลัมน์ที่ 3 มันจะใช้ค่า Close จากคอลัมน์ 3 (1,23) ค่า Factor (0,4) และค่าของตัวมันเองก่อนหน้าซึ่งก็คือ AMA

(1,0363) คูณด้วย (1-Factor = 0,6) จะมีการคำนวณดังนี้ (ปัดเลขทศนิยม 4 ตำแหน่ง) 1,23 * 0,4 + 1,0363 * 0,6 = 1,1138

ถ้าคุณดูที่ตัวเลขในแถวที่ 12 คุณอาจพบว่าค่าเหล่านี้มีลักษณะเหมือนค่า moving average ของราคาปิด ซึ่งมันเป็นความจริง เราจึงนำเสนอวิธีการคำนวณ EMA โดยใช้ฟังก์ชั่น AMA

การสร้างลูปใหม่ (New looping)

Amibroker รุ่น 4.40 นำความสามารถในการวนของโค้ด โดยใช้ For และ while loop และใช้คำสั่ง IF-else เพื่อควบคุมการทำงานของการไหล ซึ่งสามารถทำได้ 2 วิธีด้วยกัน วิธีแรก คือ ตามที่ได้กล่าวไว้ก่อนหน้าด้านบน ส่วนอีกวิธีหนึ่งก็คือการสร้าง LOOPS เพื่อการทำงานที่ซับซ้อนมากยิ่งขึ้น ในตัวอย่างดังต่อไปนี้จะเป็นการสร้าง ตัวแปร exponential averaging เหมือนตัวอย่างด้านบน แต่คราวนี้จะใช้วิธีการ LOOPS แทน โดยใช้โค้ดต่อไปนี้

```
Period = ... การคำนวณบางอย่าง
vaexp[ 0 ] = Close[ 0 ]; // ค่าเริ่มต้นค่าแรก
 for( i = 1; i < BarCount; i++ )
{
        // คำนวณค่าตัวแปรของ เฟคเตอร์การปรับให้เรียบ (smoothing factor)
         Factor = 2/(Period[ i ] + 1 );

        //คำนวณค่าของ i-th ที่เป็นองค์ประกอบของอาร์เรย์
        // ใช้ราคาปิดของบาร์นั้นๆ ( close[ i ] ) และค่าเฉลี่ยของบาร์ก่อนหน้า ( vaexp[ i - 1 ] )
        vaexp[ i ] = Factor * Close[ i ] + ( 1 - Factor ) * vaexp[ i - 1 ];
}
```

มันจะคล้ายกับการเขียนโปรแกรมภาษาอื่นๆ เช่น C / Pascal ดังนั้นคนที่มีประสบการณ์เกี่ยวกับการเขียนโปรแกรมมาบ้าง อาจพบว่ามันง่ายที่จะเข้าใจ

ถ้าคุณเป็นมือใหม่ผมขอแนะนำให้คุณเรียนรู้ การประมวลผลที่หลากหลายก่อนที่จะ ใช้การวนลูปที่ซับซ้อนมากขึ้น

หากคุณกำลังมีปัญหาในการเขียนโปรแกรม ผมขอแนะนำให้คุณสร้างอาร์เรย์ในตัวอย่าง ใน Excel เพื่อตัวคุณเอง หากเป็นปัญหาอีก ลองรับความช่วยเหลือจากเพื่อนคุณสิ - โดยเฉพาะอย่างยิ่งถ้าเพื่อนของคุณเป็นนักบัญชี

--- ขอบคุณคุณ Geoff Mulhall เป็นอย่างสูงสำหรับหัวข้อ original article in the newsletter ซึ่งนำมาใช้เป็นพื้นฐานของคู่มือฉบับนี้---

สร้างอินดิเคเตอร์ด้วยตัวคุณเอง

(Creating your own indicators)

มีสองวิธีในการสร้าง indicators ของคุณเอง:

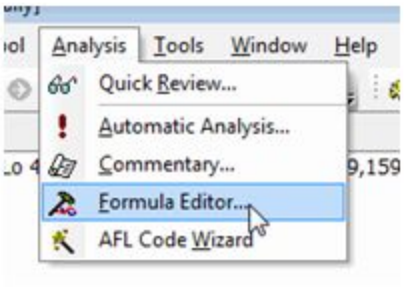
1) ใช้อินเตอร์เฟซการลากและวาง

2) โดยการเขียนสูตรของคุณเอง

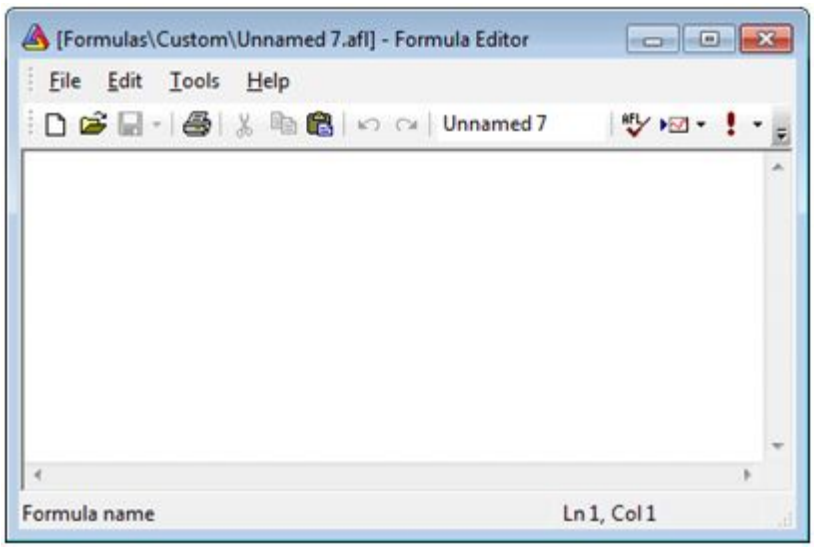
วิธีแรก ใช้อินเตอร์เฟซการลากและวางนั้นเป็นเรื่องที่ง่ายมาก ๆ
วิธีที่สอง เกี่ยวข้องกับการเขียนสูตรใน AFL (Amibroker Formular Language) ในตัวอย่างนี้เราจะกำหนด "Indicator" ที่จะแสดงกราฟปริมาณเส้น (line volume) (ตรงข้ามกับตัวปริมาณกราฟแท่ง)

เพียงทำตามขั้นตอนเหล่านี้

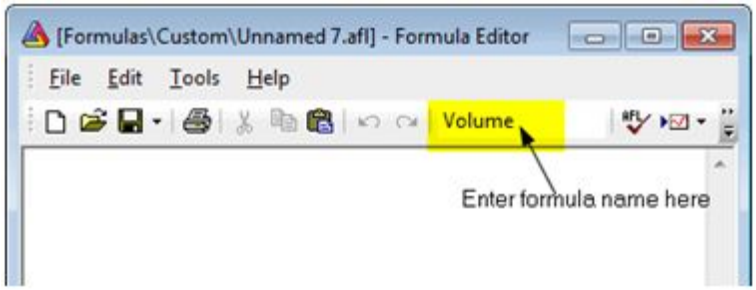
1. เลือก Analysis->Formula Editor จากแถบเมนูที่แสดงด้านล่าง:



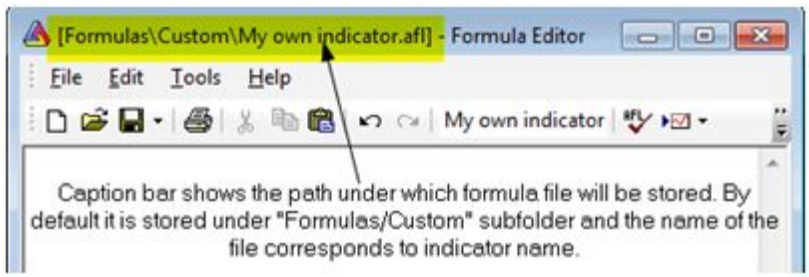
2. คุณจะเห็นหน้าต่างข้อความดังที่แสดงในรูปด้านล่าง บนจอของคุณ :



3. ตอนนี้คลิกหนึ่งครั้ง ในช่องสีเหลืองดังรูปด้านล่างเพื่อใส่ชื่อของ Indicator ที่คุณกำลังจะสร้าง:



ตอนนี้คุณสามารถที่จะแก้ไขชื่อของ Indicator ของตัวคุณเองได้แล้วในที่นี้ใช้ชื่อว่า "My own indicator" หลังจากที่คุณกด ENTER ชื่อจะทำการอัพเดต ด้วยชื่อใหม่ที่คุณตั้ง แสดงดังรูปด้านล่าง:

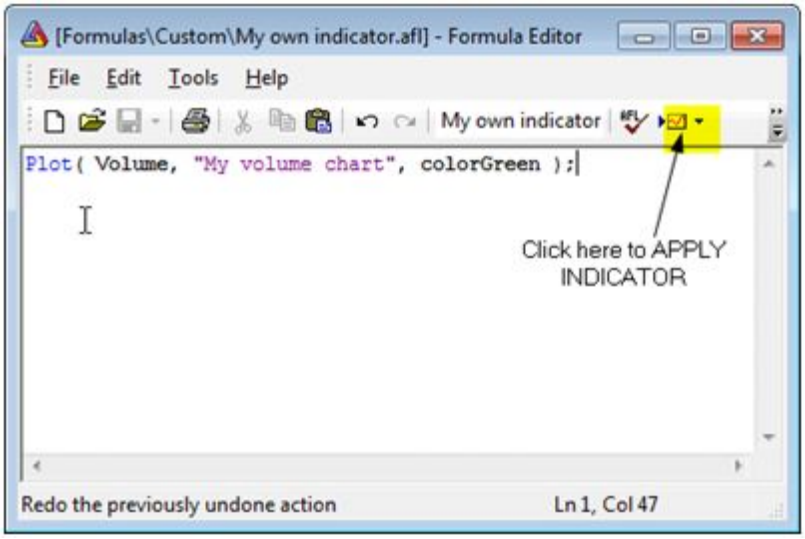


4. ที่นี้เขียนโค้ดดังต่อไปนี้:

```
Plot( Volume, "My volume chart", colorGreen );
```

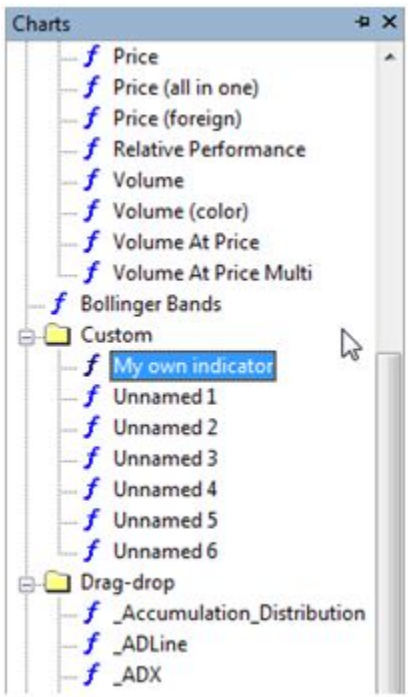
สูตรนี้จะสั่งให้ Amibroker พล็อตในตัวอาร์เรย์ Volume พารามิเตอร์ที่สองระบุชื่อ Indicator ที่จะทำการพล็อต และ พารามิเตอร์ที่สามกำหนดสี

ภาพด้านล่างแสดงแก้ไขสูตรหลังจากโค้ด:

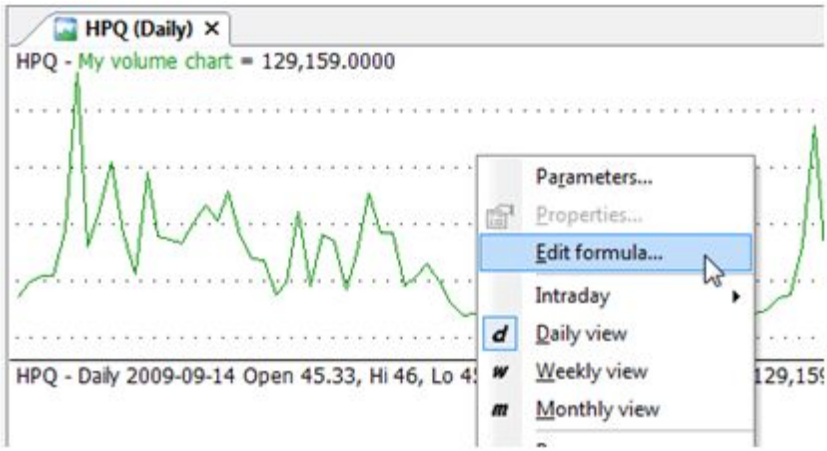


5. คลิก Apply (หรือ เลือก Tools->Apply indicator เมนู)

ตอนนี้ Indicator ที่คุณได้เขียนจะแสดงผลออกมาเป็นกราฟ นอกจากมันยังจะเก็บ Indicator ของคุณไว้เป็นสูตรอีกด้วย



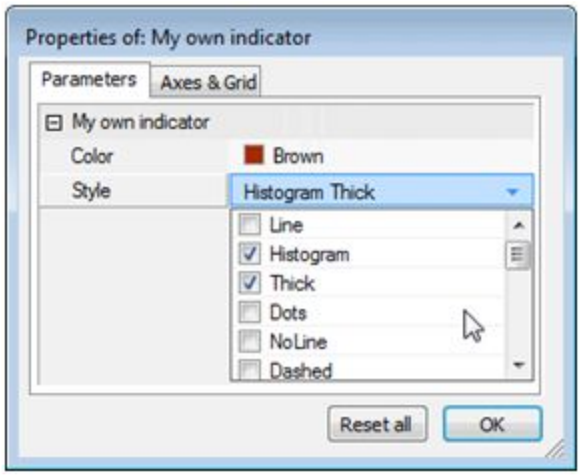
สามารถแก้ไขสูตรได้โดย การคลิกขวาบน pane ของกราฟ และเลือก Edit Formula (หรือแค่กด Ctrl+E)



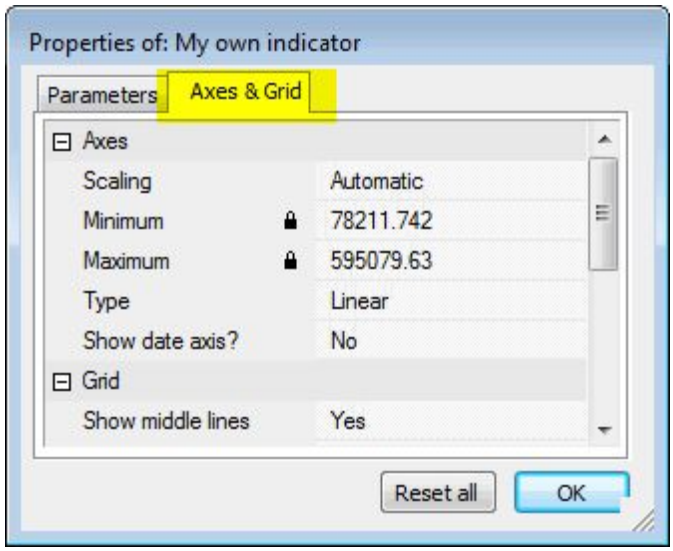
และ ปรับปรุงสูตรโดยการ:

Plot(Volume, "My volume chart", ParamColor("Color", colorGreen),
ParamStyle("Style", 0, maskAll));

หลังจากนั้นเลือก Apply indicator เพื่อให้มันแสดงผลใหม่ เพื่ออัพเดตสูตรที่เราได้ทำการแก้ไข ตอนนี้คลิกขวาที่ pane ของกราฟอีกครั้ง และเลือก Parameters (หรือ กด Ctrl+R)



ในแท็ป "Axes & Grid" คุณจะสามารถเปลี่ยนการตั้งค่าสำหรับแกนในส่วนนี้ และ การใช้กราฟที่แสดงผลในลักษณะต่างๆดังนี้ styles, colors, title และ parameter ใน Indicator นั้นๆ



การใช้ปรับใช้ styles, colors, titles และ parameters ในกราฟ Indicator

(Using graph styles, colors, titles and parameters in Indicators)

Amibroker สามารถที่จะปรับแต่งรูปแบบต่างๆ และ ปรับสีของกราฟในตัวชี้วัดที่กำหนดเองได้ คุณสมบัติเหล่านี้นั้น ช่วยเพิ่มความยืดหยุ่นในการออกแบบตัวชี้วัดของคุณได้มากเลยทีเดียว บทความนี้จะอธิบายวิธีการใช้รูปแบบและสี ดังนี้

Plot() function

เป็นฟังก์ชั่นที่ใช้ในการพล็อตกราฟ มันประกอบด้วย argument ทั้งหมดถึง 9 argument ด้วยกัน โดยที่ argument ที่จำเป็นต้องใส่ คือ 3 อันดับแรก ส่วนอีก 6 argument ที่เหลือ ขึ้นอยู่กับความต้องการของผู้ใช้

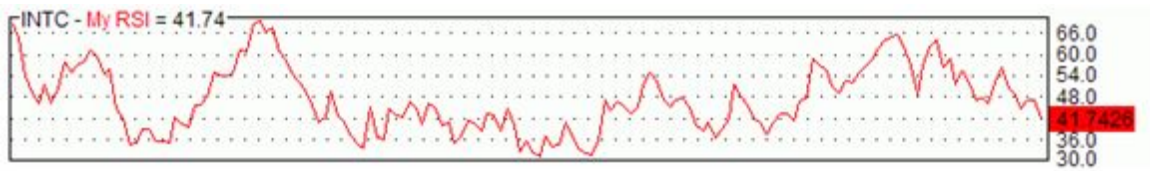
Plot (Array, “Name”, “Color”, style = StyleLine , MINVALUE = Null, MAXVALUE = Null, XShift = 0 ZOrder = 0 width = 1)

- array parameter มายถึง ข้อมูลที่จะใช้ในการ plot
- name parameter หมายถึง ชื่อของของ indicator นี้
- color หมายถึง สีของ indicator นี้
- style หมายถึง รูปแบบของ Chart ใน indicator นี้ ถ้าไม่กำหนด จะแสดงผลเป็นกราฟเส้นโดยอัตโนมัติ
- minvalue and maxvalue หมายถึง ค่าต่ำสุด สูงสุด ของ indicator นี้
- XShift หมายถึง การเลื่อนแผนภูมิ

- ZOrder หมายถึง กำหนดตำแหน่งแกน Z ของการพล็อตที่กำหนด
- width หมายถึง ความหนาของ Chart

ตัวอย่าง

Plot(RSI(14), "My RSI", colorRed);



การกำหนดค่าของสี (Color Constants)

สีที่กำหนดเองนั้นหมายถึง สีที่ผู้ใช้เป็นผู้กำหนดเอง ซึ่งสามารถทำได้โดยไปที่ Tools->Preferences->Colors

colorCustom1 = 0
colorCustom2 = 1
colorCustom3 = 2
colorCustom4 = 3
colorCustom5 = 4
colorCustom6 = 5
colorCustom7 = 6
colorCustom8 = 7
colorCustom9 = 8
colorCustom10 = 9
colorCustom11 = 10
colorCustom12 = 11

colorCustom13 = 12

colorCustom14 = 13

colorCustom15 = 14

colorCustom16 = 15

colorBlack = 16

colorBrown = 17

colorDarkOliveGreen = 18

colorDarkGreen = 19

colorDarkTeal = 20

colorDarkBlue = 21

colorIndigo = 22

colorDarkGrey = 23

colorDarkRed = 24

colorOrange = 25

colorDarkYellow = 26

colorGreen = 27

colorTeal = 28

colorBlue = 29

colorBlueGrey = 30

colorGrey40 = 31

colorRed = 32

colorLightOrange = 33

colorLime = 34

colorSeaGreen = 35

colorAqua = 35

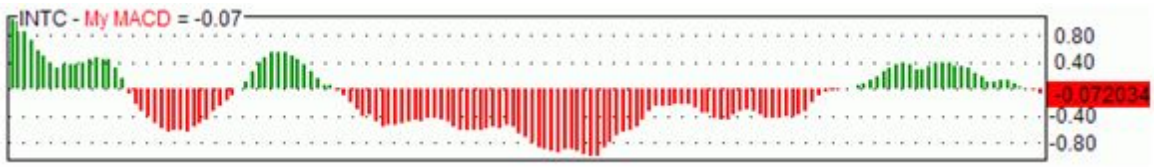
```
colorLightBlue = 37
colorViolet = 38
colorGrey50 = 39

colorPink = 40
colorGold = 41
colorYellow = 42
colorBrightGreen = 43
colorTurquoise = 44
colorSkyblue = 45
colorPlum = 46
colorLightGrey = 47

colorRose = 48
colorTan = 49
colorLightYellow = 50
colorPaleGreen = 51
colorPaleTurquoise = 52
colorPaleBlue = 53
colorLavender = 54
colorWhite = 55
```

ในตัวอย่างต่อไป จะทำการโค้ด MACD ที่พล็อตเป็นสีเขียวเมื่อมันอยู่เหนือศูนย์ และ มีสีแดงเมื่อมันอยู่ต่ำกว่าศูนย์

```
dynamic_color = IIf( MACD() > 0, colorGreen, colorRed );
 Plot( MACD(), "My MACD", dynamic_color  );
```

นอกจากการเปลี่ยนสีตามเงื่อนไขที่เรากำหนดได้แล้วนั้น เรายังสามารถเปลี่ยนลักษณะการพล็อตของ MACD ให้เป็น histogram แทนการพล็อตเป็นเส้นได้อีกด้วย

```
dynamic_color = IIf( MACD() > 0, colorGreen, colorRed );
Plot( MACD(), "My MACD", dynamic_color, styleHistogram | styleThick  );
```



หากคุณต้องการที่จะพล็อตกราฟแท่งเทียน คุณสามารถทำได้โดยใช้ฟังก์ชัน styleCandle ได้เลยดังตัวอย่าง

```
Plot( Close, "Price", colorBlack, styleCandle );
```

การพล็อตแบบดั้งเดิมที่มีการใช้สี (ราคาขึ้นบาร์สีเขียวและราคาลงบาร์สีแดง) ที่เราต้องทำก็เพียงแค่ระบุสี ตามเงื่อนไขที่คุณกำหนด

```
Plot(Close, "Price", IIf(Close > Open, ColorGreen, ColorRed), StyleBar|StyleThick);
```

การกำหนดรูปแบบของกราฟ (Style) (Style Constants)

Style สามารถที่จะกำหนดด้วยตัวแปรหลายตัวแปร เช่นคุณอาจจะอยากให้กราฟเส้นออกมาในรูปแบบของกราฟเส้นที่มีความหนา คุณก็จำเป็นที่จะต้องใช้คำสั่ง 2 คำสั่งประกอบเข้าด้วยกัน

styleLine = 1 - กราฟเส้นปกติ (เป็นค่า default) styleHistogram = 2 -กราฟ histogram
styleThick =4 - เพิ่มลักษณะให้มีความหนา
styleDots = 8 - กราฟจุด

styleNoLine = 16 - กราฟไม่มีเส้น

styleDashed = 32 - กราฟเส้นประ

styleCandle = 64 - กราฟแท่งเทียน

styleBar = 128 - กราฟบาร์

styleNoDraw = 256 - ไม่พล็อตกราฟ (แสดงแค่เฉพาะแกนของกราฟ)

styleStaircase = 512 - ลักษณะการพล็อตคล้ายๆขั้นบันได

styleSwingDots = 1024 - ใช้เป็นจุดแบ่งกลางสำหรับรูปแบบ staircase

styleNoRescale = 2048 - ไม่มีการเปลี่ยนสเกล

styleNoLabel = 4096 - ไม่มีป้ายแสดงค่าต่างๆของกราฟ

stylePointAndFigure = 8192 - พล็อตรูปแบบ point and figure

styleArea = 16384 - พล็อตกราฟให้เป็นพื้นที่ (เปรียบเทียบได้กับกราฟ histogram ที่มีความกว้างมากๆ)

styleOwnScale = 32768 - พล็อตกราฟด้วยสเกลของตัวมันเอง (เป็นอิสระต่อกันกับกราฟอื่นๆ)

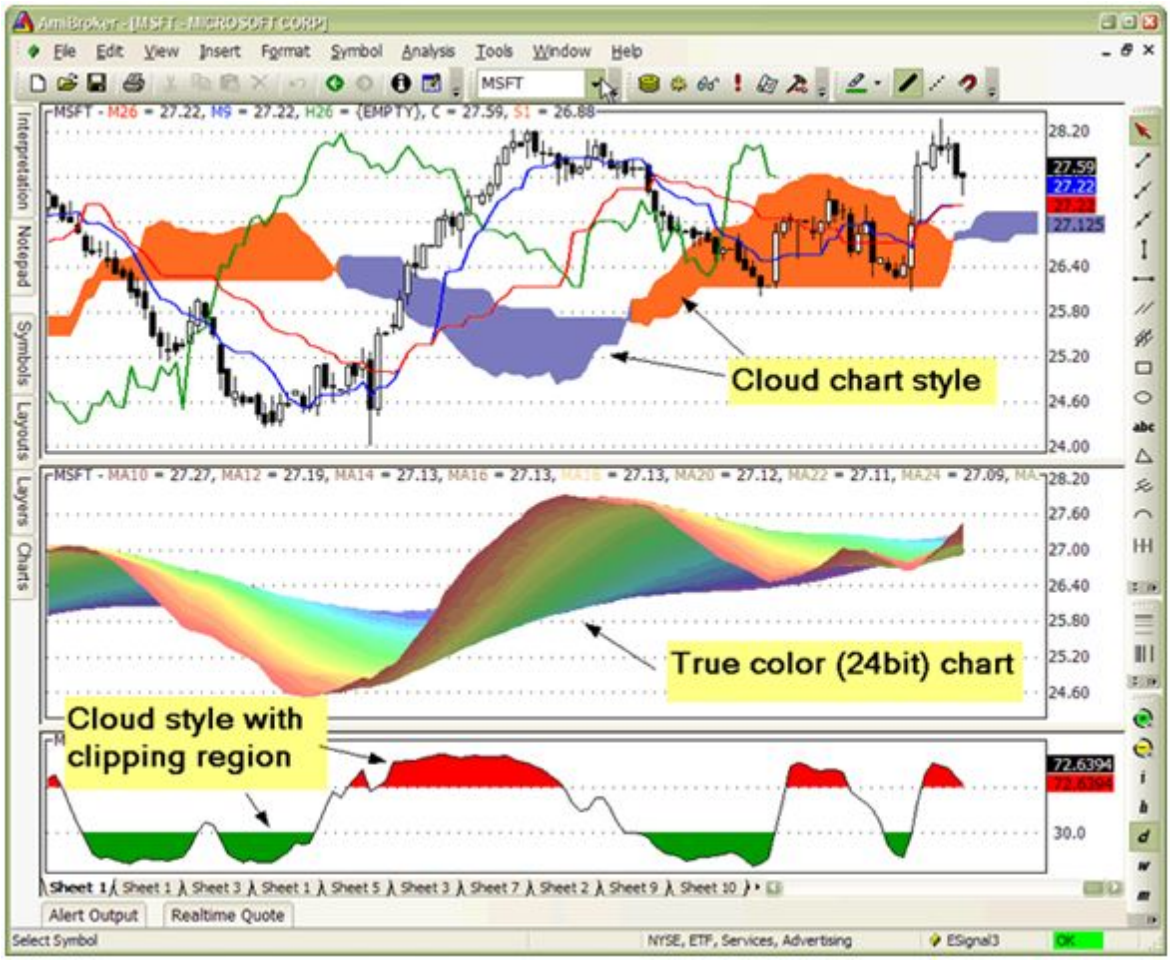
styleLeftAxisScale = 65536 - พล็อตโดยใช้สเกลของแกนด้านซ้าย (เป็นอิสระของแกนด้านขวา axis)

styleNoTitle = 131072 - ไม่แสดงค่าของกราฟนี้ด้วย หัวข้อที่เป็น string

styleCloud = 262144 - วาดกราฟลักษณะคล้ายเมฆ "cloud" (ดูได้จากตัวอย่างด้านล่าง)

styleClipMinMax = 524288 - พล็อตแยกพื้นที่ระหว่าง ระดับ Min และ ระดับ Max (หมายเหตุ: style รูปแบบนี้มักใช้กับการพล็อตส่วนใหญ่ไม่ได้)

styleGradient - (ใหม่ตั้งแต่เวอร์ชั่น 5.60) - การไล่ระดับของกราฟ สีของการไล่ระดับด้านบนถูกระบุไว้ในพารามิเตอร์ของสี ในฟังก์ชั่น Plot() การไล่ระดับของสีของส่วนล่างก็เช่นกัน สีของพื้นหลังสามารถกำหนดได้โดย SetGradientFill ฟังก์ชั่น styleGradient สามารถใช้ร่วมกับ styleLine ได้เป็นอย่างดี



สูตรของ pane ตรงกลางที่เป็นสีเรนโบว์ มีตัวอย่างโค้ดแสดงดังนี้

```
side = 1;
increment = Param("Increment",2, 1, 10, 1 );
for( i = 10; i < 80; i = i + increment )
{
up = MA( C, i );
down = MA( C, i + increment );

if( ParamToggle("3D effect?", "No|Yes", 1 ) )
```

```
side =  IIf(up<=down AND Ref( up<=down, 1 ), 1, 0.6 );
 PlotOHLC( up,up,down,down, "MA"+i, ColorHSB( 3*(i - 10), Param("Saturation", 128, 0,
255), side * Param("Brightness", 255, 0, 255 ) ), styleCloud | styleNoLabel);
}
```

สูตรของ pane ด้านล่างเป็นดังนี้ โดยใช้การประยุกต์ของฟังก์ชัน styleClipMinMax เข้ามาช่วยเพื่อแบ่งระดับของกราฟ ให้เปลี่ยนสีให้เป็นไปตามเงื่อนไขต้องการ

```
SetChartOptions(0,0,ChartGrid30 | ChartGrid70 );
r = StochK(14);
Plot( r, "StochK", colorBlack );
PlotOHLC( r,r,50,r, "", IIf( r > 50, colorRed, colorGreen ), styleCloud | styleClipMinMax,
30, 70);
```

X-shift feature

XShift เป็นพารามิเตอร์ที่สามารถที่จะทำให้คุณ Shift กราฟได้ สามารถทำได้โดยโดยระบุเป็นจำนวนบาร์ที่เราต้องการ Shift เรามาลองดูตัวอย่างการ Shift ที่ใช้กับ Moving average กัน

```
Periods = Param("Periods", 30, 2, 100 );
Displacement = Param("Displacement", 15, -50, 50 );

Plot( MA( C, Periods ), _DEFAULT_NAME(), ColorCycle, styleLine, 0, 0, Displacement );
```

PlotForeign() function

ด้วยฟังก์ชันนี้จะเป็นเรื่องง่ายที่จะสร้างกราฟของราคาของหุ้นหลายๆตัวพร้อมกันโดยใช้ฟังก์ชั่น PlotForeign:

PlotForeign(tickersymbol, name, color/barcolor, style = styleCandle | styleOwnScale, minvalue = {empty},maxvalue = {empty}, xshift = 0)

โดยใน argument แรกจะบอกถึงชื่อ ticker ที่เราต้องการนำมาพล็อต argument ที่ 2 คือ ชื่อที่เราต้องการจะตั้ง argument ที่ 3 คือสี argument ที่ 4 คือ style ที่เราต้องการจะพล้อต argument ที่ 5,6 คือค่า min max และ สุดท้ายคือ การ shift ตัวอย่างการ plot เช่น

```
PlotForeign( "^DJI", "Dow Jones", colorRe );
PlotForeign( "^NDX", "Nasdaq 100", colorBlue );
PlotForeign( "^IXIC", "Nasdaq Composite", colorGreen );
```

การพล็อตกราฟหลายๆอันด้วยสเกลที่ต่างกัน (Multiple plots using different scaling)

ใช้คำสั่ง styleOwnScale และ styleLeftAxisScale ทำให้สามารถพล้อต 2 กราฟหรือมากกว่านั้นได้

```
minimum = LastValue( Lowest( Volume ) );
maximum = LastValue( Highest( Volume ) );

Plot( Close, "Price", colorBlue, styleCandle );
 */ ตัวอย่างการพล็อตทั้งสองอันด้านล่างนี้จะใช้ฟังก์ชัน OwnScale
แต่ในส่วนของสเกลที่แสดงจะเป็นสเกลปกติเนื่องจากเราได้กำหนดค่า min และ max ของแกน Y
ไว้*/
Plot( Volume, "Volume", colorGreen, styleHistogram | styleThick | styleOwnScale,
minimum, maximum );
```

Plot(MA(Volume, 15), "MA volume", colorRed, styleLine |styleOwnScale, minimum, maximum);

ถ้าหากเปลี่ยนมาใช้ : styleLeftAxisScale = 65536 - จะสามารถที่จะพล็อตกราฟมากกว่าหนึ่งกราฟได้โดยใช้สเกลปกติ แต่จะต่างกันที่สเกลของแกนด้านขวา (right axis)

ตัวอย่าง : การพล็อต ราคา โวลุ่ม และ moving average :

```
// พล็อตราคา และ moving average ของตัวมันเอง
Plot( Close, "Price", colorWhite, styleCandle );
 Plot( MA( Close, 20 ), "MAC", colorRed );

// พล็อตโวลุ่ม และ moving average ของตัวมันเอง โดยใช้สเกลทางด้านซ้ายมือ
Plot( Volume , "Volume", colorBlue, styleLeftAxisScale | styleHistogram | styleThick );
Plot( MA( Volume,15), "MAV", colorLightBlue, styleLeftAxisScale );
```

การสร้างพารามิเตอร์ใหม่ๆขึ้นมาเพิ่ม เพื่อให้ง่ายเวลาพล็อต Ribbon ยกตัวอย่างเช่น:

```
Plot( Close, "Price", colorBlue, styleCandle );
Plot( 2, /* กำหนดความสูงของ ribbon ในหน่วยของ %ความกว้างของ pane*/ "Ribbon", IIf( up, colorGreen, IIf( down, colorRed, 0 )), /* เลือกสี */
styleOwnScale|styleArea|styleNoLabel, -0.5, 100 );
```

การใช้การกำหนดพารามิเตอร์ด้วยตนเอง (Using custom defined parameters)

Amibroker ช่วยให้ผู้ใช้ สามารถที่จะกำหนดการสร้างพารามิเตอร์เองได้ โดยพารามิเตอร์จะสามารถปรับใช้กับ indicator ได้อย่างสะดวกและรวดเร็ว

- Param("name", default, min, max, steps, incr = 0);
- ParamStr("name", "default");
- ParamColor("name", defaultcolor);
- ParamStyle("name", defaultval = styleLine, mask = maskDefault);

ฟังก์ชันเหล่านี้จะทำให้คุณสามารถกำหนดค่าพารามิเตอร์ใน indicator ของคุณได้เอง และ เมื่อคุณใช้ฟังก์ชั่น Param พารามิเตอร์ต่างๆที่คุณต้องการที่จะแก้ไขสามารถทำได้โดยเพียงแค่การคลิกขวาบน pane ของกราฟ หรือ กด Ctrl + R เท่านั้น ไม่จำเป็นต้องเข้ามาแก้ในโค้ด

ตัวอย่าง อย่างง่ายที่สุดในการใช้ ฟังก์ชันดังต่อไปนี้:

```
period = Param("RSI period", 12, 2, 50, 1 );
Plot( RSI( period ), "RSI( " + period + ") ", colorRed );
```

คลิกขวาที่ chart และ เลือก "Parameter" จากนั้นเลื่อนลงมาคุณจะเห็น RSI พล็อตด้วยระยะเวลาที่แตกต่างกัน (กำหนด Period ต่างกัน)

โค้ดตัวอย่างด้านล่างแสดงให้เห็นถึงวิธีการใช้ ParamStr เพื่อใช้กับ ticker และ ParamColor เพื่อใช้กับสีของกราฟ

```
ticker = ParamStr( "Ticker", "MSFT" ); sp = Param( "MA Period", 12, 2, 100 ); PlotForeign(
ticker, "Chart of "+ticker,
```

ParamColor("Price Color", colorBlack), styleCandle); Plot(MA(Foreign(ticker, "C"), sp), "MA", ParamColor("MA Color", colorRed));

ตัวอย่างง่ายๆของ formula โดยการพล็อตจุดสูงสุด ต่ำสุดของ กราฟ sine:

```
Cycle = Param("Cycle Months", 12, 1, 12, 1 )*22;//264==12 เดือน,22==1เดือน
xfactor = Param("Stretch",1,0.1,2,0.1);//1==1 ปี ,2==2 ปี
xshift = Param("slide",0,-22,22,2)/3.1416^2;//slide curve 1==5 วัน

x = 2*3.1416/Cycle/xfactor; y = sin(Cum(x)-xshift);

Plot(C,"Daily Chart", colorBlack, styleCandle | styleNoLabel);
Plot(y, "cycle =" + WriteVal(Cycle*xfactor/22,1.0)+"months",
colorBlue,styleLine|styleNoLabel|styleOwnScale);
```

คลิกขวาบน Chart และ เลือก "Parameters" จากนั้นลองเลือกเปลี่ยนค่าพารามิเตอร์บางอย่าง คุณจะเห็นการเปลี่ยนแปลงของกราฟได้ทันที

สำหรับใครอยากศึกษาเกี่ยวกับ Parameter เพิ่มเติม คุณสามารถที่จะศึกษาเพิ่มเติมได้ที่ Tutorial: Using drag-and-drop interface

การพล็อตข้อความตัวหนังสือลงบนกราฟ (Plotting texts at arbitrary positions on the chart)

PlotText("text", x, y, color, bkcolor = colorDefault) โดยที่
x - คือ x-coordinate ในหน่วยของบาร์ (เหมือนกับ LineArray)
y - คือ y-coordinate ในหน่วยของราคา

color คือ สีของตัวหนังสือ , bkcolor คือ สีของพื้นหลัง ถ้า bkcolor ไม่ได้ระบุไว้ (หรือให้ค่าเท่ากับ colorDefault) ตัวหนังสือที่แสดงจะถูกเขียนลงบนพื้นหลังที่เป็นพื้นหลังแบบ TRANSPARENT(พื้นหลังแบบไม่มีสี หรือโปร่งใส)

ตัวอย่างเช่น

```
Plot(C,"Price", colorBlack, styleLine );
Plot(MA(C,20),"MA20", colorRed );

Buy=Cross( C, MA(C,20 ) );
Sell= Cross( MA( C, 20 ), C ); dist = 1.5*ATR(10);
for( i = 0; i < BarCount; i++ )
{
	if( Buy[i] ) PlotText( "Buy\n@" + C[ i ], i, L[ i ]-dist[i], colorGreen );
	if( Sell[i] ) PlotText( "Sell\n@" + C[ i ], i, H[ i ]+dist[i], colorRed, colorYellow );
}

PlotShapes( Buy * shapeUpArrow + Sell * shapeDownArrow, IIf( Buy, colorGreen,
colorRed ) );
```

การใส่สีแบบไล่ระดับลงบนพื้นหลัง (Gradient fill of the background)

ฟังก์ชั่นนี้จะช่วยให้คุณสามารถเติมสีพื้นหลังโดยการไล่ระดับสีของ indicator ได้ ขอให้คุณทราบไว้ว่าการไล่ระดับพื้นหลังด้วยวิธีนี้จะมีความเป็นอิสระต่อ (ไม่มีความเกี่ยวข้องต่อกัน) กันจากสีพื้นหลังของ Chart ที่มีอยู่เดิมแล้ว พารามิเตอร์มีดังนี้ :

topcolor - ระบุสีด้านบนของการไล่ระดับสี
fill bottomcolor - ระบุสีด้านล่างของการไล่ระดับสี

titlebkcolor - (เสริม) สีพื้นหลังของข้อความที่แสดงผล ถ้าไม่ได้ระบุไว้ สีที่กำหนดไว้ด้านบนจะถูกนำมาใช้โดยอัตโนมัติ

ตัวอย่าง

SetChartBkGradientFill(ParamColor("BgTop",colorWhite),ParamColor("BgBottom", colorLightYellow));

การไล่ระดับสีบนพื้นที่ของกราฟ (Gradient fill area charts)

การไล่ระดับสีที่เรียบง่าย สามารถแสดงผลได้โดยใช้ :

Plot(C, "C", colorDefault, styleGradient | styleLine);

ในกรณีที่ต้องการควบคุมรายละเอียดของการไล่ระดับของสีมากกว่าปกติ คุณควรที่จะเรียกใช้ ฟังก์ชั่นนี้ก่อนเป็นอันดับแรก SetGradientFill(topcolor, bottomcolor, baseline, baselinecolor)ก่อนที่คุณจะไปใช้ฟังก์ชั่นพล็อต Plot()

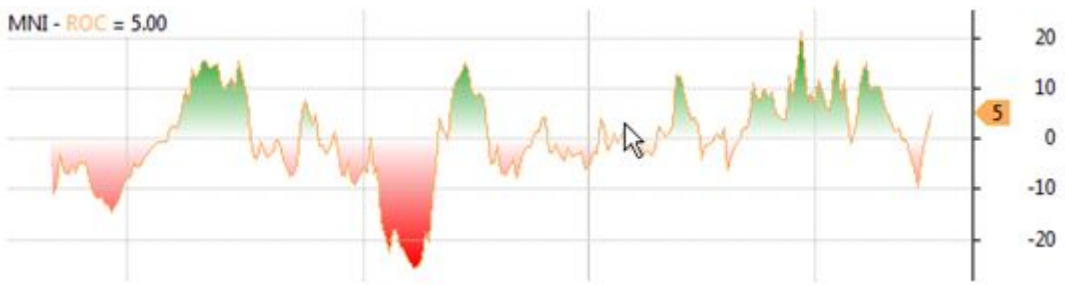
ส่วนพารามิเตอร์ที่ใช้เป็นตัวเลือกของคุณคือ baseline/baselinecolor นั้นสามารถให้สีของกราฟมีการไล่ระดับที่ย้อนกลับได้(เช่นกราฟของ underwater equity) และประกอบด้วย 3 สีด้วยกันคือในส่วนของ top->baseline->bottom นอกจากนี้ คุณสามารถดูโค้ดของกราฟ Underwater Equity เพื่อใช้เป็นกรณีศึกษาการไล่ระดับของสีแบบย้อนกลับได้(โดยที่มี baseline อยู่บนจุดสูงสุด) พารามิเตอร์ Baseline นั้นจะเป็นการระบุค่า ค่าหนึ่งบนแกน Y ในส่วนของ พารามิเตอร์ baselinecolor นั้นจะเป็นการระบุสีของการไล่ระดับ ที่ใช้ในแต่ล่ะส่วน ถ้าคุณไม่ได้ระบุสีของ baselinecolor จะมีการไล่ระดับของสีแค่เพียง 2 ส่วนเท่านั้นคือ การไล่ระดับจาก topcolor->bottomcolor

ยกตัวอย่างเช่นหน้าจอแสดงผลของการไล่ระดับของ three-color gradient Rate Of Change จะใช้สีเขียว ที่จุด "top" สำหรับค่าที่เป็นบวก สีของพื้นหลังเหมทอนกับสีของ"baseline" และ สีแดงเป็นจุด "bottom" สำหรับค่าที่เป็นลบ การระบุเงื่อนไขข้างต้นสามารถเขียนโค้ดออกมาได้ดังตัวอย่างด้านล่าง:

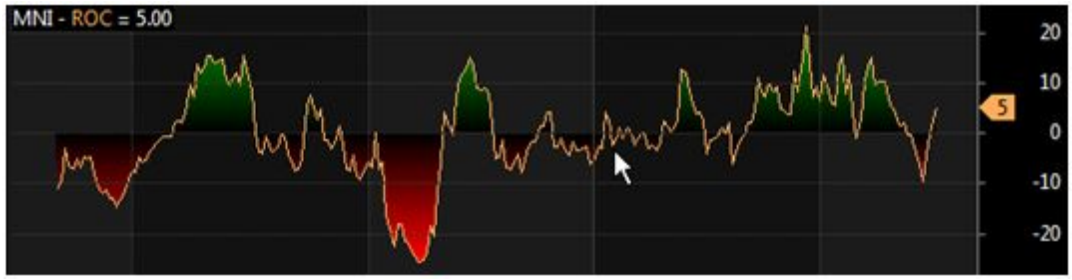
การสีไล่ระดับของสีด้านบน จะถูกระบุโดยอาร์กิวเมนต์ topcolor และ การสีไล่ระดับของสีด้านล่าง จะถูกระบุโดยอาร์กิวเมนต์ botttomcolor พารามิเตอร์ที่ไม่จำเป็นต้องใส่ไว้ก็ได้ก็คือ (baselinecolor) สำหรับการใช้งานตัวอย่างของการไล่ระดับสี Chart ย้อนกลับ (มีพื้นฐานที่ด้านบน) พารามิเตอร์พื้นฐานระบุตำแหน่งแกน Y ของพื้นฐาน Chart พารามิเตอร์ baselinecolor ระบุสีของการไล่ระดับสีที่จะนำมาใช้ในระดับที่ว่า หาก baselinecolor ไม่ได้ระบุแล้วเท่านั้นการไล่ระดับสี 2 สีคือพล็อต (topcolor -> bottomcolor)

ยกตัวอย่างเช่นในการแสดงสามสีไล่ระดับสีอัตราการเปลี่ยนแปลงที่จะใช้สีเขียวเป็น "ด้านบน" สีสำหรับค่าบวกสีพื้นหลังเป็น "พื้นฐาน" สีและสีแดงเป็น "ด้านล่าง" สีค่าลบก็พอที่จะเขียน

```
SetGradientFill( colorGreen /*ด้านบน*/, colorRed /*ด้านล่าง*/, 0 /*ระดับของ baseline*/,
GetChartBkColor() /*สีของ baseline*/);
Plot( ROC( C, 14), "ROC", colorLightOrange, styleLine | styleGradient, Null, Null, 0, -1);
```



หรือ ถ้าในกรณีที่ใช้ธีมสีดำ



การทำให้กราฟมีความหนา (Super thick charts)

Plot(C, "Close", colorDefault, styleBar, Null, Null, 0, 1, -20 /* เส้นของกราฟจะมีความหนาในหน่วยของ%ของบาร์*/);

ตอนนี้คุณสามารถรับสายหนาสุดที่แสดงในตัวอย่างด้านล่างนี้ (เส้นกราฟที่มีความหนา10 พิกเซล):

Plot(C, "Close", colorRed, styleLine, Null, Null, 0, 1, 10 /*หนา 10 พิกเซล*/);

จิปาถะ (Miscellaneous)

Variable	Usage	Applies to
Title	กำหนดชื่อหัวข้อด้วยตัวหนังสือ (text) หากคุณใช้ตัวแปร title คุณสามารถระบุสีต่างๆใน string นั้นๆได้ เราแนะนำให้คุณใช้ฟังก์ชั่น AFL EncodeColor เพราะมันเป็นวิธีการที่ง่ายกว่าการใช้ การโค้ดตามปกติ EncodeColor (ColorNumber) และ คุณสามารถเขียนตัวอย่างข้างต้นได้ดังนี้ Title = "This is written in " + EncodeColor(colorViolet) + "violet color " + EncodeColor(	Indicators

	colorGreen) + "and this in green"; สามารถใส่คำบรรยายใต้เส้นหลายๆอัน ได้โดยเพียงใช้ \ n ในการแบ่งคำบรรยาย ตัวอย่างเช่น: Title = "This is 1st line\nThis is second line"; เพื่อความครบถ้วนสมบูรณ์: สีนอกจากนี้ยังสามารถระบุโดยใช้ลำดับ Espace แต่ก็ไม่แนะนำเพราะ เป็นเรื่องยากที่จะเขียนและยากที่จะอ่าน \\ ลำดับ cXX XX ซึ่งเป็น 2 หลักหมายเลขดัชนีการระบุสี \\ C38 - กำหนดสีม่วงมีลำดับพิเศษ \\ C-1 ที่จะเริ่มต้นรีเซ็ตสีแกน ตัวอย่างเช่น เพื่อความครบถ้วนสมบูรณ์ : การกำหนดสีต่างๆสามารถทำได้โดย การใช้วิธีการโค้ดแบบปกติ(การเรียงลำดับ) แต่เราไม่แนะนำให้ทำอย่างนั้น เพราะมันยากที่จะเขียน และ ยากที่จะอ่าน \\cXX เป็นการเขียนเรียงลำดับ เมื่อ XX คือเลข 2 หลักที่ใช้ระบุสีของดัชนี\\c38 - กำหนดให้เป็นสี violet \\c-1 คือการ resets ไปสู่ค่า default ของสีของแกน ยกตัวอย่างเช่น Title = "This is written in \\c38violet color \\c27and this in green";	
Tooltip	ล่าสมัยในเวอร์ชั่น 5.40. การใช้หน้าต่างของข้อมูล แทนที่จะใช้ฟังก์ชัน Plot() ด้วย styleHidden ถ้าคุณต้องการที่จะเพิ่มค่าที่คุณกำหนดด้วยตัวเองไปยัง data tooltip สามารถทำได้ดังตัวอย่างดังต่อไปนี้	Indicators

	Plot(my_value, "MyValueForTooltip", colorBlack, styleHidden);	
GraphXSpace	ว่าหนดว่าควรจะมีพื้นที่ว่างเท่าไหร่อยู่เหนือ/ใต้กราฟ(ในหน่วย %) ตัวอย่างเช่น: GraphXSpace = 5; เพิ่มพื้นที่ว่าง 5% ทั้งบนและล่างของกราฟ เมื่อเราไม่ได้กำหนดค่าของ GraphXSpace ค่า default ของมันจะเท่ากับ 2%	Indicators

GraphLabelDecimals	(ใหม่ในเวอร์ชั่น 5.90) เป็นการควบคุมตำแหน่งของทศนิยมในกราฟ (เช่น หากคุณใช้ GraphLabelDecimals = 2; ค่าต่างๆที่แสดงบนหน้ากราฟจะมีค่าทศนิยม 2 ตำแหน่ง)	Indicators
GraphZOrder	ตัวแปร GraphZOrder จำช่วยให้คุณเปลี่ยน ลำดับของการพล็อตลงบนกราฟได้ เมื่อ GraphZOrder ไม่ได้ถูกกำหนดค่าไว้ หรือมีค่าเป็นศูนย์(false) - คำสั่งลำดับเก่าที่สุดจะถูกใช้ก่อน (last to first)และเมื่อ GraphZOrder เท่ากับ 1 (true) - จะเป็นการเรียงลำดับในทางตรงกันข้าม	Indicators

คุณจะสร้างตัว Exploration ด้วยตัวเองได้อย่างไร

(How to create your own exploration)

เครื่องมือ Exploration นั้น ถือได้ว่าเป็นอีกหนึ่งเครื่องมือ ที่มีประโยชน์มากๆใน Amibroker มันมีลักษณะคล้ายๆกับการ Scan แต่ต่างกันตรงที่ ผลลัพธ์ของกระบวนการจะไม่ได้โชว์แต่สัญญาณการซื้อขายเพียงเท่านั้น แต่ยังสามารถนำไปใช้ Screen ข้อมูลตามที่ผู้ใช้ต้องการได้อีกด้วย ซึ่งให้มันได้ผลลัพธ์ของข้อมูลมากกว่าการ Scan แบบปกติ

การทำงานของ Exploration นั่นง่ายมาก เพียงแค่เราใส่คำสั่ง filter ซึ่งเป็นตัวคัดกรองว่าข้อมูลไหนตรงตามเกณฑ์ที่เราต้องการ หากข้อมูลนั้นตามเกณฑ์ของเรา มันจะถูก Assign เลข 1 และ แสดงบนรีพอร์ทของเรา

ตัวอย่างเช่น ถ้าหากเราจะต้องการค้นหาเครื่องหมาย (หุ้น) ที่มีราคาปิดมากกว่า 50 บาท ก็ให้เขียนโค้ดดังนี้

```
filter = close > 50;
```

(หมายเหตุ: หากจะเขียนโค้ดใหม่ให้เปิด Formula Editor โดยการไปที่ Analysis->Formula Editor menu พิมพ์โค้ดลงไป และ เลือก Tools->Send to Analysis menu ใน Formula editor)

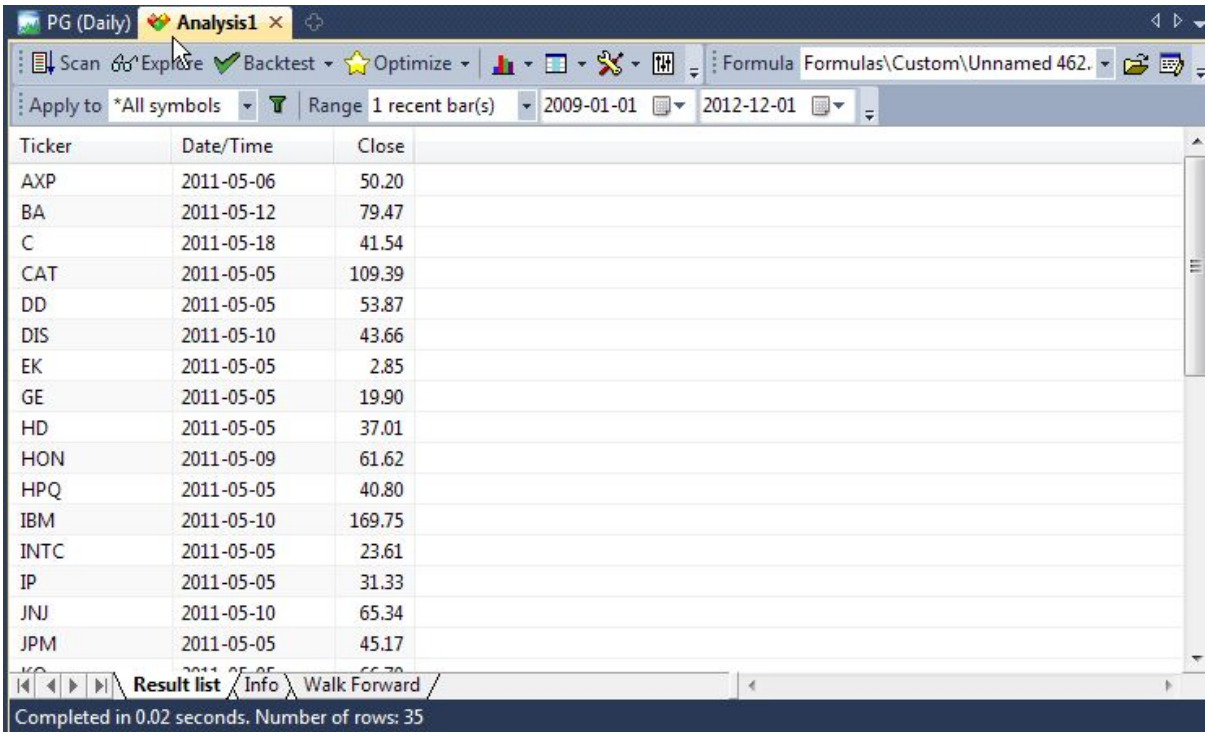
การ Explore นั้น หากผู้ใช้เลือก "All quotations" ข้อมูลทั้งหมดที่ตรงตามเกณฑ์ของผู้ใช้ก็จะขึ้นมาในรีพอร์ท หากคุณต้องการให้มันสแกนเฉพาะเพียง bar ล่าสุดก็ให้เลือกใช้ "1 recent bar(s)"

ผู้ใช้ยังสามารถสร้าง Column ใน Exploration เพื่อนำออกไปยังที่อื่นๆได้ (จากผู้แปล : เช่นหากคุณจะนำไปวิเคราะห์ต่อใน excel) โดยการใช้คำสั่ง AddColumn (พิมพ์ลงในโค้ดของการ Explore)

AddColumn(Close, "Close");

ข้อมูลแรกในคำสั่ง AddColumn คือ “array” ที่ผู้ใช้ต้องการ ส่วนข้อมูลตัวที่สองคือ “ชื่อ” ของช่อง column ที่ผู้ใช้ต้องการจะตั้ง

ถ้าหากคุณกดปุ่ม Explore คุณจะพบกับรูปแบบรีพอร์ทดังนี้



Ticker	Date/Time	Close
AXP	2011-05-06	50.20
BA	2011-05-12	79.47
C	2011-05-18	41.54
CAT	2011-05-05	109.39
DD	2011-05-05	53.87
DIS	2011-05-10	43.66
EK	2011-05-05	2.85
GE	2011-05-05	19.90
HD	2011-05-05	37.01
HON	2011-05-09	61.62
HPQ	2011-05-05	40.80
IBM	2011-05-10	169.75
INTC	2011-05-05	23.61
IP	2011-05-05	31.33
JNJ	2011-05-10	65.34
JPM	2011-05-05	45.17

สังเกตได้ว่าจะมีอยู่ 3 คอลัมน์ ที่ชื่อ Ticker, Date/Time และ Close (array ที่ผู้ใช้ต้องการจะ explore)

หากคุณกดปุ่ม Export ไฟล์จะถูกเปลี่ยนเป็นไฟล์สกุล CSV(comma separated values) ซึ่งสามารถนำไปเปิดบน Excel ได้

คำสั่ง AddColumn ยังสามารถกำหนดรูปแบบคอลัมน์ที่หลากหลายได้มากขึ้น กว่าตัวอย่างข้างต้นได้อีก ฟังก์ชั่นเต็มๆของมันประกอบด้วยข้อมูลดังนี้ (ข้อมูลที่คุณจำเป็นต้องกรอกนั้นมีเพียงแค่ 2 ข้อมูลแรกเท่านั้น)

AddColumn(array, name, format = 1.2, textColor = colorDefault, bkgndColor = colorDefault)

format - จะกำหนดตัวเลขทศนิยมลงท้าย โดย default แล้ว ตัวเลขจะมีทศนิยม 2 ตำแหน่ง (1.2) ถ้าหากเปลี่ยนเป็น 1.5 คือ ทศนิยม 5 ตำแหน่ง 1.0 ไม่มีทศนิยม ดังนั้น หากคุณเขียนโค้ด

AddColumn(Close, "Close", 1.4);

จะให้ราคาปิดทศนิยม 4 ตำแหน่ง

นอกจากนี้ยังมีรูปแบบ Format อีกมากมาย

formatDateTime – ฟอร์แมตรูปแบบวันและเวลาตามที่ผู้ใช้ต้องการ

AddColumn(DateTime(), "Date / Time", formatDateTime);

formatChart - สามารถแปรผล ASCII character codes :

ตัวอย่างเช่น:

```
Buy=Cross(MACD(),Signal());
Sell=Cross(Signal(), MACD());

Filter=Buy OR Sell;
SetOption("NoDefaultColumns", True );
AddColumn( DateTime(), "Date", formatDateTime );
AddColumn( IIf( Buy, 66, 83 ), "Signal", formatChar );
```

textColor และ bkgndColor จะปรับเปลี่ยนสีบนรีพอร์ทของผู้ใช้

ตัวอย่างเช่น หากผู้ใช้ต้องการจะให้ตัวเลขราคาปิดเป็นสีเขียวเมื่อ rate of change เทียบกับ 1 วันที่แล้วมากกว่าศูนย์ ถ้าหากน้อยกว่าศูนย์ให้เปลี่ยนเป็นสีแดง ให้เขียนโค้ดดังนี้

```
AddColumn( Close, "Close", 1.4, IIF( ROC(C, 1 ) > 0, colorGreen, colorRed ) );
```

ตัวอย่าง

ผู้ใช้สามารถ export database ไปยัง CSV file โดยเขียนโค้ดดังนี้

```
filter = 1; /* ใช้คำสั่งนี้กับทุกๆสัญลักษณ์ */

AddColumn(Open,"Open",1.4);
AddColumn(High,"High",1.4);
AddColumn(Low,"Low",1.4);
AddColumn(Close,"Close",1.4);
AddColumn(Volume,"Volume",1.0);
```

ตัวอย่างนี้จะได้ผลลัพธ์ออกมาเป็น เครื่องหมายที่มีปริมาณการเทรดจำนวนมาก

```
filter = volume > 5000000; /* ปรับแต่งปริมาณ Volume ตามต้องการ
เพื่อกรองหลักทรัพย์ที่มีสภาพคล่องน้อยๆออกไป */

AddColumn(Close,"Close",1.4);
AddColumn(Volume,"Volume",1.0);
```

หรือ ให้เครื่องหมายที่มี volume มากกว่า 30% ของ 40-day exponential average

```
filter = volume > 1.3 * ema( volume, 40 );
AddColumn(Close,"Close",1.4);
```

```
AddColumn(Volume,"Volume",1.0);
```

คุณสามารถ Export มันเพื่อนำไปใช้วิเคราะห์เพิ่มเติมตามต้องการได้

```
filter = close > ma( close, 20 ); /* คัดแต่หุ้นที่มีราคาปิดสูงกว่า MA 20 วัน*/

AddColumn( macd(), "MACD", 1.4 );
AddColumn( signal(), "Signal", 1.4 );
AddColumn( adx(), "ADX", 1.4 );
AddColumn( rsi(), "RSI", 1.4 );
AddColumn( roc( close, 15 ), "ROC(15)", 1.4 );
AddColumn( mfi(), "MFI", 1.4 );
 AddColumn( obv(), "OBV", 1.4 );
AddColumn( cci(), "CCI", 1.4 ); AddColumn( ultimate(), "Ultimate", 1.4 );
```

อีกตัวอย่างหนึ่งที่ใช้ลูกเล่นกับสีของผลลัพธ์

```
Filter =1;

AddColumn( Close, "Close", 1.2 );
AddColumn( MACD(), "MACD", 1.4 , IIf( MACD() > 0, colorGreen, colorRed ) );
AddTextColumn( FullName(), "Full name", 77 , colorDefault, IIf( Close
< 10, colorLightBlue, colorDefault ) );
```

กราฟ Scatter (X-Y) พล็อต ใน Exploration (Scatter (X-Y) charts in Exploration)

เวอร์ชั่น 5.60 มี Feature ใหม่คือ scatter X/Y charts เพื่อนำไปใช้หาความสัมพันธ์ของ หลักทรัพย์แต่ละตัวในเชิง Correlation, Risk และอื่นๆ ซึ่งเป็น feature ที่พัฒนามาจาก "Risk/yield" map

ผู้ใช้สามารถสร้าง scatter X/Y charts โดยการใส่ XYChartAddPoint สำหรับค่า X,Y ที่ผู้ใช้ต้องการที่จะใส่บนกราฟ

ยกตัวอย่างเช่น คุณต้องการจะหาความสัมพันธ์ของ MFE/Profit และ MAE/Profit ก็สามารถใช้ XYChartAddPoint AFL function เพื่อใช้ในการช่วยหาความสัมพันธ์ดังกล่าวได้

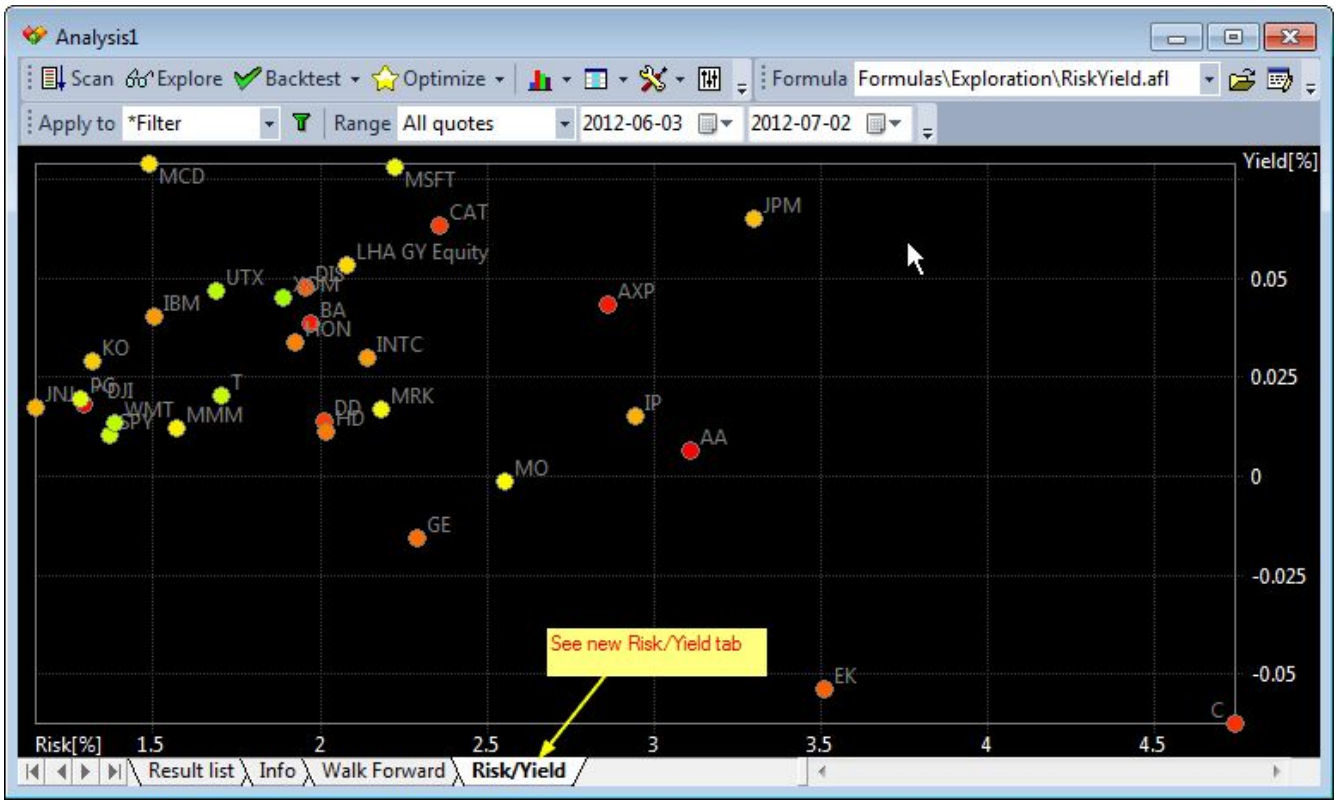
หากต้องการจะสร้าง risk/yield scatter chart จาก Feature นี้ ให้คุณทำตามขั้นตอนดังต่อไปนี้

1. File -> New -> Analysis

2. เลือก "Formulas\Exploration\RiskYield.afl" file (มุมขวาตามในภาพด้านล่าง)

3. กด Explore บนหน้าต่าง Analysis window

4. ผู้ใช้จะเห็น "Risk/Yield" tab อยู่ด้านล่าง ให้คลิกที่ tab นั้น โปรแกรมจะแสดงผล ดังภาพดังด้านล่างนี้




```
// XY scatter chart
// นี่คือแผนภาพ Risk-Yield map
// หมายเหตุ เลือกใช้ข้อมูลรายสัปดาห์
// คำนวณหาผลตอบแทนเฉลี่ยรายสัปดาห์ (yield)
// และ SD ของผลตอบแทนนั้นๆ(risk)

Filter=Status("lastbarinrange");
Length = SelectedValue( BarIndex());
 Chg = ROC( C, 1 ); // yieldของ1bar//
yield = MA( Chg, Length - 1);
risk =   StDev( Chg, Length - 1);
AddColumn(yield,"yield");
AddColumn(risk,"risk");
```

```
Clr = ColorHSB( 2 * Status("stocknum") % 255, 255, 255 );

XYChartAddPoint( "Risk/Yield", Name(), risk[ Length ], yield[ Length ] , Clr );
XYChartSetAxis("Risk/Yield", "Risk[%]", "Yield[%]");
```

เคล็ดลับส่งท้าย (Final tip)

คุณสามารถจะเรียงลำดับข้อมูลบนรีพอร์ทของคุณได้ เพียงแค่คลิกที่ Header ของแต่ละคอลัมน์ มันจำทำการเรียงข้อมูลตามลำดับ

วิธีการเขียนคอมเม้นท์บนชาร์ท

(How to write your own chart commentary)

หนึ่งในประโยชน์ของการใช้ภาษา AFL นั้น คือการเขียนคอมเม้นท์บนชาร์ทของผู้ใช้ โดยมีหลักการดังนี้

1. แต่ละคอมเม้นท์จะประกอบไปด้วย 2 ส่วน ส่วนที่หนึ่งคือ ตัวอักษรธรรมดา (Static Text) และส่วนที่สองคือ ภาษา AFL
2. AmiBroker จะโชว์คอมเม้นท์ของหลักทรัพย์ที่ผู้ใช้เลือกอยู่ และข้อความจะเปลี่ยนไปตามวันเวลาที่ผู้ใช้เลือก (Dynamic Content)
3. ข้อความตัวอักษรธรรมดา และ ส่วนที่เป็นภาษา AFL จะออกมาในหน้าต่างคอมเม้นท์
4. ผู้ใช้สามารถพลอตสัญญาณซื้อขายเข้าไปในหน้าต่างได้อีกเช่นกัน

Commentaries สามารถเข้าถึงได้โดยไปที่ Analysis->Commentary เมื่อคุณเปิดหน้าต่าง Commentary ออกมาคุณจะเห็นแท็บ 2 แท็บ Commentary และ Formula บนแท็บ Formula

ผู้ใช้สามารถพิมพ์ภาษา AFL เพื่อให้ Amibroker ประมวลผลออกมา ส่วนในหน้าต่างแท็บ Commenta ในส่วนถัดๆไปจะอธิบายขั้นตอนการเขียนคอมเม้นท์เพิ่มเติม

การเขียนอักษรธรรมดา (Writing static texts)

ทุกๆครั้งที่ผู้ใช้จะเขียนอักษรธรรมดา จะต้องมีเครื่องหมายคำพูดคลุมประโยคนั้นไว้ และลงท้ายด้วย ; เสมอ เช่น

"This is sample static text statement";

คุณสามารถเขียนหลายๆประโยคได้ และ แต่ละประโยคจะขึ้นเป็นบรรทัดใหม่บนหน้าต่าง

"This is first line of text"; "This is second line of text";

ผู้ใช้ลองพิมพ์ประโยคด้านบนลงบนหน้าต่าง Formula และ เปิดหน้าต่าง Commentary จะพบกับข้อความที่ไม่มี เครื่องหมายคำพูดหรือ ; ประกบไว้อยู่เนื่องจาก Amibroker ได้แปลงข้อความในเครื่องหมายคำพูดเป็น String เอาไว้และโชว์แต่ String บนหน้าต่าง Commentary

คำแนะนำสำหรับผู้ใช้ คือ ควรจะพิมพ์ ฟังก์ชัน printf ก่อนจะพิมพ์ข้อความทุกๆครั้ง

printf("This is sample static text statement");

ถ้าหากต้องการจะเขียนข้อความหลายๆบรรทัดผู้ใช้สามารถใช้ break sequence ('\n') เพื่อเป็นตัวเปลี่ยนบรรทัดแทนได้ :

printf("This is first line of text\nThis is second line of text\nThis is third line of text");

คุณสามารถแบ่งข้อความออกเป็นส่วนๆได้ โดยการเขียนโค้ด ดังนี้ :

printf("This" + " is" + “single"+ ”line" + "of text");

สีและไสตล์ (Colors and styles)

ในเวอร์ชั่น 5.9 ขึ้นไป จะรองรับการเปลี่ยนสี และ ไสตล์ของข้อความ หากคุณจะต้องการทำให้ข้อความ มีความหนา ให้คุณขึ้นต้นด้วย และ ลงท้ายด้วย เช่นเดียวกันกับตัวเอียง และ หากจะเปลี่ยนสีให้ใช้คำสั่ง EncodeColor ดังตัวอย่างด้านล่าง

```
printf("<b>Bold text</b>\n");
printf("<i>Italic text</i>\n");
printf("Now " + EncodeColor( colorRed ) + "red text\n");
printf("and finally " + EncodeColor( colorGreen ) + "green <b>AND bold <i>AND
italic</i></b>\n");
printf(EncodeColor( colorBlack ) + "going back to black");
```

ข้อความที่แปรไปตามการประมวลผล (Dynamic content)

ข้อความที่กล่าวไปเมื่อสักครู่นี้เป็นเพียงข้อความตัวอักษรธรรมดา หากผู้ใช้ต้องการจะใส่ข้อความที่ผ่านการประมวลผล ผู้ใช้ต้องพิมพ์ Function เข้าไป Function 2 ตัวที่สำคัญที่สุดคือ NumToStr() และ WriteIF() ฟังก์ชัน WriteIF() จะถูกอธิบายในส่วนถัดไป เรามาเริ่มจากฟังก์ชัน NumToStr()

คู่มือ Amibroker กล่าวไว้ว่า:

SYNTAX	NumToStr(NUMBER); NumToStr(ARRAY);
RETURNS	STRING
FUNCTION	ฟังก์ชันนี้ใช้ได้แค่บน guru commentary เท่านั้น ไว้แสดงค่าที่เป็นตัวเลขของ NUMBER หรือ ARRAY

ยกตัวอย่างเช่น:

```
printf( NumToStr( close ) );
```

เมื่อคุณเปิดหน้าต่าง Commentary คุณจะพบกับราคาปิดซึ่งเป็นค่าเดียวกันกับ Bar ที่ผู้ใช้เลือกเอาไว้ ในกรณีที่ผู้ใช้เลือก Bar อื่น และ กด Refresh ราคาปิดจะเปลี่ยนไปตามราคาของวันที่คุณเลือกเอาไว้ ดังนั้น NumToStr(close) จะโชว์ราคาปิด และ ข้อมูลอื่นๆของ Bar ที่ผู้ใช้เลือก

```
printf( NumToStr( macd() ) );
```

ผู้ใช้จะเห็นค่าของ MACD ของ Bar ที่ผู้ใช้เลือกเอาไว้ เราสามารถเขียนสูตรได้ดังนี้

```
printf( "Closing price = " + NumToStr( close ) + "\n" );
printf( "Change since yesterday = " + NumToStr( close - ref( close,
-1 ) ) + "\n" );
printf( "Percent chg. since yesterday = " + NumToStr( roc( close, 1 )
) + " %%\n" );
printf( "MACD =" + NumToStr( macd() ) + " , Signal line =" +
NumToStr( signal() ) + "\n" );
```

เมื่อเปิดหน้าต่าง Commentary ผู้ใช้จะพบกับ

:

```
Closing price = 17.940
Change since yesterday = -0.180
Percent chg. since yesterday = -0.993 %
MACD = -0.001 , Signal line = 0.063
```

ผู้ใช้สามารถพิมพ์ name() date() เพื่อให้หน้าต่าง Commentary โชว์ชื่อหลักทรัพย์ และ วันที่ผู้ใช้เลือกเอาไว้ :

printf("Statistics of " + name() + " as of " + date());

ผู้ใช้สามารถใช้เครื่องหมาย printf flexible % format แทน NumToStr ในการเปลี่ยนตัวเลขเป็น String ตัวอย่างเช่น การใช้ %.2f หมายความว่าใช้แสดงตัวเลขออกมาพร้อมกับทศนิยม 2 ตำแหน่ง %.3f แปลว่าตัวเลขพร้อมทศนิยม 3 ตำแหน่ง %g แปลว่าให้แสดงตัวเลขตามทศนิยมที่เกิดขึ้นจริงๆ

ตัวอย่างเช่น

```
printf( "Closing price = %.3f\n", close );
printf( "Change since yesterday = %.3f\n", close - ref( close, -1 ) );
 printf( "Percent chg. since yesterday = %.2f%%\n", roc( close, 1 ) );
printf( "MACD = %.4f, Signal line = %.4f\n", macd(), signal() );
```

คุณจะสังเกตได้ว่าตัวโค้ด สั้น และ ดูง่ายขึ้น ส่วนแรกของฟังก์ชัน printf คือ String ที่แสดงข้อความและตัวเลข ส่วนที่สอง คือ ค่าที่ผู้ใช้อยากให้ Commentary แสดงออกมา มีอีกข้อสังเกตหนึ่งคือ หากผู้ใช้ต้องการจะใส่หน่วยเป็นเปอร์เซ็นต์จะต้องใช้เครื่องหมาย %% ซ้ำกัน 2 ครั้ง

ในกรณีที่ผู้ใช้ต้องการให้ Commentary แสดงข้อความเมื่อผ่านเกณฑ์ใดเกณฑ์หนึ่งนั้น จะต้องใช้คำสั่งที่จะอธิบายในลำดับต่อไป

เงื่อนไขต่างๆของผลลัพธ์ที่มีลักษณะเป็นตัวหนังสือ (Conditional text output)

คำสั่ง WriteIF() จะแสดงผลตามเกณฑ์ที่ผู้ใช้ตั้งเอาไว้

SYNTAX	writeif(EXPRESSION, "TRUE TEXT", "FALSE TEXT")
RETURNS	STRING
FUNCTION	ฟังก์ชันนี้ใช้ได้แค่บน guru commentary เท่านั้น ถ้าหาก EXPRESSION เป็นจริง ค่าของ TRUE TEXT จะแสดงขึ้นมาบน

commentary แต่ถ้า EXPRESSION ไม่เป็นจริง ค่าของ
FALSE TEXT จะแสดงขึ้นมาบน commentary

ผู้ใช้สามารถให้หน้าต่าง Commentary แสดงผลตามที่ต้องการ

เช่น

```
writeif( macd() > signal(), "The MACD is bullish because is is above it's signal line",
"The MACD is bearish because it is below its signal line" );
```

ผู้ใช้สามารถสร้างเกณฑ์ต่างๆมากขึ้น ดังนี้

```
"The current market condition for "+ name() + " is: ";

avgcond1 = ( c > ema( close, 200) ) + 0.1 * ( close > ema( close, 90)) + 0.1 * ( close >
ema( close , 30 ) );

avgcond2 = -( c < ema( close, 200) ) - 0.1 * ( close < ema( close, 90) ) - 0.1 * ( close <
ema( close , 30 ) );

WriteIf( avgcond1 == 1.2, "Very Bullish",
        WriteIf( avgcond11 == 1, "Bullish",
                WriteIf( avgcond1 == 1.0, "Mildly Bullish", "") ) ) +
WriteIf( avgcond2  == -1.2, "Very Bearish",
        WriteIf( avgcond2 == -1.1, "Bearish",
                WriteIf( avgcond2 == -1.0, "Mildly Bearish", "") ) );
```

โค้ดด้านบนจะแสดงผล "The current market condition for {ชื่อหลักทรัพย์ของคุณ} is: Very Bullish"เมื่อราคาปิดมากกว่า ราคาเฉลี่ย 30, 90, และ 200 วัน ในกรณีอื่นๆ Commentary จะแสดงค่า Bullish, Mildly Bullish, Mildly Bearish, Bearish หรือ Very Bearish ตามเงื่อนไข

โค้ดอื่นๆสามารถหาได้จาก AFL formula library โดยเฉพาะ MACD commentary ถึงตอนนี้ผู้ใช้ก็สามารถสร้าง Commentary เป็นของตัวเองได้แล้ว

การใช้งานเครื่องมือวาดบนสตร AFL

(Using studies in AFL formulas)

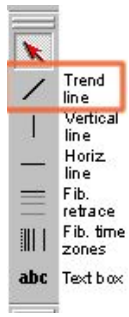
ตั้งแต่ AmiBroker เวอร์ชั่น 3.52 ขึ้นไป มีระบบการอ้างอิงเส้น หรือ สิ่งที่ผู้ใช้วาดด้วยตัวเองซึ่งถือว่าเป็นจุดเด่นเมื่อเทียบกับซอฟต์แวร์อื่นๆ

ผมจะสาธิตให้ดูวิธีการตรวจสอบว่าเส้นเทรนด์ไลน์นั้นเบรคหรือยัง สิ่งเราต้องทำ มีเพียงแค่ 3 ขั้นตอน

1. วาดเทรนด์ไลน์
2. กำหนด Study ID
3. เขียนสูตรที่ระบุว่า เส้นเทรนด์ไลน์ จะถูกเบรคอย่างไร

การวาดเส้นเทรนด์ไลน์ (Drawing trend line)

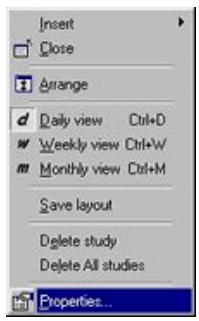
เส้นเทรนด์ไลน์ คือเส้นความชันที่ลากจากจุดหนึ่งไปยังจุดหนึ่ง



ในตัวอย่างนี้เราจะวาดเส้นเทรนด์ไลน์ขาขึ้น ซึ่งมักจะลากจากจุดต่ำสุดไปยังจุดต่ำสุดใกล้เคียงของเทรนด์ขาขึ้น หรือที่เรียกกันว่าเส้นแนวรับ

วิธีการวาดเส้นเทรนด์ไลน์นั้น ให้ผู้ใช้เลือก "Trend line" tool จาก "Draw" toolbar และหาจุดต่ำ (Troughs) อย่างน้อยสองจุดเพื่อลากเทรนด์ไลน์

กำหนด Study ID (Define study ID)



คุณสามารถดัดแปลงเส้นเทรนด์ไลน์ที่คุณวาดขึ้นมา เพียงคลิกขวาที่ตัวเส้นและ เลือก "Properties" ผู้ใช้สามารถระบุจุดกำเนิด และจุดจบของเส้นใหม่ได้ นอกจากนี้ผู้ใช้ยังสามารถเลือกสีเส้น ไสตล์ และยืดเส้นออกไปได้อีกตามต้องการ

ในตัวอย่างนี้หากผู้ใช้อยากจะศึกษาแนวโน้มในอนาคต ก็ให้เลือก right-extended trend line เพื่อให้เส้นเทรนด์ไลน์นั้นยืดออกไปทางด้านขวา

ตั้งแต่เวอร์ชั่น 3.52 ผู้ใช้สามารถกำหนด "Study ID" ได้ ซึ่งตัว "Study ID" นี้เอง ที่จะทำให้ผู้ใช้สามารถตั้งชื่อเส้นที่ผู้ใช้วาดขึ้นมาเองได้ โยที่มันจะถูกระบุด้วยตัวภาษาอังกฤษสองตัว โดยตัวอักษร Default คือ "UP" - uptrend, "DN" - downtrend, "SU" - support, "RE" - resistance, "ST" - stop loss แต่อย่างไรก็ตามคุณสามารถเรียกเส้นที่คุณวาดขึ้นมาตามต้องการ ภายใต้ข้อจำกัดที่ว่า ใส่ได้แค่ 2 ตัวอักษร ดังนั้นหากคุณวาดเส้นแนวรับในสัญลักษณ์หลายๆตัวแล้วตั้งชื่อมันว่า “SU” คุณสามารถนำตัวอักษร 2 ตัวนี้เป็นตัวอ้างอิงในโค้ดของคุณได้

ดังนั้นเราจะเรียกเส้นเทรนด์ไลน์ที่เราวาดขึ้นมาว่า "SU"

การเขียนโค้ดเพื่อระบุการเบรคของเส้นเทรนด์ไลน์ (Write the formula that checks trend line break)

ในตัวอย่างนี้เราจะหาจังหวะที่ราคาปิดเบรคเส้นเทรนด์ไลน์ที่เป็นแนวรับ ดังนี้:

```
sell = cross( study( "SU" ), close, GetChartID() );
```

สังเกตได้ว่า study() function คลุม 2 คำสั่ง :

อย่างแรกคือ StudyID ที่ระบุตัวอักษรภาษาอังกฤษสองตัว อ้างอิงเส้นที่ผู้ใช้วาด และ อย่างที่สองคือ chart ID ซึ่งคำสั่ง default นั้นคือ GetChartID() ซึ่งจะอ้างอิง indicator ปัจจุบัน ผู้ใช้สามารถศึกษาเพิ่มเติมในส่วนของ Parameter dialog, Axes & Grid: Miscellaneous: Chart ID ได้

การ Backtest ไอเดียการเทรดของคุณ

(Back-testing your trading ideas)

เกริ่นนำ (Introduction)

ผู้ใช้สามารถ Backtest กลยุทธ์การลงทุนต่างๆ กับ ข้อมูลในอดีตได้ ในหน้าต่าง Analysis ซึ่งทำให้ผู้ใช้สามารถทราบถึงจุดแข็งและจุดอ่อนของระบบการลงทุนก่อนลงทุนด้วยเงินจริง

การเขียนกฎการลงทุน (Writing your trading rules)

ขั้นตอนแรกผู้ใช้ต้องกำหนดเป้าหมาย หรือ กฎ ของระบบในการสั่งคำสั่งซื้อขาย ผู้ใช้ต้องคิดระบบขึ้นมาเองให้เหมาะสมกับความเสี่ยง ขนาดของพอร์ตโฟลิโอ การบริหารเงิน และ ปัจจัยต่างๆของตัวผู้ใช้เอง เมื่อได้กฎเหล่านั้นแล้ว นำมาเขียนเป็นสัญญาณซื้อขาย ใน AmiBroker Formula Language (รวมไปถึงการ Short และ การ Cover)

ในบทนี้เราจะมาดูระบบที่ใช้ Moving Average บอกสัญญาณซื้อขายกัน ระบบจะทำการซื้อหุ้นเมื่อราคาปิดตัดเหนือเส้น ema 45 และขายหุ้นตัวนั้นเมื่อราคาปิดตัดเส้น ema45 ลงมา exponential moving average ใน AFL คือ EMA ผู้ใช้เพียงแค่ระบุระยะเวลาเฉลี่ยของราคาปิดที่อยากจะนำมาคำนวณ

```
ema( close, 45 );
```

ตัว close เป็นค่า default ใน Amibroker บ่งบอกถึงราคาปิดของเครื่องหมายที่เราสนใจ

เราจะใช้คำสั่ง

```
buy = cross( close, ema( close, 45 ) );
```

คำสั่งด้านบนเป็นกฎการซื้อของระบบ ค่าที่ออกมาจะเป็น "1" หรือ "true" เมื่อราคาปิดตัดเหนือเส้น ema(close, 45) หลังจากนั้นเราสามารถเขียนคำสั่งขาย ซึ่งจะให้ค่า "1" เมื่อราคาปิดตัดเส้น ema(close, 45) ลงมาเช่นกัน:

```
sell = cross( ema( close, 45 ), close );
```

ข้อสังเกตคือเราใช้คำสั่ง Cross เพียงแต่สลับตัวแปรในนั้น ให้ตรงกันข้ามกัน คำสั่งทั้งสองจะออกมาในรูปแบบดังต่อไปนี้

```
buy = cross( close, ema( close, 45 ) );
sell = cross( ema( close, 45 ), close );
```

หมายเหตุ : หากจะเปิดแผงเขียนโค้ด Formula Editor สามารถเข้าได้จาก Analysis->Formula Editor พิมพ์โค้ด และ ส่งคำสั่ง Tools->Send to Analysis

การ Back Test (Back testing)

การ Backtest ระบบนั้นผู้ใช้เพียงกดคำสั่ง Back test ในหน้า analysis window ผู้ใช้ต้องมั่นใจว่าได้ใส่คำสั่งของระบบที่มีการกำหนดเงื่อนไขการ ซื้อ และ ขายไว้ก่อนทำการ Backtest เรียบร้อยแล้ว หลังจากนั้น Amibroker จะทำการวิเคราะห์เครื่องหมายที่ผู้ใช้เลือกอยู่ ตามระบบการลงทุน และ ให้ผลลัพธ์ออกมา ผู้ใช้สามารถ Backtest หลักทรัพย์เป็นพันๆตัวภายในไม่กี่นาที โดยที่ Amibroker จะคำนวณระยะเวลาโดยประมาณในการ Backtest ให้ผู้ใช้ได้เห็น หากผู้ใช้ต้องการหยุดการ Backtest เพียงกดปุ่ม cancel เท่านั้น

การวิเคราะห์ผลลัพธ์ (Analysing results)

เมื่อขั้นตอนการ Backtest เสร็จสิ้น จะมีหน้าต่างที่แสดงผลการจำลองการเทรดตามระบบ ที่ผู้ใช้ได้สร้างขึ้นมา (the Results pane) ผู้ใช้สามารถตรวจสอบสัญญาณซื้อขายเพียงดับเบิ้ลคลิก ตรงรายการ

จำลองเทรด ระบบจะแสดงจุดซื้อจุดขายเมื่อตรงตามเงื่อนไข ผู้ใช้สามารถเรียกหน้าต่างอื่นๆเพียงคลิกขวาตรงรายการเทรดจำลอง

หากผู้ใช้ต้องการผลการวิเคราะห์ผลให้ละเอียดขึ้นให้เข้าที่หน้าต่าง Report รายละเอียดของค่าต่างๆ จะอยู่ในส่วนของ report window description

การเปลี่ยน Setting (Changing your back testing settings)

ผู้ใช้สามารถเปลี่ยน Setting ของการ Backtest ซึ่งประกอบไปด้วย ขนาดของพอร์ตจำลอง ระยะเวลา จำนวนค่าคอมมิชชั่น อัตราดอกเบี้ย Maximum Loss และจุด Take profit ตามต้องการ หากเปลี่ยน Setting แล้ว ผู้ใช้ต้องกด Backtest อีกครั้ง เพื่อให้ค่าเป็นไปตาม Setting ที่เปลี่ยนไป

ตัวอย่างเช่น หาผู้ใช้ต้องการจะทดสอบข้อมูลรายอาทิตย์ให้เปลี่ยน Settings เป็น Weekly ในช่อง Periodicity และกด Back test อีกครั้ง

คำสั่ง Default (Reserved variable names)

ตัวแปร	ความหมาย	ใช้กับหน้าต่าง
buy	สั่งให้ทำการเข้าซื้อแบบ Long	Automatic Analysis, Commentary
sell	สั่งให้ทำการขายแบบปิด Long	Automatic Analysis, Commentary
short	สั่งให้ทำการเข้าซื้อแบบ Short	Automatic Analysis
cover	สั่งให้ทำการขายแบบปิด Short	Automatic Analysis

buyprice	ระบุราคาซื้อแบบ Long ตามต้องการ(หากไม่ระบุจะกำหนดตามค่า default ที่ตั้งไว้บน Automatic Analyser settings)	Automatic Analysis

sellprice	ระบุราคาขายแบบปิด Long ตามต้องการ(หากไม่ระบุจะกำหนดตามค่า default ที่ตั้งไว้บน Automatic Analyser settings)	Automatic Analysis
shortprice	ระบุราคาซื้อแบบ Short ตามต้องการ(หากไม่ระบุจะกำหนดตามค่า default ที่ตั้งไว้บน Automatic Analyser settings)	Automatic Analysis
coverprice	ระบุราคาขายแบบปิด Short ตามต้องการ(หากไม่ระบุจะกำหนดตามค่า default ที่ตั้งไว้บน Automatic Analyser settings)	Automatic Analysis
exclude	ไม่พิจารณาเครื่องหมายนั้นๆในโหมด scan/exploration/back test มีประโยชน์เมื่อผู้ใช้อยากจะโฟกัสการวิเคราะห์กับหุ้นกลุ่มๆหนึ่งโดยเฉพาะ	Automatic Analysis
roundlotsize	ใช้ในโหมด backtest เพื่อระบุหน่วยการลงทุน (ดูคำอธิบายเพิ่มเติมด้านล่าง)	Automatic Analysis (new in 4.10)
ticksize	กำหนด tick size เพื่อนำไปเทียบเคียงกับราคาที่เกิดขึ้นจาก built-in stops โดยที่ไม่ส่งผลกระทบต่อ buyprice/sellprice/shortprice/coverprice(ดูคำอธิบายเพิ่มเติมด้านล่าง))	Automatic Analysis (new in 4.10)
pointvalue	กำหนด contract point value ของ Future ได้(ดูในหมวดหมู่ backtesting futures) โดยทั่วไปแล้วจะตั้งค่าไว้ที่ 1	Automatic Analysis (new in 4.10)
margindeposit	กำหนด contract margin ของ Future ได้ (ดูในหมวดหมู่ backtesting futures)	Automatic Analysis (new in 4.10)
positionsize	กำหนดจำนวนเงิน หรือเปอร์เซ็นต์ของเงินทุนทั้งหมดในการเข้าซื้อแต่ละครั้ง (ดูคำอธิบายเพิ่มเติมด้านล่าง)	Automatic Analysis (new in 3.9)

คอนเซประดับสูง (Advanced concepts)

Amibroker นั้นมีคำสั่งที่ซับซ้อน ซึ่งจะกล่าวถึงในบทถัดๆไป ผู้ใช้ควรจะทำความเข้าใจคอนเซปการ Backtest เบื้องต้นก่อนเนื้อหาที่จะกล่าวถัดไป

เมื่อคุณพร้อมแล้ว เราจะมาแนะนำคำสั่งที่ได้กล่าวถึงเมื่อกี้

a) AFL scripting host for advanced formula writers (ขั้นตอนสำหรับนักเขียนโค้ดที่มีประสบการณ์)
b) enhanced support for short trades (การ Short)
c) the way to control order execution price from the script (การกำหนดราคาเข้าซื้อขาย)
d) various kinds of stops in back tester (การวาง Stop loss)
e) position sizing (การกำหนด Position Size)
f) round lot size and tick size (จำนวนหุ้นที่จะซื้อขาย)
g) margin account (บัญชี Margin)
h) backtesting futures (อื่นๆ)

AFL scripting host เป็นคำสั่งที่ซับซ้อนและจะถูกพูดถึงในหมวดหมู่ถัดๆไป เราจะมาแนะนำตัวอื่นๆ กัน

การชอร์ท (Short trade support)

เวอร์ชั่น 3.59 ขึ้นไปสามารถส่งคำสั่ง short ได้และคำสั่งอื่นๆ:

buy - "true" หรือมีค่าเท่ากับ 1 ให้เปิด long
sell - "true" หรือมีค่าเท่ากับ 1 ให้ปิด long
short - "true" หรือมีค่าเท่ากับ 1 ให้เปิด short
cover - "true" หรือมีค่าเท่ากับ 1 ให้ปิด short

หากผู้ใช้ต้องการใช้กลยุทธ์ stop-and-reverse system ให้แทนคำสั่งซื้อขายดังกล่าว

```
short = sell;
cover = buy;
```

ปัจจุบัน Amibroker สามารถส่งคำสั่ง long และ Short ได้ด้วยกัน

```
// กฎการเข้าซื้อและออกของ  long trade:

buy = cross( cci(), 100 );
sell = cross( 100, cci() );

// กฎการเข้าซื้อและออกของ  short trade:

short = cross( -100, cci() );
cover = cross( cci(), -100 );
```

ในกรณีนี้หาก CCI อยู่ระหว่าง -100 และ 100 คุณจะออกจากตลาด

การกำหนดราคาซื้อ (Controlling trade price)

AmiBroker สามารถให้ผู้ใช้กำหนดราคาซื้อขายตามต้องการ: buyprice, sellprice, shortprice และ coverprice

ผู้ใช้สามารถควบคุมราคาซื้อขายได้

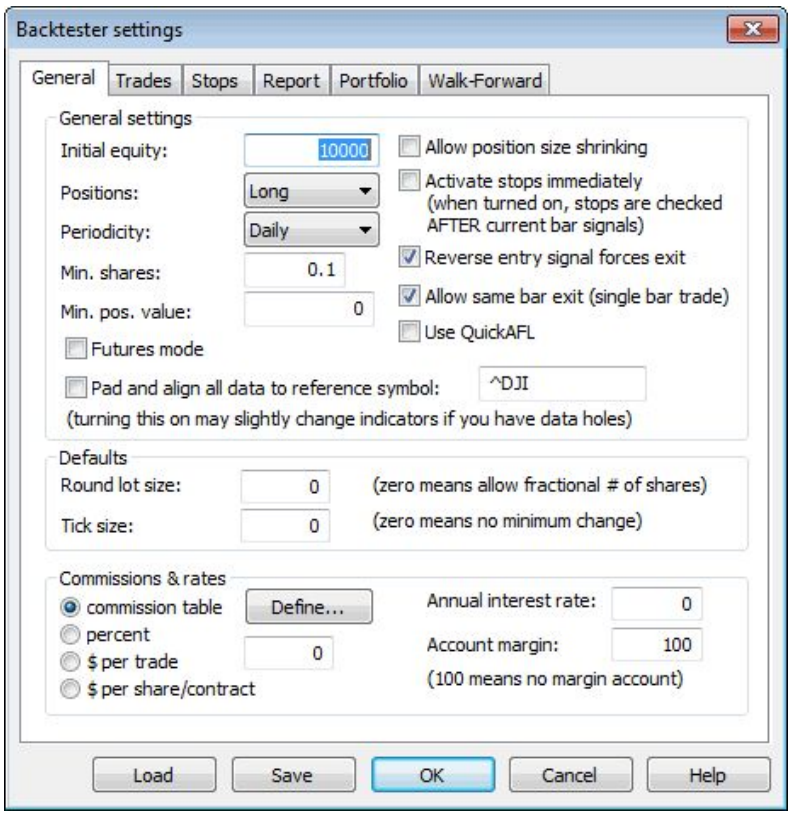
```
BuyPrice = IIF( dayofweek() == 1, HIGH, CLOSE );
//ส่งคำสั่งซื้อตอนวันจันทร์ ให้ซื้อราคา High วันอื่นๆให้ซื้อราคา Close
```

สามารถเขียนคำสั่ง Stop-Order:

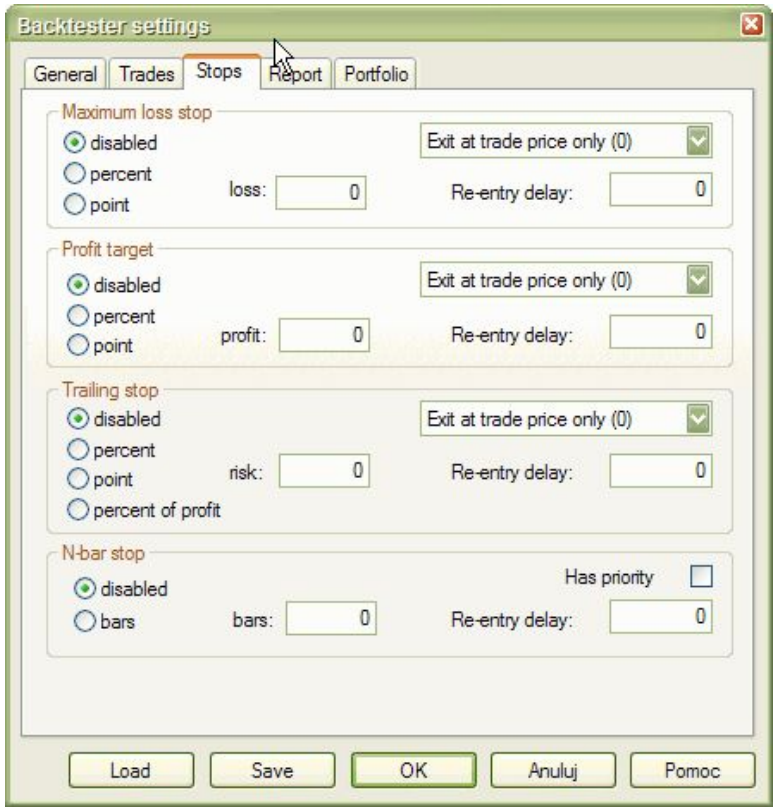
```
BuyStop = ...ใส่คำสั่งของ ระดับ Buy Stop;
SellStop = ... ใส่คำสั่งของระดับ Sell Stop;

// ถ้าหากราคายืนเหนือระดับ Buy Stop
//(high>buystop) ซื้อเมื่อราคาถึง buystop level

Buy = Cross( High, BuyStop );
// ถ้าหากราคาอยู่ต่ำกว่าระดับ Sell Stop
//( low< sellstop )ขายเมื่อราคาต่ำกว่า Sellstop level
Sell = Cross( SellPrice, SellStop);

BuyPrice = max( BuyStop, Low ); //ให้มั่นใจว่าราคาซื้อไม่ต่ำกว่า ราคาต่ำสุด
SellPrice = min( SellStop, High ); //ให้มั่นใจว่าราคาขายไม่สูงไปกว่า ราคาสูงสุด
```

AmiBroker ระบุ buyprice, sellprice, shortprice และ coverprice เป็นคำสั่ง Default

ขณะ back-testing AmiBroker จะเช็คว่าราคา buyprice, sellprice, shortprice, coverprice อยู่ในกรอบของ high-low ของแต่ละวันหรือไม่ แล ะจะปรับราคาซื้อขายเหล่านั้น ให้เป็น high price เมื่อราคาสูงกว่า ราคาที่สูงที่สุด และ ให้เป็น low price เมื่อราคาต่ำกว่า ราคาต่ำที่สุดในวันนั้น


Backtester settings
General
Trades
Stops
Report
Portfolio
Walk-Forward
General settings
Initial equity:
10000
Positions:
Long
Periodicity:
Daily
Min. shares:
0.1
Min. pos. value:
0
Futures mode
Allow position size shrinking
Activate stops immediately
(when turned on, stops are checked
AFTER current bar signals)
Reverse entry signal forces exit
Allow same bar exit (single bar trade)
Use QuickAFL
Pad and align all data to reference symbol:
^DJI
(turning this on may slightly change indicators if you have data holes)
Defaults
Round lot size:
0
(zero means allow fractional # of shares)
Tick size:
0
(zero means no minimum change)
Commissions & rates
commission table
Define...
percent
$ per trade
$ per share/contract
0
Annual interest rate:
0
Account margin:
100
(100 means no margin account)
Load
Save
OK
Cancel
Help


Backtester settings
General
Trades
Stops
Report
Portfolio
Long trades
Buy price:
Close
Buy delay:
0
Sell price:
Close
Sell delay:
0
Short trades
Short price:
Close
Short delay:
0
Cover price:
Close
Cover delay:
0
Load
Save
OK
Anuluj
Pomoc



การตั้ง Stop (Profit target stops)

ดังที่เห็นกันบนภาพด้านบนของ Setting สำหรับ Profit Target Stop นั้น ระบบจะทำการ Take Profit เมื่อ ราคาสูงที่สุดของวันนั้นอยู่เหนือ Stop Level ที่ตั้งกันเอาไว้ ซึ่งอาจจะอยู่ในรูปของ Percent จากราคา ซื้อ โดยทั่วไป Stop จะถูกดำเนินการตามราคา Sell และ ราคา Cover ที่ผู้ใช้ตั้งเอาไว้ ซึ่งสามารถปรับเปลี่ยนบน "Exit at stop" feature ได้

"Exit at stop"

ตัวอย่างเช่นเมื่อคุณเลือก Exit Stop แล้วตั้งค่าไว้ที่ +10% จากราคาซื้อ 50 ตัว stop order จะทำงานเมื่อราคาขึ้นไปสู่ 55 ถึงแม้ว่าคุณจะกำหนดราคาขายด้วยตัวเลขอื่นก็ตาม (เช่นราคาปิดที่ 56)

Maximum loss stops ก็ทำงานเช่นเดียวกัน Stop จะทำงานเมื่อราคาต่ำที่สุดของวันนั้น ต่ำกว่าจุด Stop ที่ตั้งเอาไว้

Trailing stops

การ Stop ชนิดนี้จะเป็นตัว Stop ที่เคลื่อนที่ตามราคา New High ของแต่ละวัน ทำให้ตัวจุด Stop เองไม่อยู่นิ่ง หากเมื่อราคาลงมาต่ำกว่า trailing stop ที่เคลื่อนที่ตามราคา new high ล่าสุด Stop จะทำการทันที ดังภาพตัวอย่างดังกล่าว (10% Trailing Stop)

ตัว Stop ที่กล่าวมานั้นสามารถปรับเปลี่ยนจาก Automatic analysis' Settings window หรือ ผู้ใช้สามารถโค้ดเอาเองได้ ผ่าน ApplyStop function:

ตัวอย่างเช่น :

```
ApplyStop( 2, 1, 10, 1 ); // 10% trailing stop, percent mode, exit at stop ON
```

หรือเขียนในรูปแบบ

```
ApplyStop( stopTypeTrail, stopModePercent, 10, True );
```

Trailing stops ยังสามารถในรูปแบบของ points (dollars) และ percent of profit (risk) สำหรับข้อมูลตัวที่สองนั้นหากผู้ใช้ตั้ง Stop ไว้ที่ 20% percent ของ profit (risk) ตัว stop จะทำงานเมื่อโอกาสทำกำไรคือ $100 แต่ผลกำไรนั้นตกไปเหลือ $80

Dynamic stops

ฟังก์ชั่น ApplyStop() ให้ผู้ใช้สามารถเลือกลักษณะของ Stop ต่อแต่ละการ Trade ได้ ซึ่งผู้ใช้อาจจะสร้าง volatility-based stop ต่างๆ ขึ้นมาเองก็เป็นได้

ตัวอย่างเช่น การสร้าง Stop ที่จะปรับเปลี่ยนตาม 10 day average true range:

```
ApplyStop( 0, 2, 2 * ATR( 10 ), 1 );
```

หรือ

```
ApplyStop( stopTypeLoss, stopModePoint, 2 * ATR( 10 ), True );
```

จะทำให้ ตัว Stop อยู่ห่างจากราคาเข้าซื้อ ลงมา 2 เท่าของ 10 Days ATR

เมื่อ ATR เปลี่ยนจากการเทรดในแต่ละครั้ง Stop level จะมีลักษณะที่ dynamic และ เปลี่ยนตามเช่นกัน ข้อสังเกตคือ Parameter ที่ 3 ของฟังก์ชัน ApplyStop (the amount) ขึ้นอยู่กับราคา Entry และจะอยู่จนกว่าการ trade ครั้งนั้นจะจบสิ้น ดังนั้นในตัวอย่างที่เราตั้งค่า Stop ไว้ตาม ATR(10) จากวัน entry หากผู้ใช้ปรับแต่ง ATR จะไม่มีผลต่อ Stop ที่ตั้งเอาไว้

ดูรายละเอียดเพิ่มเติมได้ที่ APPLYSTOP

การเขียนโค้ดสร้างจุด Stop (Coding your own custom stop types)

ผู้ใช้สามารถตั้งโค้ด ApplyStop ขึ้นมาเองตามต้องการ

ตัวอย่างเช่น

```
/* ตัวอย่างการตั้ง Profit-target stop บน AFL: */

Buy = Cross( MACD(), Signal() );
priceatbuy=0;

for( i = 0; i < BarCount; i++ )
{
	if( priceatbuy == 0 && Buy[ i ] ) priceatbuy = BuyPrice[ i ];

	if( priceatbuy > 0 && SellPrice[ i ] > 1.1 * priceatbuy )
	{
		Sell[ i ] = 1;
		SellPrice[ i ] = 1.1 * priceatbuy;
```

```
            priceatbuy = 0;
        }
        else
            Sell[ i ] = 0;
    }
```

Position sizing

เวอร์ชั่นใหม่ 3.9 ปรับปรุงการออกแบบ Position sizing

```
PositionSize = <size array>
```

ผู้ใช้สามารถคุมจำนวนเงินลงทุน ตามจำนวนหรือ % ของ portfolio ต่อแต่ละการเทรดได้

ระบุจำนวนเงินตามต้องการ trade
ยกตัวอย่างเช่น:

- PositionSize = 1000; // ลงทุน $1000 ในทุกๆการเทรด
- ตัวเลขติดลบบ่งบอกถึงหน่วยที่เป็น % เช่น -100..-1 มีหน่วยเป็น %:
  -100 คือ 100% ของ portfolio size,
  -33 คือ 33% ของ equity เป็นต้น:

```
PositionSize = -50; /* ลงทุนแค่ครึ่งหนึ่งของ equity */
```

หรือ แม้แต่

```
PositionSize = - 100 +RSI();
```

ค่า RSI เปลี่ยนจาก 0..100 ค่า RSI ที่ต่ำทำให้คุณลงทุนมากขึ้น

หากคุณไม่ได้ลงเงินทั้งหมด 100% เงินที่เหลือจะนำไปทบตามดอกเบี้ยที่ผู้ใช้ตั้งเอาไว้

นอกจากนี้ผู้ใช้สามารถเลือกคำสั่ง "Allow position size shrinking" ใน Setting กรณีที่ position size เกินจำนวนเงินที่ถืออยู่ เมื่อผู้ใช้เลือกคำสั่งนี้ position ในการเข้าลงทุนจะลดตาม Cash ที่เหลืออยู่

หากต้องการจะดู position sizes ให้ผู้ใช้เลือก "Trade list with prices and positionsize" ใน report mode บน AA settings window

ก่อนจะจบในส่วนนี้ นี่คือตัวอย่างของ Position size ที่เปลี่ยนไปตามค่า ATR :

```
Buy = <ใสเงื่อนไขของการซื้อของคุณที่นี่>
Sell = 0; //ขายเฉพาะเมื่อถึง Stop

TrailStopAmount = 2 * ATR( 20 );
Capital = 100000; /* หมายเหตุ: ให้ตั้งค่านี้บนหน้าต่าง Setting ในช่อง Initial Equityด้วย*/
 Risk = 0.01*Capital;
PositionSize = (Risk/TrailStopAmount)*BuyPrice;
ApplyStop( 2, 2, TrailStopAmount, 1 );
```

สรุปใจความได้ดังนี้ : total equity ต่อ 1 การลงทุนคือ $100,000, และ ตั้งระดับความเสี่ยงไว้ที่ 1% ของ total equity ระดับความเสี่ยหรือ Risk level ถูกกำหนดโดยเงื่อนไขดังนี้ ถ้าหาก trailing stop ของหุ้นที่ราคา $50 stock อยู่ที่ $45 (ค่าของ 2 ATR ตามที่เซ็ทไว้ใน Trilling Stop), ระดับขาดทุน $5 นั้นจะถูกนำไปหารด้วย $1000 risk และได้จำนวนหุ้นที่ควรจะซื้อตามความเสี่ยงที่รับได้ที่ 200 หุ้น ดังนั้นความเสี่ยงของการลงทุนในครั้งนี้ถูกตั้งขึ้นไว้ที่ $1000 แต่การกระจายความเสี่ยงนั้นขึ้นอยู่กับการซื้อหุ้น 200 หุ้น x $50/หุ้น หรือ $10,000 สรุปคือ เราใช้เงินเพียง 10% ของ Equity ในการลงทุนแต่มีความเสี่ยงที่จะเสียเงินลงทุนเพียงแค่ $1000

จำนวนหน่วยการลงทุน และ การเคลื่อนไหวของราคา (Round lot size and tick size)

Round lot size

ผลิตภัณฑ์ทางการเงินทุกชนิดย่อมมีหน่วยในการลงทุน Amibroker สามารถให้ผู้ใช้เลือกได้ว่า การลงทุนในแต่ละครั้งจะต่อซื้อผลิตภัณฑ์นั้นๆกี่หน่วยเช่น ในหลัก 10s หรือ 100s (จากผู้แปล : เช่น ในตลาดหุ้นไทยขั้นต่ำคือต้องซื้อ 100 หุ้น)

คุณสามารถเลือก หน่วยการลงทุน ต่อแต่ละหลักทรัย์ได้ในหน้าต่าง Symbol -> Information page (รูปที่. 3) จำนวนศูนย์หมายถึงไม่ระบุหน่วยลงทุน และ จะถูกตั้งเป็นค่า "Default round lot size" (การตั้งค่าสากล) จาก Automatic Analysis settings page (รูปที่ 1) ในกรณีนี้ผู้ใช้สามารถให้ระบบเข้าซื้อผลิตภัณฑ์ทางการเงินนั้นด้วยหลัก เศษส่วน (fractional number) เช่นกัน

ผู้ใช้สามารถเขียนโค้ดระบุ lot size ผ่าน AFL formula ด้วยฟังก์ชัน RoundLotSize

ตัวอย่างเช่น:

```
RoundLotSize = 10;
```

Tick size

Tick Size ระบุการเคลื่อนไหวของราคาที่ต่ำที่สุด (minimum price move) ผู้ใช้สามารถระบุได้ทั้งหมดหรือ แค่หุ้นรายตัว บนคำสั่ง Symbol->Information page (รูปที่ 3) เลข 0 จะหมายความว่าตั้งค่า "default tick size" บนหน้า Settings page (รูปที่ 1) ซึ่งหมายความว่าไม่ระบุค่าต่ำสุดของการเคลื่อนไหวของราคา

ผู้ใช้สามารถเขียนโค้ดระบุ tick size ผ่าน AFL formula ด้วยฟังก์ชัน TickSize ตัวอย่างเช่น:

```
TickSize = 0.01;
```

tick size setting จะมีผลต่อเทรดที่ Exit โดยใช้ built-in stops หรือ ApplyStop() เท่านั้น ตัว backtester นั้นจะอนุมาน tick size requirements ที่มากับตัวราคาเองอยู่แล้ว ดังนั้นการเปลี่ยนค่า tick size จะมีผลต่อแค่การเทรดที่มีการตั้ง built-in stops ทำให้จุด exit points นั้นเปลี่ยนแปลงตาม tick size ที่ผู้ใช้ระบุ ตัวอย่างเช่น ที่ประเทศญี่ปุ่น ผู้ใช้ต้องตั้งค่า ticksize = 1 ทำให้การ exit นั้นจะ exit ในจุดที่เป็นจำนวนเต็ม

บัญชี Margin (Margin account)

Account margin ระบุ % ของ margin requirement สำหรับ entire account ค่า Default ของ Account margin จะอยู่ที่ 100 ซึ่งหมายความว่าผู้ใช้จะต้องลงเงิน 100% จำนวนเต็มในการลงทุน เวอร์ชั่นนี้ Amibroker ให้ผู้ใช้สามารถตั้งค่า Margin ได้ ซึ่งก็คือเงินที่ยืมมาเพื่อใช้ในการลงทุน ตัวอย่างเช่น ถ้ากฎระเบียบปัจจุบันอนุโลมให้ใช้เงิน Margin 50% ของการลงทุน หากเงินเริ่มต้นของคุณคือ 10000 คุณจะมีกำลังในการซื้อถึง 20000 ทำให้ผู้ใช้สามารถลงทุนได้มากขึ้น เงิน Margin นี้ไม่เกี่ยวข้องกับการเทรด Futures แต่อย่างใด ผู้ใช้สามารถจำลองการเทรดหุ้นโดยใช้เงิน Margin ที่ว่านี้

Setting อื่นๆ (Additional settings)

- "Reverse entry signal forces exit" check box ใน Backtester settings
  เมื่อปรับค่าเป็น ON (default) หมายความว่าถ้าผู้ใช้เปิด Position ตรงข้ามจะหักล้าง Position ก่อนหน้านี้ ถ้าตั้งค่าเป็น OFF การเปิด Position ตรงข้ามจะไม่มีผลต่อ Position ที่เปิดค้างเอาไว้

- "Allow same bar exit (single bar trade)" ทางเลือกใน Settings
  เมื่อปรับค่าเป็น ON (default) ผู้ใช้สามารถเข้า และ ออก Postitionใน Bar เดียวๆกันทั้ง entry และ exit ถ้าตั้งค่าเป็น Off หมายความว่า การ exit จะเกิดขึ้นใน bar ถัดๆไป เท่านั้น

- "Activate stops immediately"เหมาะสำหรับผู้ใช้ที่ตั้งราคาซื้อเป็นราคา Open เพราะปัญหาในเวอร์ชั่นเก่าที่ backtester อนุมานว่าผู้ใช้จะเข้าซื้อในราคาปิด ถ้าหากผู้ใช้ไม่เลือกช่องนี้ ระบบจะอนุมาน Normal Exit ซึ่งก็คือ Sell ในราคา open ของ bar ถัดไป

ผู้ใช้อาจจะสงสัยว่าทำไมไม่เลือก buyprice หรือ shortprice ในกรณีที่เท่ากับราคาเปิด เป็นเพราะวันที่เกิด doji ระบบไม่สามารถแยกแยะได้ว่าการ Trade เข้าตอนราคาปิดหรือเปิด

- "Use QuickAFL" QuickAFL(tm) เป็น Feature ที่ทำให้ AFL คำนวณตัวเลขได้เร็วขึ้นในบางสภาพ สำหรับเวอร์ชั่น 5.14 ขึ้นไปสามารถใช้ในหน้าต่าง Automatic Analysis ได้

จุดเริ่มต้นของ Feature นี้เกิดขึ้นเพื่อให้ระบบคำนวณและวาด Chart ได้เร็วขึ้น หน้าต่าง automatic analysis window สามารถใช้ประโยชน์จากคำสั่งนี้เมื่อ parameter ของข้อมูลนั้นน้อยกว่า“All quotations"

คำอธิบายอย่างละเอียดของ Feature นี้อยู่ในลิงค์ดังกล่าว:
http://www.amibroker.com/kb/2008/07/03/quickafl/

Feature นี้สามารถใช้งานได้ทั้งใน backtester, optimization, exploration และ scan
สามารถดูในได้ในหัวข้อ :

Portfolio-level backtesting

Backtesting systems for futures contracts APPLYSTOP function description
Using AFL editor section of the guide.

Insider guide to backtester (newsletter 1/2002)

Portfolio-level backtesting

สำคัญ: โปรอ่านคู่มือนี้มาก่อนl: Backtesting your trading ideas article

Backtest ตัวใหม่นี้สามารถใช้งานได้ในระดับพอร์ตโฟลิโอ ซึ่งอ้างอิงกับ Portfolio Equity นั่นก็คือจำนวนเงินทั้งหมดที่เหลืออยู่รวมกับ Position ที่เปิดเอาไว้ portfolio backtester ทำให้ผู้ใช้สามารถผสมผสาน กลยุทธ์การลงทุน money management และ การควบคุมความเสี่ยงในระดับพอร์ตโฟลิโอ เสมือนการเทรดจริง Position size ก็จะปรับระดับตามพอร์ตโฟลิโอเช่นกัน

ขั้นตอนการติดตั้ง (HOW TO SET IT UP ?)

มีเพียง 2 ขั้นตอนในการ Setup

1. ผู้ใช้ต้องมีโค้ดที่ระบุ buy / sell / short /cover signals ดังที่อธิบายไว้ใน "Backtesting your trading ideas"

2. ผู้ใช้ต้องระบุจำนวนในการเทรดและ Position size ของการเทรดในแต่ละครั้ง

การตั้งจำนวน position สูงสุดในการเทรดหนึ่งครั้ง (SETTING UP MAXIMUM NUMBER OF SIMULTANEOUSLY OPEN TRADES)

มี 2 วิธีในการตั้งค่า จำนวนการเข้าเทรดที่สูงที่สุด

1. เข้าไปยังส่วนของ Settings เลือกหน้า Portfolio และ ใส่ตัวเลขลงไปในช่อง Max. Open Positions

2. หรือระบุจำนวนผ่านโค้ด โดยระบุฟังก์ชั่น SetOption :

```
SetOption("MaxOpenPositions", 5 );
```

// ในตัวอย่างคือการตั้งค่าให้ซื้อได้มากที่สุด 5 position

การตั้งขนาดการเทรดต่อหนึ่งการเทรด (SETTING UP POSITION SIZE)

สำคัญ: หากต้องการให้ระบบเข้าเทรดมากกว่า 1 ตัวผู้ใช้ต้องตั้งค่า Positionsize ตามกลยุทธ์ และน้อยกว่า 100% สำหรับการลงทุนในแต่ละตัว มิฉะนั้นจะไม่เหลือเงินให้ลงทุนในหุ้นตัวอื่นๆ เช่น

```
PositionSize = -25; // ในการลงทุนหนึ่งครั้งใช้จำนวนเงิน 25% ของ Portfolio Equity
```

หรือ

```
PositionSize = 5000; // ในการลงทุนหนึ่งครั้งใช้จำนวนเงิน $5000
```

วิธีการตั้งค่า positionsize และ max. open positions ให้สอดคล้องกันนั้นสามารถเขียนโค้ดได้ดังนี้

```
PosQty = 5; // ระบุจำนวน positon ที่อยากเข้าเทรด SetOption("MaxOpenPositions",
PosQty );
PositionSize = -100/PosQty; // ลงทุน 100% ของ portfolio equity หารด้วย by
max.position count
```

คุณสามารถเลือกใช้ กลยุทธ์ของ Position size ที่มีความซับซ้อนขึ้น ตัวอย่างเช่น Position Size ที่ขึ้นอยู่กับ

Volatility ของหุ้น (Van Tharp-style) :

```
PositionSize = -2 * BuyPrice/(2*ATR(10));
```

ผู้ใช้จะลงทุนด้วยเงิน 2% ของ PORTFOLIO equity และหารด้วย 2*ATR factor

การใช้ Position Score (USING POSITION SCORE)

ในกรณีที่ entry signals นั้นมีมากกว่าจำนวนเงินที่ตั้งไว้ในการเข้าซื้อ ผู้ใช้สามารถตั้ง Position Score เพื่อให้ระบบให้ความสำคัญกับหุ้นตามที่ผู้ใช้ระบุได้ ดังโค้ดดังกล่าว เป็นกกลยุทธ์ MA crossover system และ ให้ความสำคัญกับหุ้นที่มี RSI ต่ำ เมื่อเกิดสัญญาณซื้อที่มากกว่าจำนวนเงินที่ผู้ใช้ถืออยู่ ระบบจะคัดกรองหุ้นตาม RSI หุ้นที่มี RSI ต่ำที่สุดจะเข้าเทรดก่อนตัวอื่นๆ ผู้ใช้สามารถตรวจสอบเบื้องหลังการทำงานของระบบเพียงคลิกช่อง "Detailed log" บนหน้า report ให้เป็น on

โค้ดด้านล่างดังกล่าว ใช้ระบบเพื่อหาจำนวน open positions ที่เหมาะสมที่สุด (Optimal)

```
/*****
** โหมด REGULAR PORTFOLIO
** ตัวอย่าง optimization นี้ จะหาจำนวน position ที่ควรเปิดพร้อมกันในการเทรด
**
****/

SetOption("InitialEquity", 20000 );
SetTradeDelays(1,1,1,1);
RoundLotSize = 1;

posqty = Optimize("PosQty", 4, 1, 20, 1 );
SetOption("MaxOpenPositions", posqty);

// position size ที่ต้องการคือ 100% ของ portfolio equity
// หารด้วย PosQty positions PositionSize = -100/posqty;
// ระบบนี้ไม่มีความซับซ้อน
//เราสามารถทำการ optimize parameters ของ MA เช่นกัน
```

```
p1 = 10;
p2 = 22;

// MA crossover อย่างง่าย
Short=Cross( MA(C,p1) , MA(C,p2) );
Buy=Cross( MA(C,p2) , MA(C,p1) );

Sell=Short;
Cover=Buy;

// หากมีสัญญาณเข้าซื้อมากกว่าเงินลงทุนให้สร้าง Score ขึ้นมา

PositionScore = 100-RSI();

// ให้ความสำคัญกับหุ้นที่มี low RSI อันดับแรก;
```

BACKTEST MODES

AmiBroker เวอร์ชั่น 5.0 ให้เลือกวิธีการ backtest ถึง 6 วิธี:

- regular mode (backtestRegular)
- regular raw mode (backtestRegularRaw)
- regular raw + multiple positions mode (backtestRegularRawMulti)
- regular raw2 mode (backtestRegularRaw2)
- regular raw2 + multiple positions mode (backtestRegularRaw2Multi)
- rotational trading mode (backtestRotational)

โหมด "regular"ทุกตัวใช้สัญญาณเข้าออกตาม buy/sell/short/cover signals ในขณะที่โหมด "rotational" mode (aka "ranking / switching" system) ใช้ position score อย่างเดียวซึ่งจะถูกอธิบายไว้ในส่วนต่อไป

ผู้ใช้สามารถเปลี่ยนโหมดการ Backtest ได้ผ่านฟังก์ชั่น SetBacktestMode() AFL function

ความแตกต่างของ "regular" modes แต่ละตัวขึ้นอยู่กับการทำการต่อสัญญาณเข้าซื้อ ของระบบ ("redundant" หรือ "extra") สัญญาณซื้อที่ "extra" เกิดหลังจากที่มีการเปิด Position หนึ่งไว้แล้ว แต่ Position นั้นยังไม่ถูกปิด ตัว Extra Position คือ ตัวที่เกิดระหว่าง 2 จุดนั้น

ในโหมดของ regular ซึ่งเป็นโหมด Default นั้น redundant entry จะถูกตัดออกไปทันที ดังภาพด้านล่าง (กล่องที่ 2)

RAW SIGNALS

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
AAPL	Sell	Buy	Buy	Buy	Sell	Sell	Buy	Buy	Buy	Buy	Buy	Buy	Buy	Sell	Sell
MSFT	Sell	Sell	Buy	Sell	Sell	Buy	Sell	Sell	Sell	Sell	Sell	Buy	Sell	Sell	Sell
INTC	Buy	Buy	Buy	Buy	Buy	Buy	Buy	Buy	Buy	Buy	Buy	Sell	Sell	Sell	Sell
CSCO	Buy	Buy	Buy	Sell	Sell	Sell	Buy	Buy	Buy	Buy	Sell	Sell	Sell	Buy	Buy
LVLT	Sell	Buy	Buy	Buy	Buy	Sell	Sell	Sell	Sell	Buy	Buy	Buy	Sell	Sell	Sell
AMGN	Buy	Buy	Sell	Sell	Buy	Buy	Buy	Sell	Sell	Buy	Buy	Buy	Buy	Sell	Sell

PHASE 1 - POTENTIAL TRADES - MATCHING BUY WITH SELL SIGNALS - EXTRA SIGNALS REMOVED

Numbers in parentheses mean the POSITION SCORE at entry signal

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
AAPL		Buy(10)			Sell		Buy(8)							Sell	
MSFT			Buy(6)	Sell		Buy(1)	Sell					Buy(1)	Sell		
INTC	Buy(3)											Sell			
CSCO	Buy(5)			Sell			Buy(10)				Sell			Buy(1)	
LVLT		Buy(8)				Sell				Buy(3)			Sell		
AMGN	Buy(4)		Sell		Buy(3)			Sell		Buy(2)				Sell	

PHASE 2 - PICKING TOP TRADES - MAX OPEN POS = 2, TRADES PICKED HAVE HIGHEST SCORE, ONCE PICKED, REMAIN IN PLACE

GREY COLOR MEANS SKIPPED TRADE

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
AAPL		Buy(10)			Sell		Buy(8)							Sell	
MSFT			Buy(6)	Sell		Buy(1)	Sell					Buy(1)	Sell		
INTC	Buy(3)											Sell			
CSCO	Buy(5)			Sell			Buy(10)				Sell			Buy(1)	
LVLT		Buy(8)				Sell				Buy(3)			Sell		
AMGN	Buy(4)		Sell		Buy(3)			Sell		Buy(2)				Sell	

ดังที่คุณเห็นกันในภาพ หาก BUY เจอกับ SELL จะเรียกว่าการ TRADE หากระบบไม่สามารถ BUY เนื่องจาก Rank ต่ำ (ตามที่ตั้งไว้ใน Position Score) เงินไม่พอ หรือ ถือ Position ตามจำนวน Max Open Position ไว้แล้ว การ Entry หุ้นตัวใหม่จะเกิดขึ้นก็ต่อเมื่อมีสัญญาณ Exit ของหุ้นตัวเก่าเกิดขึ้น กระบวนการของการตัด Excess Signal ที่ว่านี้สามารถเรียกโดยใช้ฟังก์ชั่น ExRem() อย่างไรก็ตาม ลักษณะที่กล่าวมาจะอยู่ใน Backtest โหมดแบบ Default อยู่แล้วดังนั้นผู้ใช้ไม่ต้องเรียกฟังก์ชั่น SetBacktestMode ขึ้นมาเพื่อตั้งค่าดังกล่าวเอง

หากผู้ใช้ต้องการให้ระบบตอบสนองต่อสัญญาณซื้อทุกๆ ตัวผู้ใช้ต้องปรับโหมดเป็น backtestRegular Raw โดยใส่โค้ดดังกล่าว

```
// ใช้โหมด signal-based backtest โดยที่ redundant (raw) signals
ไม่ได้ถูกตัดออกไปแต่เข้าเครื่องหมายละ 1 position เท่านั้น

SetBacktestMode( backtestRegularRaw );
```

โหมดดังกล่าวไม่ตัด redundant entry signals และ จะตอบสนองต่อ ทุกๆ Entry Signal ที่มี Score สูงตามที่ตั้งไว้ใน Position Score แต่ตัวระบบจะเข้าเพียง 1 position ต่อ หุ้น 1 ตัวเท่านั้น หมายความว่าถ้าระบบเข้าซื้อหุ้นตัวนั้นไปแล้ว จะไม่เข้าซื้อเมื่อเกิดสัญญาณ redundant entry จะเข้าซื้อหลังจาก Sell signal เท่านั้น อย่างไรก็ตามผู้ใช้สามารถใช้คำสั่ง sigScaleIn/sigScaleOut เพื่อเพิ่ม/ลด ขนาดของ Position ทั้งนี้ค่าใน Report จะโชว์เพียงบรรทัดเดียวเท่านั้น (ไม่ระบุการเข้าแยกออกจากกัน)

ถ้าผู้ใช้ต้องการจะตอบสนองต่อทุกๆ Entry Signal และให้ Report โชว์การเข้าซื้อแยกออกจากกันให้ใช้คำสั่ง backtestRegularRawMulti ตามโค้ดดังกล่าว

```
SetBacktestMode( backtestRegularRawMulti );
```

ในโหมดนี้ ระบบจะเปิด MULTIPLE positions ต่อหุ้นเมื่อมีสัญญาณเกิดขึ้น และมีเงินสดเหลืออยู่ หากผู้ใช้ใส่คำสั่ง Scale-In/Out จะมีผลต่อทุกๆ open positions

หมายเหตุ : เนื้อหาถัดไปเหมาะสำหรับผู้มีประสบการณ์ต่อการเขียนระบบ

โหมด Raw2 เป็นโหมดพิเศษสำหรับผู้ใช้ที่จะทำ custom backtest หากผู้ใช้ ใช้แค่เพื่อการ Backtest แบบธรรมดาไม่ควรตั้งค่าโหมดนี้ ด้วยเหตุผลที่ว่าจะมี performance hit ที่ไม่เหมือนโหมดปกติ และ กินที่ memory มากกว่า

ความเหมือนระหว่างโหมด Raw และ Raw2 คือทั้งสองโหมดไม่ตัดสัญญาณ excess entry signals ออก ความแตกต่างคือ โหมด Raw จะตัด excess EXIT signals แต่โหมด Raw2 จะไม่ตัดออก

โหมด Raw2 จะเก็บสัญญาณ Excess exit signals ในกรณีที่ผู้ใช้อยากใช้กลยุทธ์ที่ข้ามสัญญารออกตัวแรกไปออกสัญญาณถัดๆไป สมมติว่าผู้ใช้อยากให้ระบบ exit ตามเงื่อนไขใดเงื่อนไขหนึ่ง แต่เงื่อนไขนั้นกลับอยู่ภายใต้เงื่อนไขอีกตัว เช่น ภายในระยะเวลาหนึ่ง หลังจากจำนวนบาร์ตามที่กำหนด หรือมีเงื่อนไข Portfolio equity ผู้ใช้สามารถเลือใช้โหมด Raw2 ดังกล่าวได้

ดังนั้นการ Backtest ในโหมดนี้จะใช้เวลามากกว่าปกติเนื่องจากระบบตอบสนองต่อทุกๆ Exit Signal โหมด Raw2 ยังกิน memory มากกว่าโหมดอื่นอีกเช่นกัน แต่อย่างไรก็ตามหากผู้ใช้ไม่ได้เขียนระบบที่ต้อง run Custom Backtest การใช้โหมด Raw หรือ Raw2 ก็จะไม่มีความแตกต่างใดๆทั้งสิ้น

ROTATIONAL TRADING

รายละเอียดของ Rotational trading ซึ่งมีสัญญาณเข้าออกตาม position score นั้นจะถูกอธิบายในหมวดของ EnableRotationalTrading

HOLDMINBARS และ EARLY EXIT FEES (HOLDMINBARS AND EARLY EXIT FEES)

HoldMinBars เป็น feature ที่ระงับสัญญาณการออก รวมไปถึงสัญญาณ Stop ทั้งหลาย รวมไปถึงการคำนวณ Trailling Stop จะไม่คำนึงถึง Bar ที่อยู่ภายใต้เงื่อนไข HoldMinBars นี้

ฟังก์ชันนี้ และ EarlyExitFee/EarlyExitBars สามารถนำไปใช้สำหรับหุ้นรายตัวได้

ตัวอย่างเช่น:

```
SetOption("HoldMinBars", 127 );
Buy=BarIndex()==0; Sell=1;

 // ถึงแม้ว่าสัญญาณการขายจะเกิดขึ้นทุกวัน แต่สัญญาณเหล่านั้นจะถูกเพิกเฉยจนกว่าจะถึง bar ที่ 128
```

ค่าใช้จ่าย Early exit (redemption) fee เกิดขึ้นเมื่อสัญญาณออกอยู่ในระยะเวลาของ N Bars หลังจากเกิดสัญญาณเข้า ค่าใช้จ่ายนี้จะรวมไปอยู่กับ Commission Fees และจะเห็นได้ใน Commission report

```
// คำสั่ง 2 ตัวนี้สามารถตั้งค่าต่อ 1 สัญลักษณ์
// และจำนวน Bar ได้
// เลือกกำหนด early exit (redemption) fee SetOption("EarlyExitBars", 128 );
// สามารถตั้ง early redemption fee% SetOption("EarlyExitFee", 2 );
```

(หมายเหตุ 180 calendar days คือ 128 หรือ 129 trading day)

```
// ใช้คำสั่ง if เพื่อตั้งค่าต่อ 1 สัญลักษณ์
if( Name() == "SYMBOL1" )
{
        SetOption("EarlyExitBars", 128 );
        SetOption("EarlyExitFee", 2 );
        }

                if( Name() == "SYMBOL2" )
        {
        SetOption("EarlyExitBars", 25 );
        SetOption("EarlyExitFee", 1 );
}
```

นอกจากคำสั่ง HoldMinBars, EarlyExitBars ผู้ใช้สามารถใช้คำสั่ง HoldMinDays หรือ EarlyExitDays ที่ใช้วันบน calendar days แทนข้อมูลตาม Bar เราสามารถเขียนโค้ดจากด้านบนโดยใช้ 2 ฟังก์ชั่นได้ดังนี้

```
// ถึงแม้ว่าสัญญาณขายจะเกิดขึ้นทุกวันแต่สัญญาณเหล่านั้นจะถูกเมินจนกว่าจะถือครบ 180 วัน

SetOption(“HoldMinDays”, 180);
Buy = BarIndex() == 0;
Sell = 1;

// คำสั่งสองตัวนี้สามารถตั้งค่าต่อ 1 สัญลักษณ์
// จำนวนวัน หรือ CALENDAR DAYS
// และ early exit (redemption) fee SetOption("EarlyExitDays", 180 );
// early redemption fee (in percent) SetOption("EarlyExitFee", 2 );

(หมายเหตุ 180 วันในปฏิทิน คือ 128 หรือ 129 วันที่ทำการเทรด)

// ใช้คำสั่ง if เพื่อตั้งค่าต่อ 1 สัญลักษณ์
if( Name() == "SYMBOL1" )
{
        SetOption("EarlyExitDays", 180 );
        SetOption("EarlyExitFee", 2 );
        }

                if( Name() == "SYMBOL2" )
        {
```

```
        SetOption("EarlyExitDays", 30 );
        SetOption("EarlyExitFee", 1 );
}
```

การแก้ไขปัญหาการเข้าซื้อพร้อมๆกัน (RESOLVING SAME BAR, SAME SYMBOL SIGNAL CONFLICTS)

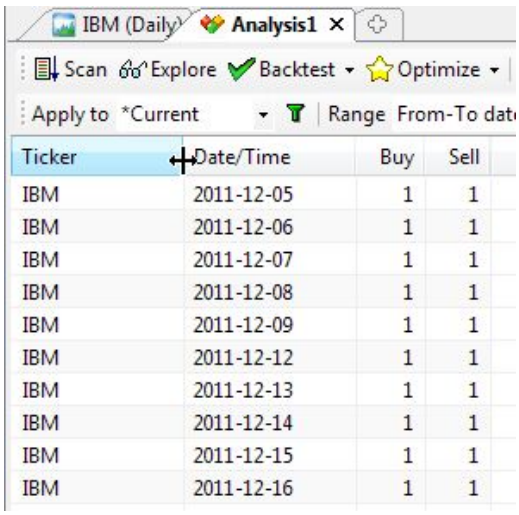
มีความเป็นไปได้ที่จะเกิดสัญญาณเข้าซื้อ และสัญญาณออกใน Bar เดียวกันตัวอย่างเช่น

```
Buy = 1;
Sell = 1;
```

เมื่อคุณใส่คำสั่ง exploration ดังกล่าว:

```
AddColumn(Buy,"Buy", 1.0 ); AddColumn(Sell, "Sell", 1.0 ); Filter = Buy OR Sell;
```

จะพบกับหน้าต่างดังนี้



Ticker	Date/Time	Buy	Sell
IBM	2011-12-05	1	1
IBM	2011-12-06	1	1
IBM	2011-12-07	1	1
IBM	2011-12-08	1	1
IBM	2011-12-09	1	1
IBM	2011-12-12	1	1
IBM	2011-12-13	1	1
IBM	2011-12-14	1	1
IBM	2011-12-15	1	1
IBM	2011-12-16	1	1

เนื่องจากตัว entry และ exit signals จะไม่มีข้อมูลเรื่องเวลาอยู่ในนั้น ดังนั้นจะไม่รู้ว่าสัญญาณไหนเกิดขึ้นก่อน เราสามารถตีความกับความหมายของมันได้ 3 แบบ:

1. only one signal is taken at any bar นั่นหมายความว่าการ Trade เกิดขึ้นใน Bar ที่ 1 และขาย ออกที่ Bar ที่ 2 และตัวถัดมาเข้าซื้อที่ Bar ที่ 3 และออกที่ Bar ที่ 4

2. both signals are used and entry signal precedes exit signal นั่นหมายความว่าการซื้อเกิดที่ Bar 1 และการขายเกิดขึ้นหลังจากการซื้อใน Bar เดียวกัน การซื้อครั้งต่อมาเกิดขึ้นที่ Bar ที่ 2 และขายออกหลังจากนั้นใน Bar ที่ 2 เช่นกัน
3. both signals are used and entry signal comes after exit signal. ในกรณีนี้สัญญาณออกตัวแรกจะถูกเพิกเฉย และการเทรดจะเกิดขึ้นใน Bar ที่มีสัญญาณซื้อ และ Exit จะเกิดขึ้นใน Bar ถัดไป หลังจากนั้นสัญญาณซื้อจะเกิดขึ้นใน Bar ที่เราพึ่ง Exit ออกไป และไป Exit ใน Bar ถัดมา เป็นอย่างนี้ไปเรื่อยไป

เราจำเป็นต้องจัดการกับปัญหาการตีความนี้ บางคนอาจจะคิดว่าเพียงกำหนดราคาซื้อ และ ราคาขายก็เพียงพอแล้ว แต่นั่นไม่ใช่วิธีการแก้ปัญหาทั้งหมด เพราะ Array ของราคาก็ไม่ได้มีข้อมูลเรื่องระยะเวลาเช่นกัน ดังนั้นหากราคาปิดและเปิดเท่ากัน (Doji) ก็ไม่สามารถแยกได้ว่าตัวใดเกิดก่อน แต่อย่างไรก็ตามโดยปกติแล้วคนจะไม่ระบุ Buy price และ Sell price อย่างแน่ชัดมักจะตั้งกันตาม Open + Slippage และขายตอน Close – Slippage

วิธีการแก้ปัญหาการตีความของสัญญาณเข้าออกใน Bar เดียวกันนั้นคือการใช้ AllowSameBarExit option และ HoldMinBars option

กรณี 1. Only one signal per symbol is taken at any bar (1 Bar 1 Signal)

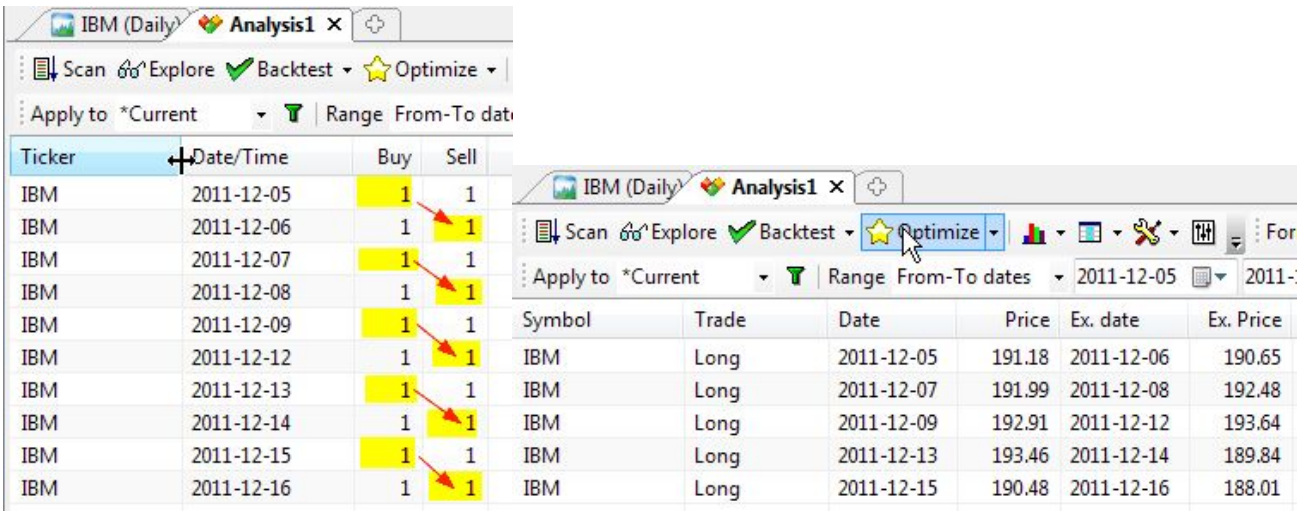
กรณีนี้เกิดขึ้นเมื่อตั้งค่า AllowSameBarExit ให้เป็น False (turned off)

กรณีนี้เราไม่ต้องสนว่าสัญญาณซื้อหรือขายเกิดขึ้นก่อนใน Bar นั้นๆ ตัวอย่างเช่นสมมติเกิด entry signal (ด้วย buy signal taking precedence over short) สัญญาณอื่นๆ ที่เกิดขึ้นหลังจากนั้นจะถูกเพิก

เฉยใน Bar นั้น ถ้าหากเรา Longในกรณีนี้ เราจะรอ Bar ถัดมาและ Sell ในBar นั้น และเคลื่อนไปยัง Bar ถัดไป (เช่นเดียวกันกับกรณี Short-Cover) ถ้าหากไม่มีสัญญาณออก เราก็จะเคลื่อนที่ไปพิจารณา Bar ถัดๆไป

```
SetOption("AllowSameBarExit", False );
Buy = 1;
Sell = 1;
```

ดังภาพดังกล่าว



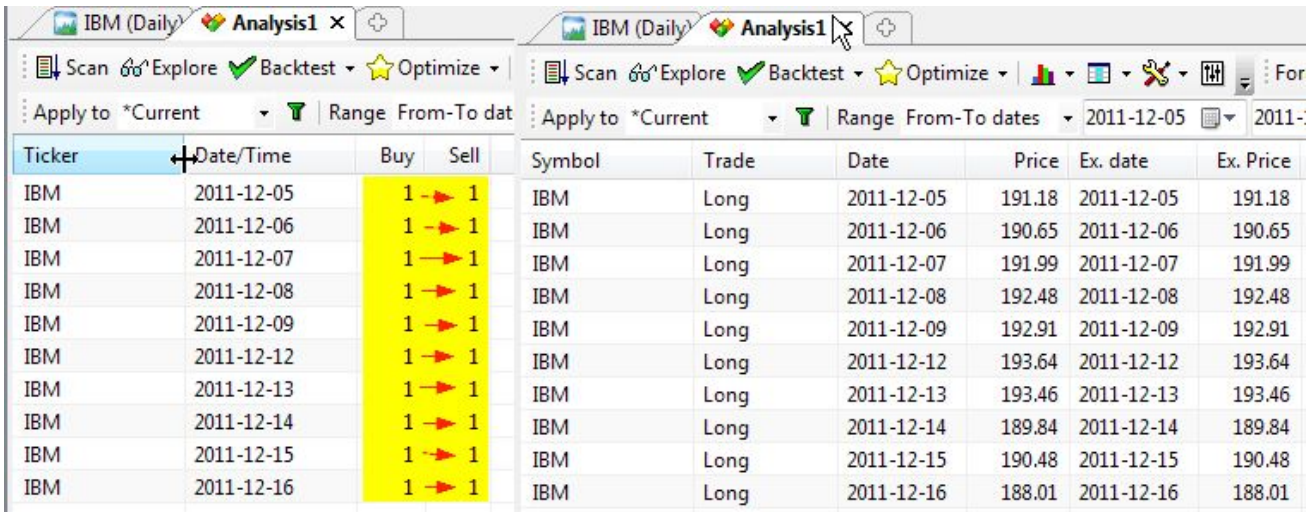
<u>กรณีที่ 2 สัญญาณในการเข้า และ สัญญาณในการออกเกิดขึ้นพร้อมๆกัน (Both entry and exit signals are used and entry signal precedes exit signal)</u>

กรณีนี้นี้จะเกิดขึ้นเมื่อผู้ใช้ตั้งค่า AllowSameBarExit ให้เป็น True (turned on) และตั้ง HoldMinBars ให้เป็นศูนย์ซึ่งก็คือค่า Default นั่นเอง

ในเคสนี้เราจะตอบสนองต่อ signal ทั้งสองอย่างทันทีทันใด เมื่อเจอสัญญาณ entry แล้วระบบจะตรวจหา exit ไว้เช่นกัน เมื่อเราตั้ง Long ไว้เราจะ Sell signal นั้นๆ หากเรา Short มันไว้เราจะ Cover ตัวนั้นๆ เราจะเคลื่อนที่ไปพิจารณา Bar ถัดไปเมื่อทำการกับสัญญาณทุกตัวใน Bar ปัจจุบันเรียบร้อยแล้ว

```
SetOption("AllowSameBarExit", True );
Buy = 1;
Sell = 1;
```

ดังรูปภาพดังกล่าว



IBM (Daily) | Analysis1 ×

Scan | Explore | Backtest | Optimize | Apply to: *Current | Range: From-To dat

Ticker	Date/Time	Buy	Sell
IBM	2011-12-05	1	1
IBM	2011-12-06	1	1
IBM	2011-12-07	1	1
IBM	2011-12-08	1	1
IBM	2011-12-09	1	1
IBM	2011-12-12	1	1
IBM	2011-12-13	1	1
IBM	2011-12-14	1	1
IBM	2011-12-15	1	1
IBM	2011-12-16	1	1

IBM (Daily) | Analysis1

Scan | Explore | Backtest | Optimize | Apply to: *Current | Range: From-To dates | 2011-12-05

Symbol	Trade	Date	Price	Ex. date	Ex. Price
IBM	Long	2011-12-05	191.18	2011-12-05	191.18
IBM	Long	2011-12-06	190.65	2011-12-06	190.65
IBM	Long	2011-12-07	191.99	2011-12-07	191.99
IBM	Long	2011-12-08	192.48	2011-12-08	192.48
IBM	Long	2011-12-09	192.91	2011-12-09	192.91
IBM	Long	2011-12-12	193.64	2011-12-12	193.64
IBM	Long	2011-12-13	193.46	2011-12-13	193.46
IBM	Long	2011-12-14	189.84	2011-12-14	189.84
IBM	Long	2011-12-15	190.48	2011-12-15	190.48
IBM	Long	2011-12-16	188.01	2011-12-16	188.01

<u>กรณีที่ 3. สัญญษณทั้งสองเกิดขึ้นพร้อมๆกัน แต่สัญญษณในการเข้ามาทีหลังสัญญาณในการออก (Both signals are used and entry signal comes after exit signal)</u>

กรณีจะเกิดขึ้นเมื่อผู้ใช้ตั้งค่า AllowSameBarExit เป็น True (turned on) และ HoldMinBars เป็น 1 (หรือมากกว่านั้น)

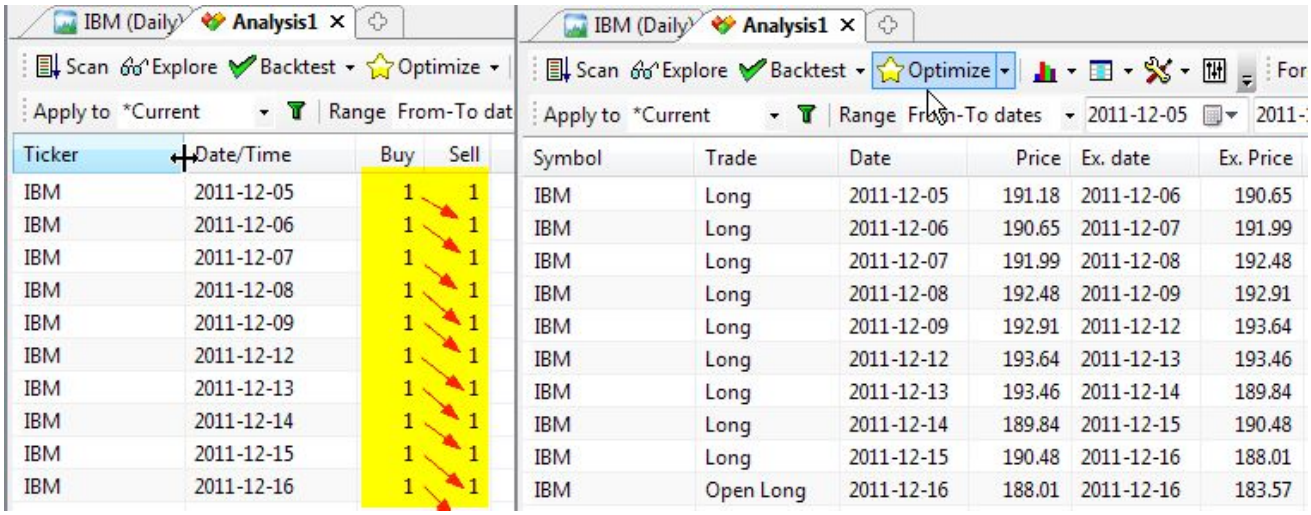
ในเคสนี้เราจะตอบสนองต่อสัญญาณใน Bar เดียวกันแต่ภายใต้ข้อจำกัดของ HoldMinBars = 1 ดังนั้นการเทรดที่เกิดขึ้นจะไม่ได้รับการ Exit ใน Bar เดียวกัน ระบบขะเคลื่อนที่ไปพิจารณา Bar ถัดไป

```
SetOption("AllowSameBarExit", True );
SetOption("HoldMinBars", 1 );
```

Buy = 1;

Sell = 1;

ดังภาพดังกล่าว



ใช้งานในรูปแบบของพอร์ต (How does it work in portfolio case?)

ระบบจะปฏิบัติในลักษณะเดียวกันต่อหุ้นหลายๆตัวโดยที่จะให้น้ำหนักการซื้อขายตาม Position Score ปัญหาของ First same-bar conflicts ในระดับพอร์ตโฟลิโอจะถูกแก้ไขเช่นเดียวกับที่ได้กล่าวไป

กลยุทธ์แบบ market-neutral, long-short balanced strategies (Support for market-neutral, long-short balanced strategies)

กลยุทธ์การลงทุนที่ต้องการจะหลีกเลี่ยง market risk หรือที่เรียกว่า market neutral สามารถทำได้โดยการ hedging ซึ่งสามารถทำได้โดยการเปิด Long และ Short equity positions ในหุ้นที่อยู่ใน Sector เดียวกัน เพื่อที่จะตัดความเสี่ยงใน Sector และ Market ให้มากที่สุด

ในเวอร์ชั่น 5.20 ผู้ใช้สามารถรันระบบ Market Neutral ได้ ด้วยคำสั่งดังกล่าว SeparateLongShortRank, MaxOpenLong, MaxOpenShort

SeparateLongShortRank backtester option

หากต้องการแยกการพิจารณาเกณฑ์การเข้าของ Longและ Short ให้ใส่โค้ดดังกล่าว:

```
SetOption("SeparateLongShortRank", True );
```

เมื่อโหมดนี้ถูก activate ตัว backtester จะพิจารณา score ตาม "top-ranked" signal lists ของ position Long และ Short แยกกันทำให้สกอร์ของแต่ละอันมีเกณฑ์ที่ไม่เหมือนกันได้ ตัวอย่างเช่น หุ้นที่คัดเลือกตามเกณฑ์เข้าซื้อแบบ long ของเรา ในขณะที่ หุ้นที่คัดเลือกตามเกณฑ์เข้าซื้อแบบ short นั้นเป็นค่าติดลบซึ่งขัดกับโหมด default ของเราที่นำค่าของ absolute value ของสกอร์ใดสกอร์หนึ่ง มาพิจารณาอย่างเดียว

หลังจากที่ผู้ใช้เปิดโหมด SeparateLongShortRank แล้ว ตัว Backtester จะทำการจับคู่ตาม สกอร์ของ Long และ Short ตัวที่สกอร์สูงสุดบนกระดาน Long จะจับคู่กับ ตัวที่สกอร์ทอปบนกระดาน Short ตามลำดับลงมาเรื่อยๆ (เมื่อสามารถจับคู่ได้ ในกรณีที่ไม่มีคู่ ระบบจะแสดงผลตามสัญญาณที่เหลือ)

ตัวอย่างเช่น:
Entry signals(score):ESRX=Buy(60.93), GILD=Short(-47.56), CELG=Buy(57.68),
MRVL=Short(-10.75), ADBE=Buy(34.75), VRTX=Buy(15.55), SIRI=Buy(2.79),

จะสังเกตได้ว่า สัญญาณ short นั้น ปะปนอยู่กับ สัญญาณ Long ในขณะที่ค่า Absolute Value ของสกอร์นั้นมีจำนวนน้อยกว่า Long Signal ในบาร์นั้นมีสัญญาณ Short แค่ 2 ตัว ตัวที่เหลือจะเป้นสัญญาณ Long ตาม Position Score ฟังก์ชันนี้มักนิยมใช้กับ MaxOpenLong และ MaxOpenShort options

MaxOpenLong / MaxOpenShort backtester options

MaxOpenLong - จำกัดจำนวน Position LONG ที่สามารถเปิดได้
MaxOpenShort - จำกัดจำนวน Position SHORT ที่สามารถเปิดได้

ตัวอย่างเช่น:

```
SetOption("MaxOpenPositions", 15 );
SetOption("MaxOpenLong", 11 );
SetOption("MaxOpenShort", 7 );
```

หากใส่เลขศูนย์ต่อท้ายหมายความว่าผู้ใช้จะไม่จำกัดจำนวน Position LONG และ SHORT ตัว Backtester จะขึ้นตรงต่อ (MaxOpenPositions) ไม่ว่าจะเปิด Position ใดก็ตาม

หมายเหตุ คำสั่งของ MaxOpenLong และ MaxOpenShort รวมกันนั้นสามารถระบุจำนวนที่เท่าหรือ ไม่เท่ากับ MaxOpenPositions

เมื่อจำนวนของ MaxOpenLong + MaxOpenShort มากกว่า MaxOpenPositions นั้น จำนวนการถือ Position ทั้งหมดจะไม่เกิน MaxOpenPositions สัดส่วนการถือจะขึ้นอยู่กับ long/short limits ที่ผู้ใช้ได้ตั้งเอาไว้ ตัวอย่างเช่น หากระบบของท่านตั้งค่าไว้ว่า MaxOpenLong จะถือ 7 position และ maxOpenShort จะถือ 7 position เช่นกันนั้น แต่ตั้งค่า MaxOpenPositions ไว้ที่ 10 หากมีสัญญาณเกิดขึ้นทั้งหมด 20 ตัว: 9 long (highest ranked) และ 11 short ระบบจะทำการเปิด Long 7 position และเปิด Short ไว้ 3 position

เมื่อ MaxOpenLong + MaxOpenShort นั้นน้อยกว่า MaxOpenPositions แต่มากกว่าศูนย์ระบบจะไม่เปิด Position เกินไปกว่า MaxOpenLong +MaxOpenShort

ฟังก์ชัน MaxOpenLong และ MaxOpenShort นั้นไม่มีผลกระทบต่อ Ranking นั่นคือ Position Score จะยังคง เลือกค่า Absolute ที่สูงที่สุดตามค่า Default

หาก position score ของผู้ใช้นั้นไม่ symmetrical นั่นหมายความว่าผู้ใช้อาจจะไม่ได้ top-ranked signals จากฝั่งใดฝั่งหนึ่งตามต้องการ ดังนั้นคำแนะนำคือให้ใช้คำสั่ง MaxOpenLong และ MaxOpenShort (หากดำเนินกลยุทธ์"market neutral") ควบคู่ไปกับ separate long/short ranking ดังคำสั่งนี้

SetOption("SeparateLongShortRank", True);

ดูเพิ่มเติม:

Backtesting your trading ideas article.

Backtesting systems for futures contracts article.

Using AFL editor section of the guide.

Insider guide to backtester (newsletter 1/2002)

การอ่านผลการ Backtest

(Reading backtest report)

หากต้องการจะดูค่าของการ Backtest ให้เลือกที่ปุ่ม Report ในหน้า automatic analysis window หากต้องการจะดูค่าของ Backtest ทั้งหมดที่ผ่านมาให้คลิกที่ปุ่มลูกศรชี้ลงตรงคำสั่ง Report และเลือก Report Explorer หากผู้ใช้คลิกข้อมูลในแต่ละบรรทัด จะมีหน้าต่าง Report โชว์ขึ้นมา

Report ในเวอร์ชั่นใหม่นี้มีข้อมูลให้มากกว่าเดิม แยกสถิติของ Position Long และ Short รายละเอียดของ Metric ต่างๆ คร่าวๆดังนี้

Exposure %

'Market exposure ของ trading system นั้นคำนวณจากแต่ละ Bar นำผลรวมของ Exposure แล้วหารด้วยจำนวน Bar ทั้งหมด Single bar exposure คือ ค่าของ open positions หารด้วย portfolio equity ของ Bar นั้นเอง

Net Risk Adjusted Return %

Net profit % หารด้วย Exposure %

Annual Return %

Compounded Annual Return % (CAR) ผลตอบแทนทบต้น

Risk Adjusted Return %

Annual return % divided by Exposure %

Avg. Profit/Loss

หรือที่รู้จัดกันในชื่อ Expectancy ($) - (Profit of winners + Loss of losers)/(number of trades), บ่งบอกถึงจำนวนเงินที่น่าจะ ได้/เสีย ต่อการเทรด 1 ครั้ง

Avg. Profit/Loss %

หรือที่รู้จัดกันในชื่อ Expectancy (%) - '(% Profit of winners + % Loss of losers)/(number of trades) บ่งบอกถึงเปอร์เซ็นต์ ได้/เสีย ต่อการเทรด 1 ครั้ง

Avg. Bars Held

sum of bars in trades / number of trades จำนวน Bar เฉลี่ยที่ถือไว้ต่อการเทรด 1 ครั้ง

Max. trade drawdown

ความห่างระหว่างจุดสูงสุด peak และจุดต่ำสุด valley ของการเทรด 1 ครั้ง ยิ่งต่ำยิ่งดี

Max. trade % drawdown

Percentage ของความห่างระหว่างจุดสูงสุด peak และจุดต่ำสุด valley ของการเทรด 1 ครั้ง ยิ่งต่ำยิ่งดี

Max. system drawdown

ความห่างระหว่างจุดสูงสุด peak และจุดต่ำสุด valley ของทั้งพอร์ต ยิ่งต่ำยิ่งดี

Max. system % drawdown

Percentage ของความห่างระหว่างจุดสูงสุด peak และจุดต่ำสุด valley ของทั้งพอร์ต ยิ่งต่ำยิ่งดี

Recovery Factor

Net profit หารด้วย Max. system drawdown

CAR/MaxDD

Compound Annual % Return หารด้วย Max. system % drawdown หากเกิน 2 ถือว่าดี

RAR/MaxDD

Risk Adjusted Return หารด้วย Max. system % drawdown หากเกิน 2 ถือว่าดี

Profit Factor

Profit of winners หารด้วย loss of losers

Payoff Ratio

อัตราส่วนของ average win / average loss

Standard Error

Standard error วัดความไม่แน่นอน/ขรุขระ ของเส้น equity line ยิ่งต่ำยิ่งดี

Risk-Reward Ratio

วัดความสัมพันธ์ระหว่าง ความเสี่ยงและ potential gain ของระบบการเทรด ยิ่งสูงยิ่งดี โดยคำนวณจากการหา slope ของ equity line (expected annual return) ด้วย standard error

Ulcer Index

Square root ของ sum of squared drawdowns หารด้วย number of bars

Ulcer Performance Index

(Annual profit - Tresury notes profit)/Ulcer Index'>Ulcer Performance Index. ปัจจุบัน tresury notes profit ถูกเซ็ทไว้ที่ 5.4.ในอนาคตผู้ใช้จะสามารถกำหนดเองได้

Sharpe Ratio of trades

คำนวณหา risk adjusted return ของ investment มากกว่า 1 ถือว่าดี หากเกิน 2 ถือว่าดีมากๆ รายละเอียดเพิ่มเติมศึกษาได้ที่ http://www.stanford.edu/~wfsharpe/art/sr/sr.htm การคำนวณ: หาค่าเฉลี่ยของ Return และ Standarn Deviation ออกมาก่อน หลังจากนั้นคูณด้วย (NumberOfBarsPerYear) / (AvgNumberOfBarsPerTrade) ซึ่งเป็นอัตราส่วนทีทำให้ค่าอยู่ในหลัก ต่อปี (Annualized) จากนั้นนำ Risk Free Rate ซึ่งถูกตั้งไว้ที่ 5 ไปลบจาก annualized average return และนำก้อนนั้นไปหารด้วย annualized SD of return

K-Ratio

ตรวจหาความไม่ต่อเนื่องของผลตอบแทน ควรจะเท่ากับ 1 หรือมากกว่านั้น คำนวณจากการหา Linear regression slope ของ equity line คูณด้วย square root ของ sum of squared deviations ของ bar number หารด้วย standard error ของ equity line และนำไปคุณด้วย square root of number of bars รายละเอียดเพิ่มเติมดูได้ที่ Stocks & Commodities V14:3 (115-118): Measuring System Performance by Lars N. Kestner

การใส่สีเข้าไปในผลการ backtest (ใหม่ในเวอร์ชั่น 5.60) (Color-coding in the backtest report)

เวอร์ชั่น 5.60 สามารถให้ผู้ใช้ใส่สีระบุค่าที่ 'good' และ 'bad' บน backtest report ได้ค่าบางค่านั้นจะถูกใส่สีไว้อยู่แล้ว โดยสี น้ำเงิน หมายถึง “ปานกลาง” เขียวหมายถึง “ดี” และแดงหมายถึง “แย่” ตัวอื่นๆจะเป็นสีดำ

การใส่สีบนค่ารีพอร์ตควรจะใส่อย่างมีจุดประสงค์ โดยทั่วไปมักจะใช้สีแดงเพื่อบ่งบอกถึงการเตือน และหมั่นควรเช็คค่าของรีพอร์ตเช่นกันไม่ใช่แค่สี

metrics เหล่านี้ถูกใส่สีไว้ตามคุณภาพของค่าดังต่อไปนี้

Net Profit, Net Profit % - bad < 0, good > 0

Annual Profit %, bad < 0, neutral betwen 0 and 10, good > 10 RAR % bad < 0, good > (10 / Exposure)

Avg. Profit/Loss all trades (Expectancy $) - bad < 0, good > 0 Avg Profit/Loss % all trades (Expectancy %) - bad < 0, good > 0

Max. system % drawdown - bad: dd worse than -30%, neutral: dd between -30 and -10%, good - -10% to 0% CAR/MaxDD, RAR/MaxDD - bad < 1, neutral between 1 and 2, good > 2

Recovery factor - bad < 1, neutral between 1 and 2, good > 2 Payoff ratio - bad < 1, neutral between 1 and 2, good > 2

เพิ่มเติม :

Old backtest report

Backtesting your trading ideas article. Portfolio Backtesting article.
Backtesting systems for futures contracts article. Using AFL editor section of the guide
Insider guide to backtester (newsletter 1/2002)

วิธีการ Optimize

(How to optimize trading system)

หมายเหตุ: เนื้อหามีความซับซ้อน ควรจะอ่านเนื้อหาก่อนหน้านี้

เกริ่นนำ (Introduction)

คอนเซปของการ Optimize นั้นเรียบง่าย ผู้ใช้ต้อมีระบบการลงทุนก่อน อาจจะเป็นระบบ Simple moving average Crossover หรืออะไรก็ตาม ในทุกๆระบบ จะมี Parameter ที่กำหนดว่าตัวระบบจะทำงานอย่างไร (ในกรณีนี้ตัวอย่างเช่นค่าของ averaging period สำหรับการ crossover) การ Optimize คือการหาค่า Parameter ที่เหมาะสมและดีที่สุดต่อระบบ เพื่อให้ได้ผลตอบแทนที่ดีที่สุดต่อ หุ้นตัวเดี่ยวๆ หรือต่อทั้งพอร์ตนั่นเอง Amibroker เป็นเพียงไม่กี่โปรแกรมที่สามารถ optimize หุ้นหลายๆตัวได้

การ Optimize นั้นผู้ใช้จะต้องกำหนดค่า Parameter ตั้งแต่ 1 ค่าไปจนถึง 10 ค่า และกำหนดค่าสูงสุด และ ต่ำสุดของแต่ละ Parameter หลังจากนั้น Amibroker จะทำการ Backtest หลายๆครั้ง โดยหยิบ parameter เหล่านั้นมาทำการ Backtest (การจับคู่ที่เป็นไปได้ทั้งหมด) เมื่อจบการ Backtest หลายๆ ครั้งนี้แล้ว Amibroker จะแสดงผลการ Backtest ทั้งหมดเรียงตาม Profit คุณสามารถเลือกค่า Parameter ตามต้องการได้

โค้ด AFL formula (Writing AFL formula)

การเขียนฟังก์ชัน Optimization นั้นมีลักษณะดังนี้

```
variable = optimize( "Description", default, min, max, step );
```

โดยที่: variable - คือตัวแปรที่เก็บข้อมูลของการ Optimize ซึ่งสามารถตั้งขึ้นโดยผู้ใช้เอง
(ค่าที่ออกมาจะเป็นค่า default variable = default;)

ในโหมด optimization จะให้ค่าตามการปรับ Parameter จากค่า Min ที่ตั้งไว้ไปจนถึงค่า Max เป็น Step เรื่อยๆไป

"Description" คือ ชื่อของตัวแปร Optimize ซึ่งชื่อนี้จะอยู่บนคอลัมน์ optimization result list

default คือ ค่า Default ที่ระบบจะเลือกใช้ในโหมด exploration, indicator, commentary, scan และ normal back test modes

min คือค่าต่ำสุดของ Parameter ที่อยาก Optimize

max คือค่าสูงสุดของ Parameter ที่อยาก Optimize

step คือจำนวนการเพิ่มขึ้นของ Parameter จาก Min ไป Max

หมายเหตุ:

- AmiBroker รับตัวแปร Optimize Variable ได้แค่ 64 ตัวเท่านั้น หากผู้ใช้ใช้ exhaustive optimization ควรจะจำกัดจำนวนของการ optimization
- การ Optimize แต่ละครั้งจะให้จำนวนผลลัพธ์ออกมาเป็น จำนวน Step ของการ Optimize คูณด้วย จำนวน Step ของการ Optimize ของตัวแปรถัดไป ตัวอย่างเช่น หากต้องการ Optimize Parameter 2 ตัว ตัวละ 10 Step จะได้ค่าผลลัพธืออกมาทั้งหมด 10*10 = 100 ค่า (Loop Backtest 100 ครั้ง)
- เรียกใช้ optimizae function ต่อหนึ่งตัวแปรด้านบนของโค้ดของคุณ เนื่องจากการเรียก optimizae function แต่ละครั้งจะทำให้เกิด optimization loop อันใหม่

- Amibroker รองรับ Multiple-symbol optimization
- Maximum search space คือ 2^{64} (10^{19} = 10,000,000,000,000,000,000) combinations

ตัวอย่างเช่น

1. การ Optimize 1 ตัวแปร:

```
sigavg = Optimize( "Signal average", 9, 2, 20, 1 );

Buy = Cross( MACD( 12, 26 ), Signal( 12, 26, sigavg ) );
Sell = Cross( Signal( 12, 26, sigavg ), MACD( 12, 26 ) );
```

2. การ Optimize 2 ตัวแปร (suitable for 3D charting)

```
per = Optimize("per", 2, 5, 50, 1 );
Level = Optimize("level", 2, 2, 150, 4 );

Buy=Cross( CCI(per), -Level );
Sell = Cross( Level, CCI(per) );
```

3. การ Optimize 3 ตัวแปร:

```
mfast = Optimize( "MACD Fast", 12, 8, 16, 1 );
mslow = Optimize("MACD Slow", 26, 17, 30, 1 );
sigavg = Optimize( "Signal average", 9, 2, 20, 1 );

Buy = Cross( MACD( mfast, mslow ) , Signal( mfast, mslow, sigavg ) );
Sell = Cross( Signal( mfast, mslow, sigavg ), MACD( mfast, mslow ) );
```

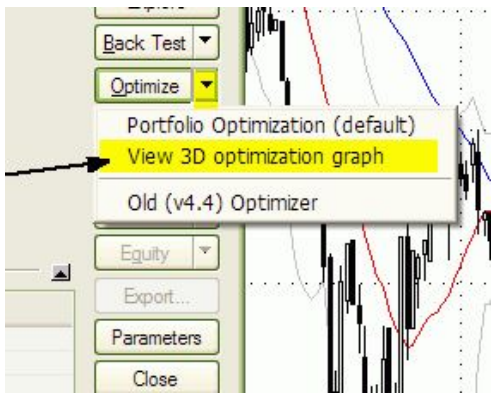
หลังจากใส่ formula ให้กดปุ่ม Optimize ในหน้าต่าง "Automatic Analysis" AmiBroker จะเริ่มทำการทดสอบ และจับคู่ค่า Parameter ต่างๆและแสดงผลออกมาใน Report ค่าที่ออกมาจะเรียงตาม Net % profit ผู้ใช้สามารถเรียงค่าตาม lowest drawdown, lowest number of trades, largest profit factor, lowest market exposure และ highest risk adjusted annual % return คอลัมน์ท้ายๆ จะบ่งบอกถึงค่าที่ผู้ใช้ Optimize

เมื่อผู้ใช้พบกับ Parameter ที่เหมาะสมที่สุดแล้ว ให้ใส่ค่านั้นลงไปในค่า Default ใน Optimize function (พารามิเตอร์ที่สองของฟังก์ชัน optimize

แสดงผล optimization แบบสามมิติ (Displaying 3D animated optimization charts)

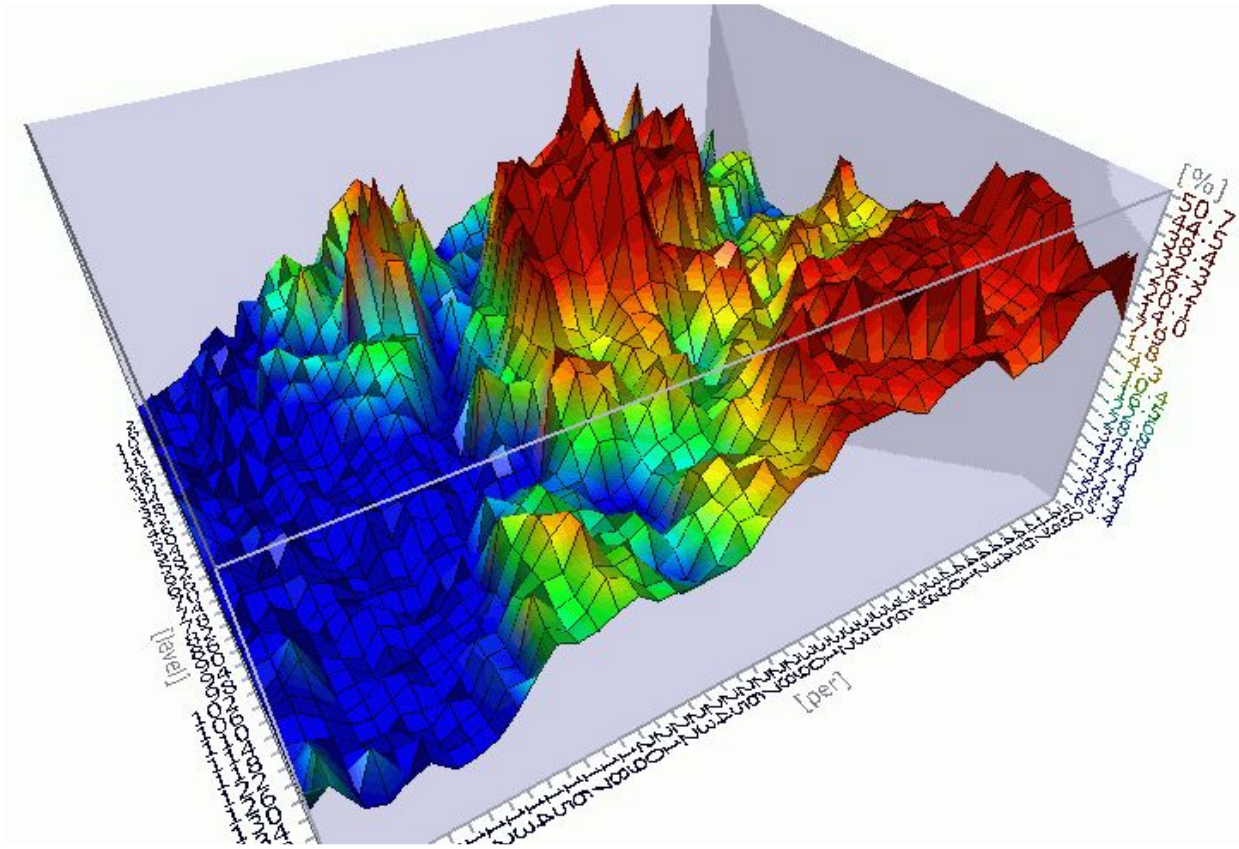
ผู้ใช้สามารถเรียกรูปกราฟสามมิติ ออกมาได้หากทำการ Optimize 2 ตัวแปร (two-variable optimization) ซึ่งต้องมีการเปิดฟังก์ชัน Optimize() 2 รอบ ตัวอย่างเช่น:

```
per = Optimize("per", 2, 5, 50, 1 );
Level = Optimize("level", 2, 2, 150, 4 );

Buy=Cross( CCI(per), -Level );
Sell = Cross( Level, CCI(per) );
```



หลังจากใส่โค้ดแล้วให้กดปุ่ม Optimize

ผู้ใช้สามารถเรียกกราฟสามมิติออกมา เพียงคลิกที่ปุ่มลูกศรเลื่อนลงตรงปุ่ม Optimize และ เรียก View 3D optimization graph เพียงเสี้ยววินาทีผู้ใช้จะพบกับกราฟ 3 มิติ ดังตัวอย่างด้านล่าง



โดยทั่วไปกราฟสามมิติจะแสดงผลในรูปของ แกน Net profit และแกน optimization variables ซึ่งผู้ใช้สามารถเลือกที่จะเปลี่ยนแกนได้ และเลือกแกนจากคอลัมน์ของ optimization result table เพียงเลือกหัวคอลัมน์ที่ต้องการ (เช่น Max. System Drawdown, CAR/Max. DD ...) เลือกเป็นแกน และคลิกให้ result เรียงตามคอลัมน์ที่ต้องการ หลังจากนั้นให้กด View 3D optimization graph อีกรอบ

ผู้ใช้สามารถใช้กราฟสามมิติในการตัดสินใจที่จะเลือกใช้ Parameter ให้เหมาะกับ Trading System ซึ่ง Parameter ที่ดีนั้นจะโชว์การเคลื่อนขึ้นของกราฟอย่างช้าๆ ไม่ใช่ขึ้นมาโดดๆ กราฟสามมิติยังเป็นเครื่องมือในการป้องกัน curve-fitting Curve-fitting (หรือ over-optimization) เกิดขึ้นเมื่อระบบใส่ตัวแปรที่เอื้ออำนวยต่อสถานการณ์ใดสถานการณ์หนึ่ง ซึ่งอาจจะไม่เกิดขึ้นในอนาคต จุดที่โดดขึ้นมาจากกราฟ สามมิติอย่างผิดปกตินั้น บ่งบอกถึงความ over-optimization ผู้ใช้ควรจะเลือก Parameter ในลักษณะที่มี พื้นที่สูงและกว้าง ส่วนตัวที่โดดออกมาเดี่ยวๆนั้นไม่เหมาะกับการใช้ในระบบ

การควบคุมกราฟสามมิติ (3D chart viewer controls)

AmiBroker's 3D chart ให้ผู้ใช้สามรถดูกราฟได้หลากหลายมุมมอง ตามคำสั่งดังกล่าว

Mouse controls: การใช้เมาส์เพื่อเลือกปรับเปลี่ยนมุมมอง

- การ Rotate - คลิกเม้าส์ซ้ายและเลื่อนไปตามแกน X/Y
- การ Zoom-in, zoom-out - คลิกเม้าส์ขวาและเลื่อนไปตามแกน X/Y
- การ Move (translate) - คลิกเม้าส์ซ้ายพร้อมกับการกด CTRL และเลื่อนไปตามแกน X/Y
- การ Animate - คลิกเม้าส์ซ้าย และเลื่อนเมาส์อย่างรวดเร็วและปล่อยเมาส์ออก

Keyboard controls: การใช้คีย์บอร์ดปรับเปลี่ยนมุมมอง

SPACE - ทำการ auto-rotate
LEFT ARROW KEY - rotate ไปทางซ้าย
RIGHT ARROW KEY - rotate ไปทางขวา
UP ARROW KEY - rotate ขึ้นด้านบน
DOWN ARROW KEY - rotate ลงด้านล่าง
NUMPAD + (PLUS) - Near (zoom in)
NUMPAD - (MINUS) - Far (zoom out)
NUMPAD 4 - เคลื่อนที่ไปทางซ้าย
NUMPAD 6 - เคลื่อนที่ไปทางขวา
NUMPAD 8 - เคลื่อนที่ไปด้านบน
NUMPAD 2 - เคลื่อนที่ลงด้านล่าง
PAGE UP - water level up
PAGE DOWN - water level down

โหมด Smart (non-exhaustive) optimization (Smart (non-exhaustive) optimization)

เกริ่นนำ (Introduction)

AmiBroker มีโหมด smart (non-exhaustive) optimization นอกเหนือจากแบบธรรมดา exhaustive search

โหมด Non-exhaustive search จะมีประโยชน์เมื่อการรวมกันของ Parameter ในระบบนั้นมีขนาดที่ใหญ่เกินกว่าโหมด exhaustive search จะประมวลได้

Exhaustive search หรือโหมดธรรมดานั้นเหมาะกับตัวแปรที่ไม่เยอะมาก สมมติว่า ผู้ใช้ต้องการจะ Optimize 2 Parameters แต่ละตัวไล่เสต็ปจาก 1 ไป 100 รวมกันก็เท่ากับ 10,000 combinations (10^2) ถ้าหากเพิ่มอีก 1 parameter เข้าไป ค่าที่ออกมาจะได้ 1,000,000 combination โหมด Exhaustive Search อาจจะยังประมวลผลได้ แต่ถ้าหากคุณเพิ่มเข้าไป 4 Parameter เพิ่มไป 5 Parameter คุณจะมีถึง 10,000 ล้าน combination ไปถึงจุดนั้นมันเป็นการยากมากที่จะต้องมาไล่ดูทีละตัว โหมด non-exhaustive จะเข้ามาแก้ปัญหานี้ได้

Quick Start

นี่คือวิธีการใช้ new non-exhaustive optimizer ที่เรียบง่ายที่สุด (ในกรณีนี้CMA-ES).

1. เปิด Formula จาก Formula Editor

2. เพิ่มโค้ดนี้ไว้บนสุดของ formula:
   OptimizerSetEngine("cmae"); // คุณสามารถใช้คำสั่ง"spso" หรือ "trib"
3. (เลือกได้ว่าจะทำขั้นตอนนี้หรือไม่) เลือกค่า Target ที่อยากจะ Optimize โดยปดติแล้วค่า Default จะ Optimize ตัว CAR/MDD เลือกผ่าน Automatic Analysis, Settings, "Walk-Forward" tab, และ Optimization target เพียงแค่นั้น

หากผู้ใช้รัน optimization โดยใช้โค้ดนี้ จะพบกับการใช้ (non-exhaustive) CMA-ES optimizer

ขั้นตอนเป็นอย่างไร (How does it work ?)

จุดประสงค์ของ optimization จะเป็นเพื่อการหาจุดต่ำสุด หรือ จุดสูงสุด ของ Function ระบบการลงทุนสามารถมองในรูป Function ได้โดย Input ที่ใส่เข้าไปคือ Parameters และ Quotation Data ส่วน output ที่ออกมาคือ ค่า Optimization Target (ซึ่งในกรณีนี้คือ CAR/MDD) และ ผู้ใช้ต้องการจะหาค่าสูงสุดของมัน

smart optimization algorithms บางรูปแบบ ใช้คอนเซปของ ธรรมชาติและวิวัฒนาการของสัตว์ - PSO algorithm, หรือ biological process - Genetic algorithms, และบางรูปแบบก้ใช้หลักคณิตศาสตร์ที่คิดค้นโดยมนุษย์ - CMA-ES.

algorithms เหล่านี้พบอยู่บ่อยๆในหลากหลายสาขาโดยเฉพาะสาขา Finance สามารถค้นหา "PSO finance" หรือ "CMA-ES finance" ใน Google แล้วคุณจะพบกับหลายๆคำอธิบายและบทความที่เกี่ยวข้อง

Non-exhaustive (หรือ "smart") methods จะเฟ้นหา global หรือ local optimum จุดสูงสุด เป้าหมายคือการหาจุดสูงสุดของทุก Parameter แต่ไม่ใช่จุดที่โดดออกมาจากพื้นที่รอบข้าง (Curve-Fitting) ระบบจะเฟ้นหา พื้นที่ที่มีระนาบสูงสุดแทนนั่นเอง

การค้นหาโดย algorithm ของ non-exhaustive จะมีลักษณะดังนี้:

a) optimizer จะสร้าง starting population ของ parameter set แบบสุ่ม
b) ระบบจะทำการ Backtest parameter set แต่ละตัวจาก กลุ่ม population ที่สร้างขึ้นมา
c) ผลของ Backtest จะถูกประเมิน ถ้าหากยังไม่พบค่าก็จะสุ่มสร้าง Population Set ใหม่ขึ้นมา
d) หากระบบค้นพบจุดสูงสุด จะหยุดการค้นหา หรือ จนกว่าจะตรงกับ stop criteria ที่วางไว้

ตัวอย่างของ stop criteria เช่น:

a) เมื่อระบบรันระบบใหม่ซ้ำแล้วซ้ำเล่าไปจนถึงจุดๆหนึ่ง (maximum iterations)
b) หยุดเมื่อค่าที่ดีที่สุดของ X generations ที่ผ่านมาเป็น 0
c) หยุดเมื่อถ้าใส่ 0.1standard deviation vector เข้าไปในแกนแล้วพบว่าค่าของ objective value ที่ดีที่สุดไม่เปลี่ยนแปลง
d) อื่นๆ

หากต้องการจะใช้ smart (non-exhaustive) optimize ในรูปแบบอื่นๆให้ใส่โค้ดโดยเริ่มต้นจาก OptimizerSetEngine function

OptimizerSetEngine("name");

Amibroker มี 3 engines ให้เลือก: Standard Particle Swarm Optimizer ("spso"), Tribes ("trib"), and CMA-ES ("cmae") - ชื่อในวงเล็บให้นำไปใส่ใน (“name”)

ผู้ใช้อาจจะต้องการ Set ค่า internal parameters ในแต่ละ Engine ให้ใช้ OptimizerSetOption function

OptimizerSetOption("name", value) function

ฟังก์ชันนี้จะสร้าง additional parameters สำหรับ optimization engine ขึ้นมา ซึ่ง paramtere เหล่านี้จะขึ้นอยู่กับ engine ที่ใช้

Engine ทั้ง 3 ตัวที่ AmiBroker (SPSO, Trib, CMAE) รองรับ สามารถเลือก Parameter ได้แค่ 2 ตัวคือ: "Runs" (number of runs) และ "MaxEval" (maximum evaluations (tests)per single run) ลักษณะของ parameter ทั้ง 2 จะขึ้นอยู่กับ engine ที่ใช้ ดังนั้นค่าที่ได้ออกมาจาก engine ที่แตกต่างกันจะไม่เหมือนกัน

ความแตกต่างของ Runs และ MaxEval ดังนี้ Evaluation (หรือ test) เปรียบเสมือน single backtest (ไว้ประเมินค่าของ objective function value) RUN คือ การไล่หาจุด optimum value ตาม algorithm ซึ่งมักจะใช้หลายๆ evaluations ในการช่วยหา

การ run แต่ละครั้งจะ RESTARTS กระบวนการ optimization ตั้งแต่แรกทุกครั้ง (new initial random population) ดังนั้นถ้าระบบยังไม่เจอค่าที่สูงที่สุดของ Population (Global Optimum) การรันอาจจะเจอจุดสูงสุดหรือต่ำสุดหลายๆจุดจากทุกๆการรันที่แตกต่างกัน ดังนั้น Run จะเป็นตัวกำหนดจำนวนการเริ่มค้นหาจุด Optimal ในขณะที่ MaxEval คือจำนวนการประเมินผลหรือ Backtest ในแต่ละ Run เพื่อหาค่าที่สูงที่สุด ถ้าหากจุด optimal นั้นหาได้ง่ายโดยอาจจะใช้เพียง1000 test (Evaluations) การรัน 5x1000 ก็เหมาะสมในการหาจุด Global Optimum เพราะระบบจะมีโอกาสน้อยที่จะแสดงค่า Local Optimum เนื่องจากการรันแต่ละครั้งจะเริ่มจาก Random Population ที่แต่กต่างกัน

การเลือก 2 parameter นี้ขึ้นอยู่กับ ปัญหาที่เจอและความซับซ้อนของระบบ การใช้ stochastic non-exhaustive method ไม่สามารถรับประกันได้เสมอไปว่าจะสามารถหาจุด global optimum ได้ คำแนะนำสำหรับการตั้งค่า Parameter คือ ตั้งค่าให้กว้างที่สุด และ คำนึงถึงระยะเวลาในการคำนวณให้เหมาะสมว่าคุณต้องการใช้ข้อมูลในตอนไหน คำแนะนำอีกอย่างคือการ คูณ 10 ด้วยจำนวน Test ที่ใส่มิติใหม่ๆเข้าไปนี่อาจจะเป็นการเพิ่มจำนวนการ test เกินความจำเป็น แต่เพื่อความแม่นยำ สิ่งสำคัญคือ smart optimization methods นั้นจะใช้งานได้อย่างมีประสิทธิภาพเมื่อ parameter นั้นเป็น continuous parameter spaces และ objective functions เองมีความราบรื่นในการหาข้อมูล ถ้าหาก parameter space เป้นแบบ discrete แล้ว evolutionary algorithms เหล่านี้อาจจะมีปัญหาในการหา optimum value ซึ่งตัว binary (on/off) parameters นี้ไม่เหมาะสมต่อการใช้ smart method เหล่านี้ ทางที่ดีหากมี parameter เป็น binary ควร Optimize แต่ continuous parameters โดยใช้ smart optimizer และเปลี่ยน binary parameters

SPSO - Standard Particle Swarm Optimizer

Standard Particle Swarm Optimizer นั้นมาจาก SPSO2007 code บนสมมติฐานที่ว่า เราจะสามารถหาผลลัพธ์ที่ดีได้หากเลือก parameters (เช่น Runs, MaxEval) ที่ถูกต้องต่อแต่ละปัญหา

การหาค่า Parameter เหล่านี้เป็นเรื่องที่ซับซ้อนและจะเปลี่ยนตามสถานการณ์

SPSO.dll จะมาพร้อมกับ full source codes ในไฟล์ย่อย"ADK" ตัวอย่างโค้ดของStandard Particle Swarm Optimizer : (ต้องการหา optimum value จาก 1000 test ค้นจาก search space 10 000 combinations)

```
OptimizerSetEngine("spso");
OptimizerSetOption("Runs", 1 );
OptimizerSetOption("MaxEval", 1000 );

sl = Optimize("s", 26, 1, 100, 1 );
fa = Optimize("f", 12, 1, 100, 1 );

Buy = Cross( MACD( fa, sl ), 0 );
Sell = Cross( 0, MACD( fa, sl ) );
```

TRIBES - Adaptive Parameter-less Particle Swarm Optimizer

Tribes เป็นอีกเวอร์ชั่นของ PSO (particle swarm optimization) non-exhaustive optimize ซึ่งจะมีลักษณะที่เป็น Adaptive และ Parameter-less adaptive รายละเอียดทางด้านวิทยาศาสตร์หาดูได้ที่: http://www.particleswarm.info/Tribes_2006_Cooren.pdf

ในเชิงทฤษฎีแล้วควรจะทำงานดีกว่า regular PSO เพราะมันสามารถ ปรับเปลี่ยน swarm sizes และ algorithm strategy ต่อปัญหาที่มันกำลังแก้อยู่นั่นเอง

ผลออกมาพบว่าทำงานได้ให้ผลคล้ายคลึงกับ PSO

สำหรับ Tribes.DLL plugin จะใช้งาน "Tribes-D" (i.e. dimensionless) variant.อ้างอิงจาก http://clerc.maurice.free.fr/pso/Tribes/TRIBES-D.zip โดย Maurice Clerc

Tribes.DLL มีโค้ดเต็มตัวอยู่ใน “ADK” Folder

Supported parameters : "MaxEval" - maximum number of evaluations (backtests) per run (default = 1000).

```
OptimizerSetOption("MaxEval", 1000 );
```

ผู้ใช้ควรจะเพิ่มจำนวน evaluations ตาม Dimension ที่เพิ่ม (จำนวน Parameters ที่อยาก Optimize)

ค่า default 1000 เหมาะสำหรับ 2 - 3 dimensions

"Runs" - number of runs (restarts). (default = 5)
คุณสามารถทิ้งค่านี้ไว้ 5
โดยทั่วไปแล้วค่านี้จะถูกตั้งไว้ที่ 5

หากต้องการใช้ Tribes optimizer ให้ใส่โค้ดดังกล่าว

```
OptimizerSetEngine("trib");
OptimizerSetOption("MaxEval", 5000 ); // 5000 evaluations max
```

CMA-ES - Covariance Matrix Adaptation Evolutionary Strategy optimizer

CMA-ES (Covariance Matrix Adaptation Evolutionary Strategy) เป็น engine ที่ซับซ้อนขึ้นไปอีก สำหรับข้อมูลเชิงวิทยาศาสตร์ scientific background see:

http://www.bionik.tu-berlin.de/user/niko/cmaesintro.html
พบว่าวิธีนี้ให้ผลดีกว่า อีก 9 วิธีที่มีความโด่งดัง (like PSO, Genetic and Differential evolution).
http://www.bionik.tu-berlin.de/user/niko/cec2005.html

CMAE.DLL ใช้ "Global" variant ของการ search และ restarts หลายๆครั้งโดยแต่ละครั้งเพิ่มจำนวน population size

CMAE.DLL มากับโค้ดเต็มรูปแบบในไฟล์ “ADK”

โดย default จำนวนของ runs (or restarts) ตั้งไว้ที่ 5 ซึ่งเราแนะนำให้คงค่านั้นไว้

ผู้ใช้อาจจะปรับเปลี่ยนได้ OptimizerSetOption("Runs", N) และจำนวนควรจะอยู่ในช่วงของ 1..10. การใส่จำนวน Run ที่มากกว่า 10 ขึ้นไปนั้นเราจะไม่แนะนำถึงแม้ว่าเป็นไปได้ก็ตาม

หมายเหตุ จำนวน run ในแต่ละครั้งจะมี Population Size ที่เพิ่มขึ้น 2 เท่า โตแบบ exponentially ดังนั้นหากมี 10 runs คุณจะจบที่ population ที่มีขนาดมากกว่าตัว population ใน run ครั้งแรกถึง 2^10 (1024 times) เท่า

นอกจากนี้ยังมี parameter อีกตัว "MaxEval" ค่า default จะอยู่ที่ ZERO ซึ่งหมายความว่าระบบจะคำนวณหา MaxEval เอง เราแนะนำให้ใช้ค่า Default นี้ ตัว algorithm จะ minimize number of evaluations และเข้าหาคำตอบให้เร็วที่สุดอยู่แล้ว และมักจะหาได้เร็วกว่ากลยุทธ์อื่นๆ

มันเป็นเรื่องปกติที่ระบบจะข้าม evaluations steps ถึงแม้ว่ามันจะหาคำตอบได้ก็ตาม และคุณไม่ควรจะแปลกใจที่ ระบบสามารถ optimize ได้รวดเร็วในบางช่วง plugin มีความสามารถที่จะเพิ่ม number of steps (Evaluation) ถ้าหากจำเป็นต่อการหาคำตอบ เนื่องจากระบบมีลักษณะที่เป็น Adaptive เปลี่ยนแปลงอยู่ตลอดเวลา ดังนั้น "estimated time left" และ "number of steps" ที่แสดงให้เห็นเป็นเพียงค่าประมาณการณ์เท่านั้น

หากต้องการจะใช้ CMA-ES optimizer ใส่โค้ดดังกล่าว:

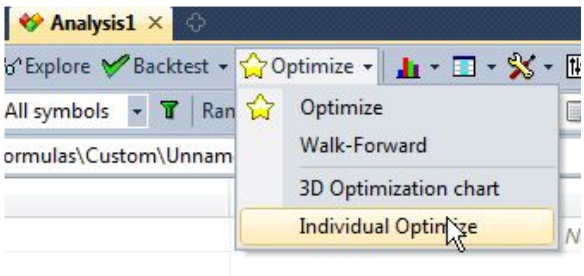
```
OptimizerSetEngine("cmae");
```

ซึ่งระบบจะ Run Optimization ด้วยค่า Default Setting

ข้อสังเกตคือสำหรับการ Optimize แบบ continouos-space search โดยใช้ algorithms นั้น ถ้าหากผู้ใช้ลด Step หรือจำนวนการเพิ่มขึ้นของค่า Parameter ใน Optimize() function นั้นจะไม่มีผลต่อระยะเวลาในการ Optimization มาก สิ่งที่มีผลต่อระยะเวลามากที่สุดคือ "dimension", นั่นก็คือจำนวนของ parameter ที่ต้องการจะ Optimize (จำนวนของฟังกช์ชัน Optimize ที่ถูกเรียก) ดังนั้น จำนวน Step สามารถตั้งได้โดยไม่มีผลกระทบต่อ optimization time โดยทฤษฎีแล้ว algorithm ควรจะหาคำตอบได้ภายใน 900*(N+3)*(N+3) backtests เมื่อ "N" คือค่า dimension โดยปฏิบัติแล้วมันจะเคลื่อนเข้าหาคำตอบเร็วกว่านั้น ตัวอย่างเช่นถ้าให้หาคำตอบเมื่อ N=3 จาก dimensional parameter space ที่ 100*100*100 = 1 million exhaustive stepsคุณสามารถหาได้ภายใน 500-900 CMA-ES steps

Multi-threaded individual optimization

ตั้งแต่ AmiBroker 5.70 ขึ้นไปนอกเหนือจากการทำ multiple-symbol multi threadin แล้วคุณยังสามารถทำ multi-threaded single-symbol optimization ได้ การใช้งานเริ่มต้นจากคลิกเลือกลูกศรลงจากปุ่ม Optimize ในหน้าต่าง New Analysis window และเลือก "Individual Optimize"



"Individual Optimize" จะใช้ทุก processor cores ที่มีอยู่เพื่อประมวลผล single-symbol optimization ทำให้ได้ค่าที่เร็วกว่าการทำ optimization แบบธรรมดา

หากตั้ง "Current symbol" mode ระบบจะ Optimize เพียงตัวนั้นตัวเดียว แต่หาก ตั้งค่า"All symbols" และ "Filter" ระบบจะทำการ Optimize แต่ละตัวตามลำดับไปเรื่อยๆ นั่นคือ Optimize ตัวที่หนึ่งเสร็จ Optimize ตัวที่สองต่อไปเรื่อยๆ

ข้อจำกัด :

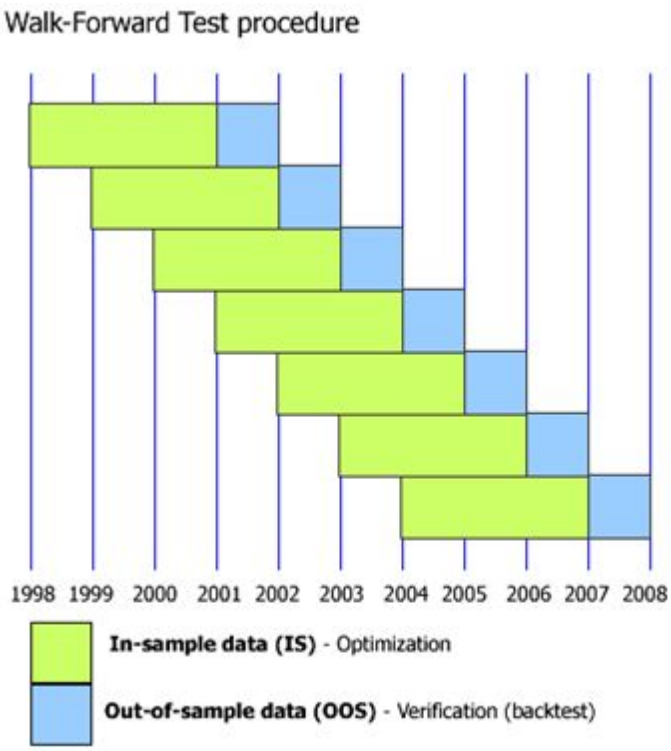
1. ไม่รองรับ Custom backtester
2. ไม่รองรับ Optimization แบบ Smart optimization engines

คำอธิบายของข้อจำกัดสามารถหาดูได้ Tutorial: Efficient use of multi-threading
สุดท้ายแล้ว Amibroker อาจจะรองรับกระบวนการเหล่านี้ในอนาคต

Walk-forward testing

คุณสมบัติเด่นใน AmiBroker ตั้งแต่เวอร์ชั่น 5.10 ขึ้นไปก็ คือ โหมดการทดสอบ Walk-Forward โดยอัตโนมัติ การทดสอบ Walk-Forward โดยอัตโนมัติ นั่นหมายถึง ระบบๆหนึ่ง ที่เราออกแบบไว้ ด้วยสมมุติฐานที่ว่า เราทำการ Optimize พารามิเตอร์ต่างๆของข้อมูลในอดีต (in-sample) และ หลังจากนั้น เราจะวัดผลศักยภาพของระบบของเรา ด้วยการทดสอบมันต่อไปข้างหน้าซึ่งต่อจากข้อมูลชุดเก่าอีกทีหนึ่ง (out-of-sample) (การ วัดศักยภาพของระบบนั้น จะต้องดูผลลัพธ์จากการทดสอบข้อมูล Out of sample เท่านั้น เราจะไม่วัดผลจากข้อมูลที่เรานำมาทำการ Optimize) โดยกระบวนการนี้ เราไม่จำเป็นต้องทดสอบโดยให้ข้อมูลทั้งสองช่วงที่ต่อเนื่องกันก็ได้ (ข้อมูล Out of Sample กับ In Sample ไม่จำเป็นต้องต่อเนื่องกัน)

ภาพประกอบต่อไปนี้แสดงให้เห็นถึงวิธีการทำงานของระบบการทำงานกระบวนการนี้ (โปรดทำความเข้าใจรูปดังกล่าวก็อ่านหัวข้อถัดๆไป)

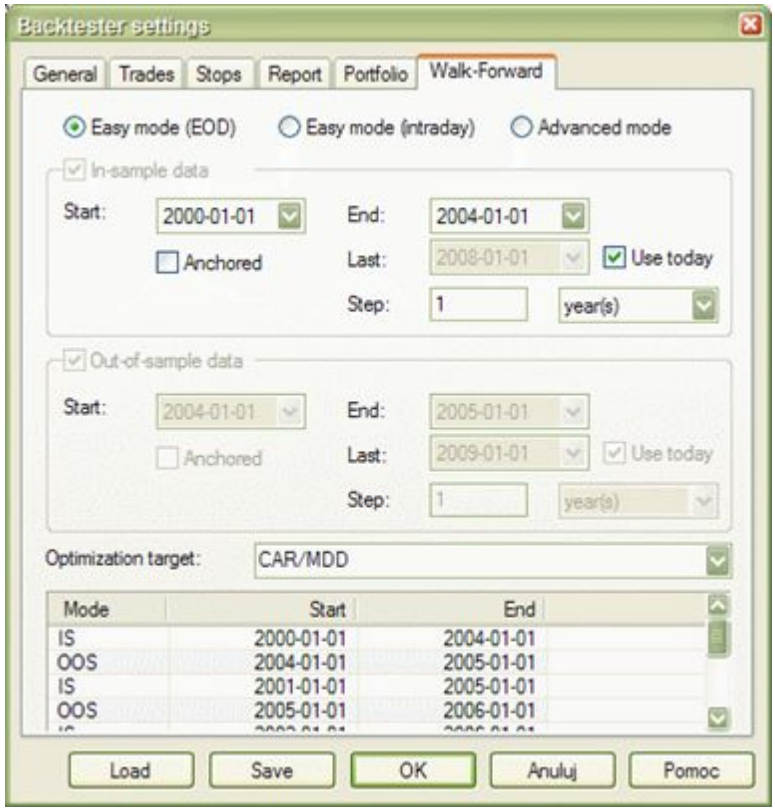


จุดประสงค์ของการทำ Walk Forward คือ การทดสอบว่า ไม่ว่าจะเป็นในช่วงเวลาใดก็ตาม ระบบของเราที่ทำการ Optimize ไว้นั้น จะเป็นอย่างไร มันจะมีความสมจริงที่เป็นไปได้ (realistic) หรือ เกิด curve-fitting กันแน่ ศักยภาพของระบบสามารถที่จะวัดผลของความสมจริงได้ (realistic) มีเงื่อนไขสองอย่าง คือ ผ่านการทดสอบข้อมูลในส่วนที่เราไม่เห็นข้อมูลมาก่อน (ข้อมูล Out of sample) และ อีกข้อหนึ่งคือ เมื่อเราออกแบบระบบเสร็จเรียบร้อยแล้วเท่านั้น ในการเทรดจริง ประสิทธิภาพของระบบควรจะใกล้เคียงกับตอนที่เราทำการ Optimize เพราะฉนั้น ระบบ จะใช้งานได้ดีก็ต่อเมื่อมันถูกทดสอบด้วย walk forward แล้ว หรือ เราสามารถที่จะพูดได้ในอีกทางหนึ่งคือ เราไม่ได้สนใจเกี่ยวกับผลลัพธ์ของการทดสอบในช่วง In sample (เพราะมันแน่นอนอยู่แล้วว่าผลลัพธ์มันจะต้องออกมาว่าดี) สำหรับการทดสอบกันข้อมูล Out of Sample นั้น จะเป็นการประมาณการที่มีความสมจริง และ วัดประสิทธิภาพของระบบออกมาได้ ว่าเกิดปัญหาจาก curve fitting หรือไม่ และ ถ้าหากผลลัพธ์จาก Out of sample ออกมาไม่ดีแล้วนั้น คุณก็ไม่ควรที่จะนำระบบนี้มาเทรดจริงเช่นกัน

ค่าพารามิเตอร์ที่เป็นผลลัพธ์ที่ได้มาจากการ optimize นั้น มีหลักฐานบ่งชี้ว่าการที่เรา optimize ข้อมูลในอดีตที่เพิ่งผ่านมา จะได้ผลลัพธ์ที่ดีกว่าการที่เรา optimize ข้อมูลที่ผ่านมานานมากๆแล้ว พวกเราคาดหวังเป็นอย่างยิ่งเสมอว่าพารามิเตอร์ที่เราเลือกในการ optimize นั้น จะสอดคล้องกับสภาวะของตลาดในตอนที่เรานำระบบไปใช้ เราจะได้สามารถนำมันไปใช้ได้ทันที (คือ Optimize เสร็จ ก็หวังว่าพารามิเตอร์ที่เรา Optimize ไว้นั้นมันจะยังคงใช้ได้ดีอยู่) แต่ช่วงเวลาดังกล่าวที่เราเลือกมาทดสอบนั้น อาจจะเป็น หรืออาจจะไม่เป็นช่วงเวลาที่ตลาด ผ่านวัฐจักรทั้งแบบ ขาลง และ ขาขึ้น ก็ได้ ดังนั้น คุณควรที่จะระมัดระวังในการเลือกช่วงระยะเวลาที่จะการนำมาใช้สำหรับการทดสอบ ข้อมูลเพิ่มเติมเกี่ยวกับการออกแบบระบบ และ การตรวจสอบการใช้กระบวนการ การทำ walk-forward รวมถึง ขั้นตอนและทุกประเด็นที่เกี่ยวข้องนั้น เราแนะนำให้คุณที่จะอ่านหนังสือของ Howard Bandy : "Quantitative Trading Systems" (ดูลิงค์ได้บนหน้าเว็ปไซต์ AmiBroker) เพื่อที่คุณจะได้มีความเข้าใจที่มากขึ้น

เพื่อการใช้งานฟังชันก์ Walk-Forward optimization โปรดทำตามขั้นตอนดังต่อไปนี้ :

1.ไปที่ Tools-> Automatic Analysis (ถ้าหาไม่เจอให้เข้าหน้า Analysis เหมือนตอนที่เราทำ backtest)
2. คลิกปุ่ม Setting และ เลือกไปที่แถบข้อมูลที่ชื่อว่า Walk-Forward



3. ทีนี้คุณก็จะเห็นการตั้งค่าการทำ Walk Forward ในส่วนของ In-sample เพื่อทำการ optimization และ out-of-sample เพื่อทำการ backtest

วันที่ Start และ End คือการเลือกช่วงเวลาที่ทำการเริ่มต้น และ สิ้นสุด ช่วงเวลาดังกล่าวที่เราตั้งค่าไว้จะเคลื่อนที่ไปข้างหน้าทีละ Step จนกระทั้ง End (ไปตรงกับวันสุดท้ายที่เราตั้งค่าไว้พอดี)

วันที่เริ่มต้น (Start) เราสามารถตั้งค่าให้มันเคลื่อนที่ไปข้างหน้าเป็น Step หรือ ถ้าอยากตั้งล็อกให้วันที่ทำการเริ่มต้นที่เวลาเดียวกันก็ได้ หากติ๊กที่ช่อง Anchored

นอกจากนี้ หากคุณติ๊กช่อง Use today ข้อมูลในช่อง Last นั้นจะไม่ถูกนำมาพิจารณา แต่จะดึงข้อมูลในวันปัจจุบันมาใช้แทน

โดยค่าเริ่มต้นของโปรแกรม จะถูกเลือกไว้ให้เป็น “EASY MODE” ซึ่งเป็น กระบวนการที่ง่ายที่สุดของการตั้งค่าในการทำทำ Walk Forward

ซึ่งมันมีสมมุติฐานดังนี้:

a) ส่วนของ Out-of-sample จะต่อเนื่องกับชุดข้อมูลในส่วนของ In-sample
b) ระยะความกว้างของ Out-of-sample หนึ่งช่วงจะมีค่าเท่ากับความกว้างของ Step

นอกจากสมมุติฐาน 2 ข้อ ดังกล่าวแล้ว “EASY Mode” จะใช้ วันสุดท้าย(END) ของ in-sample ให้เป็นวันเดียวกับวันที่เริ่มต้น (START) ของ out-of-sample และ จะใช้ช่วงเวลที่ระบุใน STEP ของ In-sample เป็นตัวกำหนดเวลาในการทำ out-of-sample

ค่าของ Step ที่ตั้งไว้ใน In-sample และ Out-of-sample นั้นจะถูกตั้งค่าไว้เป็นค่าเดียวกัน (“EASY Mode” นั้นจะรับประกันความถูกต้องของกระบวนการ Walk-Forward ในการตั้งค่า)

คุณควรจะใช้ Easy mode (EOD) เมื่อคุณทดสอบข้อมูลที่เป็น End-of-day หรือ Easy mode (Intraday) เมื่อคุณทดสอบข้อมูลที่เป็น Intraday สิ่งที่แตกต่างกันก็คือ ใน “EOD mode” วันสุดท้าย (END) ของช่วงเวลาก่อนหน้า และ วันเริ่มต้น (START) ของช่วงเวลาถัดไป จะเป็นวันเดียวกัน เพื่อป้องกันการเกิดช่องว่างในการทำการทดสอบระหว่าง 2 ช่วงเวลา ส่วน “Intraday mode” จะตั้งให้วันเริ่มต้น (START) ของช่วงเวลาถัดไป เป็นวันถัดไป หลังจากวันสุดท้าย (END) ของช่วงเวลาก่อนหน้า ซึ่งการทำแบบนี้ ทำให้สามารถรับประกันได้ว่า จะไม่เกิดการทำซ้ำเมื่อทดสอบข้อมูลแบบ intraday

ใน Advanced mode ผู้ใช้งานจะสามารถควบคุม ค่าต่างๆได้ที่ถูกล็อกไว้ใน Easy Mode ได้ทั้งหมด เพื่อที่จะขยายขอบเขตการทำงานของกระบวนการทำ Walk Forward หน้าต่างจะเปิดให้เลือกตั้งค่าในส่วนของ In-sample และ Out-sample โดยการติ๊กเครื่องหมายถูก (สามารถที่จะให้ทำการ backtest โดยปราศจากการ Optimize ได้) การตั้งค่าทั้งหมดจะแสดงผลทันทีในช่องหน้าต่างแสดงที่ด้านล่าง ในส่วนของ IS/OOS และ วันเวลาที่กำหนด

4. ในช่องของ Optimization target คือ การเลือกผลลัพธ์การ Optimize ที่เราต้องการให้แสดงออกในรูปของ Colum ซึ่งมันจะถูกใช้เพื่อเรียงข้อมูลเพื่อหาผลลัพธ์ที่ดีที่สุดอย่างหนึ่ง (เอาไว้ให้เราใช้ sort ข้อมูล) ในหน้าต่างของการทดสอบ ค่าเริ่มต้นที่โปรแกรมตั้งไว้ก็คือ CAR/MDD ซึ่งคุณสามารถที่จะเปลี่ยนมันเป็นค่าอื่นๆ ก็ทำได้เช่นกัน ตามที่คุณต้องการ นอกจากนี้คุณสามารที่จะใส่ค่า เพื่อที่จะสร้างค่าผลลัพธ์ที่คุณต้องการจะให้โปรแกรมแสดงผลลัพธ์ออกมาได้เช่นกันด้วยการใช้ custom backtester interface

5. หลังจากที่คุณตั้งค่าต่างๆเรียบร้อยแล้วให้คุณกลับไปที่ Automatic Analysis

6. คุณจะเห็บแถบด้านล่างของการทดสอบ ให้คุณเลือกไปที่แถบชีทที่เขียนว่า “Walk Forward Optimization” โปรแกรมจะ รันลำดับของการ optimize และการ backtest รวมถึงผลลัพธ์ในหน้าต่างแสดงผลของ Walk Forward ขณะที่การ optimize กำลังประมวลผลอยู่ คุณสามารถที่จะคลิกปุ่ม “MINIMIZE” โปรแกรมจะแสดงผลลัพธ์ของการ Walk Forward ขณะที่มันกำลังประมาณผลได้ทันที

การสร้างเส้น IN-SAMPLE และ OUT-OF-SAMPLE (IN-SAMPLE and OUT-OF-SAMPLE combined equity)

เส้น equities ในส่วนของ in-sample และ out-sample นั้นถูกแยกเก็บกันไว้อย่างชัดเจนใน ticker ชื่อ ~~~ISEQUITY และ ~~~OSEQUITY(ช่วงเวลาที่ต่อเนื่องกันของ IS และ OOS จะถูกแบ่งแยกกันในคนละสเกล เพื่อจะทำให้เส้น equity มีความต่อเนื่องกัน โดยทั่วไปแล้วจะเป็นการพูดถึง compounding profits)

คุณสามารถแสดงผลของเส้น equityทั้งในส่วน IS และ OOS ดังต่อไปนี้ได้:

```
PlotForeign("~~~ISEQUITY","In-Sample Equity", colorRed, styleLine);
PlotForeign("~~~OSEQUITY","Out-Of-Sample Equity", colorGreen, styleLine);
Title = "{{NAME}} - {{INTERVAL}} {{DATE}} {{VALUES}}";
```

บันทึกสรุปผลลัพธ์โดยรวมจากข้อมูล OUT-OF-SAMPLE (ใหม่สำหรับเวอร์ชั่น 5.60) (OUT-OF-SAMPLE summary report (new in 5.60))

นี่เป็นการเปลี่ยนแปลงครั้งใหญ่ในการทำการทดสอบแบบ walk Forward เพื่อที่จะสร้างบันทึกสรุปผลจากข้อมูล Out-of-sample ออกมา สิ่งที่สำคัญที่สุดในการเปลี่ยนแปลงในครั้งนี้คือ ในการลำดับข้อมูลในแต่ละชุดนั้น เงินทุนเริ่มต้นจะใช้เท่ากับเงินทุนของช่วงก่อนหน้า (เอาเงินทุนทั้งหมดหลังจากการทดสอบช่วงก่อนหน้าเสร็จ) จากที่แต่เดิมใช้เป็นค่าเงินทุนเริ่มต้นคงที่เลย (เช่น เงินทุนเริ่มต้น 1 แสน พอทดสอบข้อมูลชุดใหม่ก็ใช้ 1 แสนไปเรื่อยๆ ไม่คำนึงถึงต้นทุนก่อนหน้าที่เปลี่ยนแปลงไปจากการลงทุน) การเปลี่ยนแปลงนี้จะเหมาะสมสำหรับการคำนวณทางสถิติ / ตัวชี้วัดทั่วทุกส่วนจากการทดสอบข้อมูล out-of-sample

บันทึกสรุปผล จะแสดงถึงข้อมูลตัวชี้วัดต่างๆอย่างถูกต้องจากข้อมูล out-of-sample ทั้งหมด แต่บันทึกสรุปผล[แบบกำหนดเอง] จะต้องถูกระบบโดยผู้ใช้ ดังนี้

1 ค่าเริ่มต้น, 2 ค่าสุดท้าย, 3 ค่ารวม, 4 ค่าเฉลี่ย, 5 ค่าต่ำสุด, 6 ค่าสูงสุด

โดยค่า default ของผลสรุปรวมนั้นจะแสดงค่าสุดท้ายของตัวชี้วัดที่เรากำหนด เว้นแต่ว่า ผู้ใช้จะกำหนดตัวชี้วัดอย่างเฉพาะเจาะจงในรูปแบบ

bo.AddCustomMetrics() call

bo.AddCustomMetrics คือ การกำหนดค่าพารามิเตอร์ที่จะแสดงผลใหม่ - CombineMethod

bool AddCustomMetric(string Title, variant Value, [optional] variant LongOnlyValue, [optional] variant ShortOnlyValue , [optional] variant DecPlaces = 2, [optional] variant CombineMethod = 2)

วิธีการนี้จะเพิ่มตัวบ่งชี้ที่เรากำหนดเองในผลลัพธ์การ backtest ของเราในส่วนของ backtest report , backtest "summary" และ ผลลัพธ์ของการ optimization

Title คือชื่อของตัวบ่งชี้ที่จะแสดงในบันทึกสรุปผล, Value คือ ค่าของตัวบ่งชี้นั้นๆ , LongOnlyValue, ShortOnlyValue เพื่อรองรับ ค่าของการ long/short เพื่อสร้างคอลัมในบันทึกการสรุปผลและ ตัวสุดท้าย DecPlaces จะคอยควบคุมจำนวนตำแหน่งทศนิยมที่จะแสดงผลออกมา

ข้อมูลการสนับสนุน CombineMethod มีค่าต่างๆดังนี้ :

1 ค่าเริ่มต้น - บันทึกสรุปผล จะแสดงค่าตัวบ่งชี้ที่กำหนดจากขั้นตอนเริ่มแรกสุดของข้อมูล out-of-sample

2 ค่าสุดท้าย (default), - บันทึกสรุปผล
จะแสดงค่าตัวบ่งชี้ที่กำหนดจากขั้นตอนสุดท้ายสุดของข้อมูล out-of-sample

3 ค่ารวม - บันทึกสรุปผล จะแสดง ”ผลรวม”
ของค่าตัวบ่งชี้ที่เรากำหนดจากขั้นตอนทั้งหมดของข้อมูล out of sample

4 ค่าเฉลี่ย - บันทึกสรุปผล จะแสดง “ค่าเฉลี่ย”
ของค่าตัวบ่งชี้ที่เรากำหนดจากขั้นตอนทั้งหมดของข้อมูล out of sample

5 ค่าต่ำสุด - บันทึกสรุปผล จะแสดง “ค่าต่ำที่สุด”
ของค่าตัวบ่งชี้ที่เรากำหนดจากขั้นตอนทั้งหมดของข้อมูล out of sample

6 ค่าสูงสุด - บันทึกสรุปผล จะแสดง “ค่าสูงที่สุด”
ของค่าตัวบ่งชี้ที่เรากำหนดจากขั้นตอนทั้งหมดของข้อมูล out of sample

หมายเหตุ : วิธีการคำนวณตัวชี้วัดนั้นมีความซับซ้อนเป็นอย่างมาก ยกตัวอย่างคือ การหาค่าเฉลี่ย มันจะมีการคำนวณทางคณิตศาสตร์ที่ไม่ถูกต้องเท่าไหร่นักหากคำนวณเอง ผลสรุปของตัวบ่งชี้ที่มีอยู่แล้วในโปรแกรม จะมีการคำนวณทางคณิตศาตร์ที่ถูกต้อง out-of-the-box (ยกตัวอย่างเช่น มันจะไม่ทำการหาค่าเฉลี่ยโดยตรง แต่จะหาวิธการคำนวณที่เหมาะสมกับตัวบ่งชี้นั้นๆ) ความขัดแย้งนี้เกิดจากการสร้างตัวบ่งชี้ด้วยตัวเอง เนื่องจากพวกเขาใช้ user-definable (โหมดที่ผู้ใช้เป็นคนกำหนดตัวแปรด้วยตนเองทั้งหมด) ซึ่งมันขึ้นอยู่กับผู้ใช้ที่จะเลือกรวม (combining) วิธีการต่างๆเข้าด้วยกัน มันยังคงเกิดขึ้น การวิธีการคำนวณแบบประมาณค่าต่างๆ

การจำลองด้วยวิธี Monte Carlo

(Monte Carlo Simulation of your trading system)

หมายเหตุ: หัวข้อขั้นสูง โปรดแน่ใจแล้วว่าคุณได้อ่านหัวข้อก่อนหน้า

บทนำ (Introduction)

โดยทั่วไปแล้ว หากพูดถึงวิธีการทำ "Monte Carlo" จะถูกแทนในเรื่องของวิธีการทางคอมพิวเตอร์ที่มีการทดสอบแบบ ซ้ำแล้วซ้ำอีก เพื่อให้ได้คุณสมบัติตามกระบวนการกำหนดไว้ มันถูกสร้างขึ้นโดย นักนักคณิตศาสตร์ ชาวโปแลนด์ที่มีชื่อว่า Stanislaw Ulam เรื่องมีอยู่ว่า ขณะนั้นเค้าทำงานเกี่ยวกับการวิเคราะห์อาวุธนิวเคลียร์ที่ Los Alamos lab ในขณะที่เขาไม่สามารถที่จะวิเคราะห์ กระบวนการทางกายภาพที่ซับซ้อนได้ด้วยวิธีการทางคณิตศาสตร์ทั่วๆไป เขาก็นึกขึ้นได้ว่าเขาสามารถที่จะสร้างชุดการทดลองแบบสุ่มขึ้นมาเพื่อที่จะเอาไว้สังเกตุมันได้

เกี่ยวกับเรื่องราวและวิธีการเกี่ยวกับ Monte Carlo โดยทั่วไปแล้ว คุณสามารถเข้าไปศึกษาข้อมูลเพิ่มเติมได้ที่นี่ https://en.wikipedia.org/wiki/Monte_Carlo_method

ในการพัฒนาระบบเทรดนั้น การสร้างแบบจำลอง Monte Carlo หมายถึง กระบวนการของการสุ่มลำดับของการเทรด เพื่อประเมินผลทางสถิติของระบบการเทรดนั้นๆ มันมีอยู่หลายวิธีการทีเดียว ที่จะคำนวณผลลัพธ์ออกมาด้วยวิธีการที่แตกต่างกันไป (แตกต่างกันเมื่อลึกลงไปรายละเอียด) แต่วิธีการคำนวณที่ตรงไปตรงมามากที่สุด และดูสมจริงมากที่สุดก็คือวิธีการทำ bootstraping ซึ่งดำเนินการสุ่มตัวอย่างโดยการแทนที่ trade list ที่เกิดจากการ backtest ลงไป

ไปที่ https://en.wikipedia.org/wiki/Bootstrapping_(statistics) เพื่อศึกษาเกี่ยวกับการพิจารณาด้วยวิธี bootstrapping

วิธีการจำลองด้วยวิธี Monte Carlo แบบต่างๆนั้น จะช่วยให้ระบบเทรดมีความยั่งยืน (robustness) และช่วยหาว่า โอกาสทางสถิติที่ระบบจะพังเป็นเท่าไหร่ รวมถึงสามารถที่จะหาค่าทางสถิติแบบอื่นๆ ได้อีกด้วย

แล้วมันทำงานอย่างไรบน AmiBroker? (How does it work in AmiBroker?)

เพื่อที่จะทำ Monte Carlo simulation (หรือการทดสอบ bootstrap) กับระบบของคุณ, กระบวนการบน Amibroker เป็นดังนี้ :

A. สร้างชุดข้อมูลที่จะทำการป้อน

A.1 ดำเนินการ backtest ระบบเทรดของคุณ เพื่อสร้างชุดข้อมูลการเทรดดังเดิมทั้งหมด N เทรด

B. ทำซ้ำ (1000+ ครั้ง)

B.1 สุ่มเลือก trade list จากข้อมูลการเทรดดังเดิม เพื่อที่จะนำมาเรียงและสร้างข้อมูลการเทรดชุดใหม่ๆออกมาก ซึ่งจะส่มจากการเทรดทั้งหมด N ครั้ง (ข้อมูลทั้งหมดที่เกิดจากวิธีนี้ เรียกว่า realization)

ชุดข้อมูลที่เกิดจากการสุ่มดังกล่าวจะยังมีจำนวนข้อมูลในการเทรดเท่าเดิม ระบบจะทำการสุ่ม รายการเทรดบางรายการก็อาจจะถูกข้ามไป หรือ ถูกใช้ซ้ำก็ได้ (เป็นการเปลี่ยนแปลงโดยการทำซ้ำ หรือการเปลี่ยนแปลงโดยการแทนที่) (จากผู้แปล : ตรงนี้เหมือนกับการที่เราสุ่มเลือกรายการเทรดที่มีตั้งแต่ A ถึง D (สมมุติให้มีการสุ่ม 4 ครั้ง) สมมุติว่าเราหยิบได้ A เมื่อเราเอา A เก็บเข้าไปแล้วการสุ่มครั้งหน้าเราอาจจะได้ A อีกก็ได้ และเมื่อเราจบการสุ่มทั้งหมด 4 ครั้งแล้วก็ไม่จำเป็นที่ต้องสุ่มได้ครบทั้ง A B C D ถูกไหมครับ อาจจะขาดตัวใดตัวหนึ่งไปก็ได้) เนื่องจากข้อมูลของการทำ realization ที่ไม่ซ้ำกันคือ N^N (ดังนั้นแค่การเทรด 100 เราก็จะมี 10000 realization ที่แตกต่างกันเรียบร้อยแล้ว) และด้วยจำนวนการเทรดที่มีนัยยะสำคัญนั้น (>100) โอกาสที่จะเลือกได้ข้อมูลที่มีการเรียงกันของ trade list ดังเดิมชุมเดิมนั้น แทบจะเป็นศูนย์เลยทีเดียว

B.2 การคำนวณ gain/loss ในแต่ละการสุ่มเลือกนั้น จะใช้ position sizing ในการสร้าง equity ของระบบ

B.3 บันทึก equity ของระบบโดยการกระจายตัว (distribution)

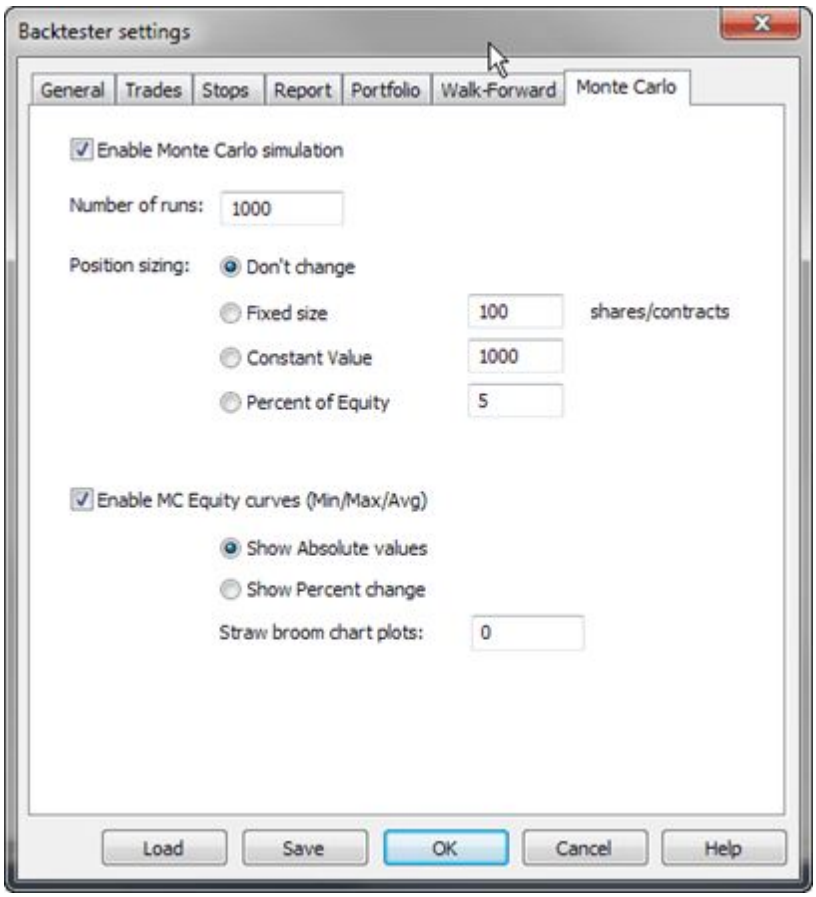
C. กระบวนการ

C.1 ประมวลผลข้อมูลที่ได้จากขั้นตอน B เพื่อสร้างการกระจายตัวทางสถิติ และ กราฟ ขึ้นมา

ขั้นตอนทั้งหมดที่กล่าวมาด้านบนนี้ จะเกิดขึ้นเมื่อคุณกดปุ่ม Backtest แค่ปุ่มเดียวเท่านั้น ในหน้าต่าง New Analysis กระบวนการจำลองด้วยวิธี Monte Carlo บน Amibroker นั้น รวดเร็วมากและใช้ เวลาแค่เสี้ยววินาทีเท่านั้น บนการ backtest แบบปกติ

การตั้งค่า (Settings)

การที่การจำลองด้วยวิธีการ Monte Carlo จะทำงานได้นั้น สามารถควบคุมทั้งหมดได้จากหน้า Analysisi Setting ในส่วนของแถบ Monte Carlo



Enable Monte Carlo simulation

เมื่อกล่องนี้ถูกติ๊กเมื่อไหร่ก็ตาม ระบบจะทำการจำลองแบบ Monte Carlo โดยอัตโนมัติ โดยเป็นส่วนหนึ่งของการ backtest (ข้อมูลจะแสดงผลอยู่ใน backtest report ในทางแถบด้านขวาเมื่อทำการ backtest เสร็จ)

Number of runs

แสดงถึงจำนวนครั้งในการทำการจำลองแบบ Monte Carlo (ควรที่จะมากกว่า 1000 ครั้ง)

Position sizing

เพื่อเลือกการใช้ position sizing ในการทำแบบจำลองด้วยวิธีการ Monte Carlo :

Don't change - ใช้ขนาดของ position size ดั้งเดิมที่ใช้อยู่ในระหว่างการ backtest

Fixed size - ใช้จำนวนคงที่ของจำนวนหุ้น และ สัญญา ต่อการเทรด

Constant value - ใช้จำนวนเงินเท่าเดิมในการเทรดแต่ละครั้ง

Percent of equity - ใช้เปอร์เซ็ยของการจำลองของมูลค่า equity ก่อนหน้า

Enable MC equity curves (Min/Max/Avg)

เพื่อเปิดกราฟ Monte Carlo equity (รวมถึง ค่าสูงสุด, ต่ำสุด และ ค่าเฉลี่ยของ equity นอกจากนี้ยังมีกราฟที่แสดงถึง equity ในแต่ละการสุ่มแต่ละครั้งอีกด้วย)

หมายเหตุ : เส้นสีเขียว และ สีแดง (min/max equity) ไม่ได้หมายถึง สัญญาณที่แย่ที่สุด หรือ ดีที่สุด พวกมันหมายถึงจุดที่ bar-by-bar สูงที่สุด (max) และ ต่ำที่สสุด(min) ของ equities ทั้งหมดที่ถูกสร้างด้วยวิธีการจำลองแบบ Monte Carlo

ดังนั้นพวกมันจึงเป็นทั้ง จุดที่ดีที่สุด ของ equity ทั้งหมด แต่ก็เป็นจุดที่แย่ที่สุด ของ equity ทั้งหมดเช่นกัน นอกจากนี้ เส้นสีน้ำเงิน (avg) แสดงถึงค่าเฉลี่ยของ equity ทั้งหมด

Show absolute value - แสดงมูลค่า แอปโซลูท ของ equities

Show Percent change - แสดง equities เป็น "rate of change" ตั้งแต่ที่มันเริ่มมา

Straw broom chart plots - กำหนดจำนวนการทดสอบ equites ซึ่งควรสร้างออกมาในรูปของกราฟ Straw broom (หากจำนวนเยอะมากๆ จะทำให้กระบวนการส่วนนี้เกิดความล่าช้า)

การอ่านและตีความผลลัพธ์จากการทดสอบแบบ Monte Carlo (Interpreting the results)

ผลลัพธ์ของการจำลองด้วยวิธีการ Monte Carlo จะแสดงผลไว้ในหน้า "Monte Carlo ของผลลัพธ์การ backtest

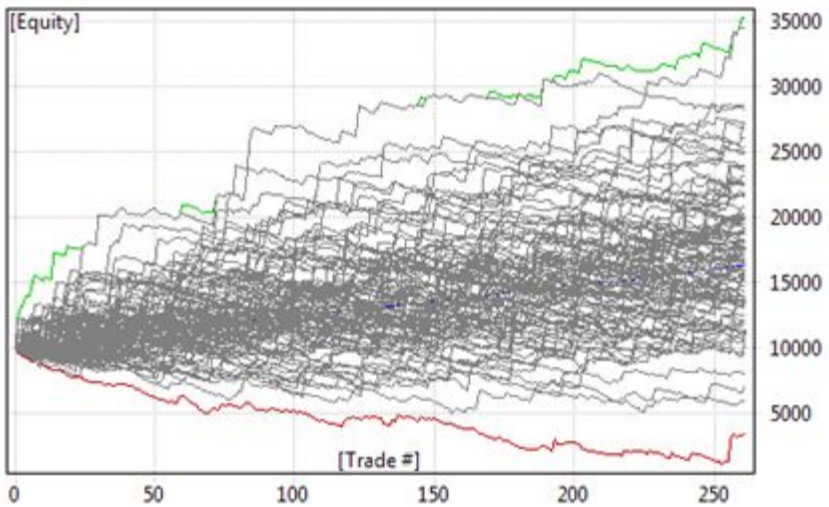
ด้านบนสุดของหน้า เราจะเห็นตารางที่แสดงถึงค่าสำคัญๆทางสถิติ ที่เกิดจากการกระจายตัวของผลลัพธ์การจำลองด้วยวิธี Monte Carlo และนี่ คือตัวอย่างของผลลัพธ์ (บริเวณที่เน้นเป็นสีน้ำเงินเพื่อง่ายต่อการอธิบาย ตอนผลลัพธ์ออกมาจริงๆจะไม่มี) ในตัวอย่าง จะเริ่มต้นด้วยเงิน 10,000 โดยสอบกับข้อมูล End-of-day เป็นเวลา 7 ปี

	Final Equity	Annual Return	Max. Drawdown $	Max. Drawdown %	Lowest Eq.
1%	5706	-7.37%	1302	7.23%	3618
5%	7987	-3.02%	1549	9.76%	5853
10%	9706	-0.41%	1726	11.32%	6690
25%	12851	3.48%	2136	14.38%	8107
50%	16174	6.78%	2747	19.77%	9135
75%	19632	9.64%	3563	27.63%	9640
90%	23258	12.21%	4626	38.48%	9922
95%	25269	13.48%	5292	45.47%	10000
99%	29139	15.71%	7685	63.82%	10000

คอมลัมแรก แสดงถึงระดับเปอร์เซนต์ไทด์ (ค่าด้านล่างจะให้ค่าเปอร์เซ็นต์ไทด์ที่ใช้ในการสังเกตการที่ระบบจะล้มเหลว) ลองมาดูกันที่ ระดับเปอร์เซ็นต์ไทด์ที่ 10 มันบอกว่าจากการสังเกตข้อมูล 10% ได้ผลลัพธ์ออกมาดังค่าตามตารางนี้ ยกตัวอย่างเช่น ค่าของ annual return ที่ ระดับเปอร์เซ็นต์ไทด์ที่ 10 จะเท่ากับ -0.41% หมายถึง 10% จากการทดลองทั้งหมด (realizations) จะมี annual profit น้อยกว่าหรือเท่ากับ

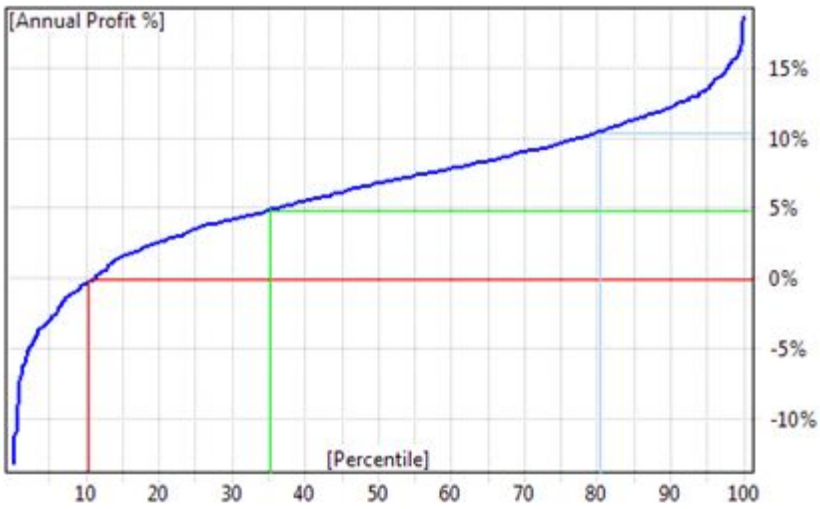
ที่แสดงไว้ (-0.41%) ดังนั้นเราสามารถที่จะบอกได้ว่ามีโอกาส 10 % ที่ระบบของเราจะไม่ทำเงินให้เราเลย นอกจากนี้ ค่า max. drawdown ที่ระดับเปอร์เซ็นต์ไทด์ที่ 90 (38.48%) หมายถึง จากตัวอย่าง 90% ของการทดลองทั้งหมด drawdown จะมีค่าน้อยกว่า 38.48% ดังนั้น สิ่งที่ตารางนี้จะบอกคุณก็คือระบบนี้มันมีโอกาสเพียงแค่ 10% ที่ drawdown จะมีค่าสูงกว่านั้น ถ้าหากเราสังเกตเพิ่มเติมที่ระดับ 99% ก็จะเห็นว่ามันมีโอกาสถึง 99% ที่ drawdown ของระบบเราจะ น้อยกว่า 63.82%

ด้านล่างจะแสดงถึงค่า min/avg/max และ straw broom chart ของ equity ออกมา



หมายเหตุ : ย้ำอีกครั้งว่า เส้นสีเขียว และ สีแดง (min/max equity) ไม่ได้หมายถึง สัญญาณที่แย่ที่สุด หรือ ดีที่สุด พวกมันหมายถึงจุดที่ bar-by-bar สูงที่สุด (max) และ ต่ำทีสสุด(min) ของ equities ทั้งหมดที่ถูกสร้างด้วยวิธีการจำลองแบบ Monte Carlo ดังนั้นพวกมันจึงเป็นทั้ง จุดที่ดีที่สุด ของ equity ทั้งหมด แต่ก็เป็นจุดที่แย่ที่สุด เช่นกัน และ เส้นสีน้ำเงิน (avg) แสดงถึงค่าเฉลี่ยของ equity ทั้งหมด และเส้นที่เทา เป็นตัวแทนของเส้น equity ในแต่ละการทดสอบ ซึ่งเราสามารถเห็นได้ว่า ถึงแม้จะใช้ระบบการเทรดเดียวกัน ก็สามารถที่จะสร้างผลลัพธ์ที่แตกต่างกันออกมาได้ เมื่อเงื่อนไขของสภาพตลาดเปลี่ยนแปลงไป และ การจำลองด้วยวิธี Monte Carlo ก็พยายามที่จะ สร้างผลลัพธ์ที่แตกต่างกันออกมา และ เตรียมค่าทางสถิติให้คุณนำไปพิจารณาข้อดี ข้อเสียของระบบแล้ว

หลังจาก straw broom chart คุณก็สามารถที่จะหา cumulative distribution function (CDF) chart ของ equity สุดท้ายได้ ไม่ว่าจะเป็น CAR, drawdowns และ lowest equity (และอีกเช่นกัน เส้นสีเขียว และ สีแดง แสดงไว้ให้เพื่อการสังเกตในคู่มือเล่มนี้เท่านั้น ในโปรแกรมจริงๆจะไม่มี)



Cumulative distribution แสดงถึงข้อมูลเดิมที่อยู่ในตารางด้านบนของหน้า Monte Carlo แต่แสดงออกมาในรูปของกราฟ อีกครั้ง เมื่อเราสังเกตรูปแบบการกระจายตัวของ annual profit % (CAR) เราสามารถที่จะประมาณได้ว่า ใน 10 % ของระบบทั้งหมดจะไม่ break even (ยังคงให้ค่า CAR ที่ติดลบอยู่) และ เราสามารถที่จะประมาณได้ ค่าที่ประมาณ 35% CAR ของเราก็ยังคงจะต่ำกว่า 5 % นอกจากนี้นั้น Profits มากกว่า 10% จะเกิดขึ้นก็ต่อเมื่อ เป็นค่าที่ดีที่สุด 20 % ของการทดลองทั้งหมด

ในส่วนของกราฟอื่นๆใน หน้า Monte Carlo ก็จะมีรูปแบบการตีความที่เหมือนกับตัวอย่างด้านบน

กราฟแสดง ค่า equity ตอนเมื่อจบการทดลองนั้น จะแสดงออกมาในรูปของ cumulative distribution function ของมูลค่า equity ตอนสุดท้าย (ในจุดสิ้นสุดของการ backtest)

กราฟของ Annual return ก็จะแสดงถึง cumulative distribution function ของ compound annual ในหน่อยของเปอร์เซ็นต์จากการ backtest

กราฟของ Max. Drawdown $ และ Max. Drawdown % แสดงcumulative distribution function ของ drawdowns (ระยะทางจากสุด peak ถึง valey ในหน่อย เงิน/เปอร์เซ็นต์) ที่ระบบประสบเจอระหว่างการ backtest

กราฟของ Lowest Equity แสดงcumulative distribution function ของ lowest equity ที่ระบบประสบเจอระหว่างการ backtest

จะควบคุมมัน จากการใส่สูตร/โค้ด (formula) ได้อย่างไร? (How to control it from the formula level?)

นอกเหนือจากการตั้งค่าที่ได้กล่าวไว้ด้านบนแล้ว คุณสามารถที่จะใช้คำสั่ง SetOption() ฟังก์ชัน เพื่อควบคุมการทำงานของ การสร้างแบบจำลองแบบ Monte Carlo ได้เช่นกัน นอกจากนี้คุณสามารถที่จะเรียกดูค่าต่างๆได้ด้วยการใช้ GetOption ฟังก์ชัน

```
SetOption( "MCEnable", 0 );  // value == 0 หมายถึงไม่เรียกใช้ Monte Carlo simulation
SetOption( "MCEnable", 1 ); // value == 1 หมายถึง เรียกใช้ Monte Carlo เฉพาะในการทำ backtest (default)
SetOption( "MCEnable", 2 ); // value == 2 หมายถึง ใช้ Monte Carlo ในทุกๆการทดสอบ (ในทุกๆ mode แม้กระทั้งการ optimization ซึ่งจะทำให้กระบวนการยาวนานมาก)
```

โปรดทราบไว้ว่าไม่แนะนำเป็นอย่างมากให้เปิดการใช้งาน Monte Carlo ในการ optimization ถ้าคุณไม่ได้ต้องการที่จะ ใช้มันเพื่อหาค่า target นั้นจริงๆ ในอีกนัยหนึ่ง การใช้การกระจายตัวของ Monte Carlo บนกระบวนการ optimization กระบวนการ Monte Carlo จะเสียเวลาเอามากๆ และ ในขณะที่การทำจะใช้เวลาในการ backtest หนึ่งครั้งใช้เวลาเพียงไม่กี่ร้อยมิลลิวินาทีเท่านั้น ซึ่งมันแทบจะไม่มีเสียเวลาอะไรเลย แต่ในกรณีการทำ optimizations หลายๆตัวแปร คุณสามารถที่จะทำให้เวลาในการทำงานของมันเพิ่มขึ้นได้อย่างง่ายดาย ดังนั้นหากคุณไม่ต้องการใช้ การกระจายตัวของ Monte Carlo ในการสร้างตัวบ่งชี้จริงๆ และ เพื่อหาเป้าหมายในการ optimization อย่าเปิดใช้งาน Monte Carlo ในการ optimization จะดีกว่า

SetOption("MCRuns", 1000); // กำหนดจำนวนครั้งในการ รัน Monte Carlo (realizations) ค่าพารามิเตอร์ต่างๆของ Monte Carlo สามารถที่จะใช้ SetOption และก็เรียกใช้ด้วย GetOption ได้

- "MCChartEquityCurves" (true/false)
- "MCStrawBroomLines" (0..100)
- "MCPosSizePctEquity" (0..100)
- "MCPosSizeMethod" - 0 - don't change, 1 - fixed size, 2 - constant amount, 3 - percent of equity (ความหมายเหมือนตอนตั้งค่า)
- "MCPosSizeShares" (ตัวเลข),
- "MCPosSizeValue" (ตัวเลข)
- "MCPosSizePctEquity" (ตัวเลข)

จะเพิ่มตัวบ่งชี้ที่กำหนดขึ้นเองในการทำ Monte Carlo ใน backtest Report ได้อย่างไร? (How to add custom metric based on MC test distribution(s) to the backtest report ?)

นอกไปจาก Monte Carlo report ตามปกติแล้ว คุณสามารถที่จะสร้างตัวบ่งชี้ต่างด้วยคุณเองโดยใช้ฟังก์ชั่น GetMonteCarloSim() ถ้าคุณสร้างตัวบ่งชี้ใหม่ขึ้นมาเอง โปรด ศึกษาเพิ่มเติมในส่วนของ "How to add custom metrics to backtester report" เป็นอันดับแรก

การจำลองด้วย Monte Carlo นั้นมีฟังก์ชันเดียวคือ GetValue("field", percentile) ที่ใช้ในการเข้าถึงค่า CDF ที่จำแนกได้ดังต่อไปนี้

- "FinalEquity"
- "CAR"
- "LowestEquity"
- "MaxDrawdown"
- "MaxPercDrawdown"

ทีนี้เราจะพาไปดูตัวอย่างของโค้ดเบื้องต้น ที่เพิ่มรายงานในส่วนของระดับ เปอร์เซ็นต์ไทด์ที่ 30 ลงไป โดยประกอบด้วย FinalEquity และ CAR

```
SetOption( "MCEnable", True );
SetOption( "MCRuns", 1000 );
SetCustomBacktestProc( "" );

if( Status( "action" ) == actionPortfolio )
{
	bo = GetBacktesterObject();

	bo.Backtest(); // รันค่า default ของกระบวนการ backtest

	// เข้าสู่กระบวนการแสดงผลลัพธ์ของ Monte Carlo
	// note 1: มันอาจจะ NULL ถ้า Monte Carlo ไม่ได้ถูกเรียกใช้
	// note 2: ผลลัพธ์ Monte Carlo จะแสดงผลหลังจากการ Backtest() หรือ PostProcess
	// เมื่อ การจำลองแบบ Monte Carlo เสร็จกระบวนการในขั้นตอนสุดท้าย
	 mc = bo.GetMonteCarloSim();

	if( mc )
	{
		// ใช้ระดับเปอร์เซ็นต์ไทด์ที่ 30 ของ final equity และ การกระจายตัวของ CAR
		 bo.AddCustomMetric( "FinalEq30", mc.GetValue( "FinalEquity", 30 ) );
		bo.AddCustomMetric( "CAR30", mc.GetValue( "CAR", 30 ) );

		// คุณสามารถที่จะรวมค่าทางสถิติของ Monte Carlo ด้วยค่าทางสถิติปกติได้
		 st = bo.GetPerformanceStats(0);
```

```
            bo.AddCustomMetric( "CAR30/MDD", mc.GetValue( "CAR", 30 ) /
            st.GetValue( "MaxSystemDrawdownPercent" ) );
        }
    }
```

เมื่อเราเพิ่มตัวบ่งชี้ใหม่เข้าไปในระบบแล้ว เราก็สามารถที่จะใช้ตัวบ่งชี้ใหม่นี้ เพื่อเป็นเป้าหมายในการ Optimization ได้เช่นกัน (อย่าลืมที่จะเปลี่ยน MCEnable เป็นเลข 2) และ ใช้ในกระบวนการ Walk Forward เพื่อเป็น objective function ได้อีกด้วย เพื่อที่จะเลือกตัวบ่งชี้ที่จะทำการ optimization คุณจึงใส่ชื่อตามประเภทที่คุณต้องการลงใน AddCustomMetric ในช่อง "Optimization Target" บนกล่องการตั้งค่าในหน้า Walk Forward วิธีนี้จะทำให้คุณสามารถรัน optimization / walk forward โดยตรงได้จากการจำลองการกระจายตัวด้วยวิธี Monte Carlo ดังนั้น ยกตัวอย่างเช่น แทนที่คุณจะใช้ CAR/MDD คุณสามารถที่จะใช้ CAR30/MDD (ระดับเปอร์เซ็นต์ไทด์ ที่ 30 ของ Monte Carlo CAR เปรียบเทียบกับ max. system drawdown)

จะใช้การสุ่มแบบ Monte Carlo แทนการใช้การทดสอบแบบ bootstrap ได้อย่างไร? (How about Monte Carlo randomization instead of bootstrap test?)

การสุ่มด้วยวิธี Monte Carlo นั้น แตกต่างจากการทดสอบแบบ bootstrap เพราะว่า มันไม่ได้ใช้ trade list ตัวจริง (realized) จากการ backtest โดยตรง แต่พยายามที่จะใช้ "ผลลัพธ์แต่ละอันของข้อมูล realized หรือ ข้อมูลเชิงสมมุติฐาน" ยกตัวอย่างเช่น เมื่อระบบเทรดเกิดสัญญาณในการซื้อจำนวนมากกว่าที่เราจะสามารถซื้อได้ เราจะจะเลือก ว่าการเทรดไหนควรจะเทรด หรือการเทรดไหนที่เราจะข้ามไป โดยปกติแล้ว การเลือกนี้จะอยู่ในส่วนของตัวแปร PositionScore ในโปรแกรม Amibroker จะเป็นตัวบอกตัวไหนที่ควรจะเทรดมากกว่ากัน ในกระบวนการทดสอบแบบสุ่มนั้น แทนที่เราจะใช้การพิจารณาจาก PositionScore คุณจะได้การสุ่มมาแทน ถ้าหากเกิดจำนวนสัญญาณมากกว่าที่จะเข้าเทรดได้ กระบวนการในขั้นตอนนี้จะสุ่มเพื่อเลือกการเทรดที่จะเข้าเทรด ตอนนี้การใช้ Optimize() ฟังก์ชัน และ การสุ่ม PositionScore เราสามารถที่จะรัน เป็นพันๆครั้งได้เพื่อทำการสุ่มในขั้นตอนการทดสอบด้วย Monte Carlo randomization :

```
step = Optimize( "step", 1, 1, 1000, 1 ); // 1000 backtests
```

```
//ด้วยการสุ่มเลือกการเทรดจากข้อมูลทั้งหมดที่เรามีอยู่ (โปรดทำให้แน่ใจด้วยว่าคุณรันบท watch
lists ที่มีจำนวนรายชื่อมากๆ)
PositionScore = mtRandom();
```

การทดสอบด้วยการสุ่ม มีข้อเสียเปรียบใหญ่ๆอยู่ข้อหนึ่ง: คือมันไม่สามารถใช้ได้ในหลายๆกรณี เมื่อระบบไม่สร้างจำนวนสัญญาณที่มากเพียงพอในแต่ละ bar ซึ่งมันก็จะไม่มากเพียงพอที่จะถูกเลือก ที่สำคัญยิ่งไปกว่านั้นคือ วิธีการสุ่มแบบ Monte Carlo จะไปขัดกับสมมุติฐานที่ว่าโอกาสในการเทนดแต่ละครั้งนั้นมีโอกาสเท่ากัน เพราะในหลายๆกรณีมันไม่เป็นอย่างนั้น บ่อยครั้งที่ระบบการเทรดที่เฉพาะเจาะจงของเรากำหนดให้เลือกการเทรดในหลายๆโอกาสจาก การเรียงลำดับ หรือ การให้คะแนน เมื่อระบบใช้ score (เรียงลำดับ) เป็นเหมือนแก่นหลักของระบบ (เช่นระบบที่ใช้การ rotation) ถ้าหากคุณแทนที่การใช้ score ด้วยการสุ่ม มันคือการที่คุณกำลังทดสอบ white noise(ถ้าจะให้แปล คำจำกัดความของมันคือ เสียงที่เราใช้กลบเกลื่อนเสียงอื่น อาจจะเป็นเสียงพัดลม หรือเสียงทีวี ซึ่งจะเป็นเสียงที่ดังสม่ำเสมอ) เท่านั้นไม่ใช่การทดสอบระบบ

Pyramiding (scaling in/out) และ การปรับใช้อัตรตราแลกเปลี่ยนค่าเงินในการ backtest พอร์ทการลงทุน

(Pyramiding (scaling in/out) and mutliple currencies in the portfolio backtester)

สำคัญ : โปรดแน่ใจว่าคุณได้อ่านหัวข้อ Backtesting your trading ideas article และ Portfolio Backtesting จบแล้ว

เริ่มต้นจาก เวอร์ชั่น 4.70 ผู้ทำการ backtest นั้นสามารถที่จะ position scaling และ ใช้สกุลเงินหลายสกุลได้

หมายเหตุ ฟังก์ชันสูงนี้จะทำงานเพื่อสนับสนุน ผู้ทำการ backtest เป็นพอร์ทการลงทุนเท่านั้น ฟังก์ชัน Old single-security backtester และ single-security equity() function นั้นไม่สนับสนุนการใช้ฟังก์ชั่นนี้

Pyramiding / Scaling

การใช้ sigScaleIn / sigScaleOut ถูกนำเข้ามาเพื่อตอบสนองความต้องการของผู้ใช้เมื่อต้องการที่จะ scale-in/out (จากผู้แปล : การทำ Pyramiding คือการซื้อ/ขาย หลายๆครั้งไม่ได้จบแค่เพียงครั้งเดียว หรือที่เราชอบเรียกว่าการเข้าซื้อขายหลายๆไม้นั้นล่ะครับ)

สิ่งที่คุณต้องทำทั้งหมดในการทำ Pyramiding คือ

- ใช้ sigScaleIn เพื่อสร้างตัวแปรในการ BUY/SHORT ถ้าคุณต้องการที่จะ scale-in (เพิ่มขนาดของ size) LONG/SHORT
- ใช้ sigScaleOut เพื่อสร้างตัวแปรในการ BUY/SHORT ถ้าคุณต้องการที่จะ scale-out (ลดขนาดของ size) LONG/SHORT

การ Scaling size ถูกกำหนดด้วยตัวแปร PositionSize ในกรณีที่ว่าเราไม่ได้ทำการ ซื้อ/ขาย ด้วยเงินที่เท่าเดิม แต่ใช้เงินที่เพิ่มขึ้น หรือ ลดลง

สำคัญมาก: โปรดทราบว่าในการ backtest นั้นจะถือว่าการ scale-in/out ในแต่ละครั้งนั้นนับเป็นการเทรดหนึ่งครั้ง (จะแสดงให้เห็นแต่ละครั้งของการเทรดใน trade list) ข้อแตกต่างเพียงหนึ่งเดียวเมื่อเปรียบ เทียบกับการเทรดธรรมดานั่นคือ มันจะคำนวณค่าเฉลี่ยของ ราคาเข้าซื้อ (และ อัตราเฉลี่ยของคู่เงิน) คิดมาจาก ราคาที่เข้าซื้อบางส่วน และ ค่าเฉลี่ยของ ราคาขาย (และ อัตราเฉลี่ยของคู่เงิน) คิดมาจาก ราคาที่ขายออกบางส่วน นอกจากนี้มันจะ แสดงถึงราคาซื้อ/ขาย โดยเฉลี่ยอีกด้วย ส่วนค่า commission แน่นอนว่าจะคำนวณอย่างถูกต้องตามจำนวนที่แบ่งในการ เข้าซื้อ/ขาย จริง

หากคุณต้องการที่จะทราบรายละเอียดเกี่ยวกับการ scaling คุณต้องรัน backtest ด้วยโหมด "DETAIL LOG" มันเป็นทางเดียวเท่านั้นที่คุณจะสามารถเห็นว่า การทำงานของการ scaling-in /out มันทำงานอย่างไร และ ราคาเฉลี่ยนั้นมันคำนวณออกมาได้อย่างไร

โปรดทราบ ว่าการ scaling-in/-out และ multiple-currency นั้นรับรองเฉพาะ portfolio backtest การ backtest แบบเก่า เช่น การใช้ Equity() ฟังก์ชันจะไม่สามารถใช้กับการ scaling-in/out หรือการ multiple currencies ได้ (ระบบจะไม่สนใจการใช้คำสั่ง scaling นั้นๆ)

ตัวอย่าง อย่างง่าย (Easy examples)

ตัวอย่างที่ 1: dollar-cost averaging (แต่ละเดือนซื้อหุ้นด้วยจำนวนเงินที่เท่าๆกัน) (Example 1: dollar-cost averaging (each month you buy stocks for fixed dollar amount))

```
FixedDollarAmount = 500;
MonthBegin = Month() != Ref( Month(), -1 );
FirstPurchase = Cum( MonthBegin ) == 1;
Buy = IIf( FirstPurchase, 1, // True (or 1) เป็นตัวแทนของสัญญาณการซื้อ  IIf( MonthBegin,
sigScaleIn, // ในแต่ละเดือเพิ่มขนาดของ position
0 ) ); // ไม่มีสัญญาณ

Sell = 0; // เราจะไม่ทำการขาย

PositionSize = FixedDollarAmount;
```

ตัวอย่างที่ 2: dollar-cost averaging (เป็นโค้ดอย่างง่าย เนื่องจาก amibroker ถือว่าหากมีสัญญาณครั้งแรกให้ซื้ออยู่แล้ว) (Example 2: dollar-cost averaging (simplified formula because AB treats first sigScaleIn as buy anyway))

```
FixedDollarAmount = 500;
MonthBegin = Month() != Ref( Month(), -1 );
FirstPurchase = Cum( MonthBegin ) == 1;
Buy = IIf( MonthBegin, sigScaleIn, 0 ); //ในแต่ละเดือเพิ่มขนาดของ position

Sell = 0; //เราจะไม่ทำการขาย
```

```
PositionSize = FixedDollarAmount;
```

ตัวอย่างที่ 3: เพิ่มขนาดของ position เมื่อเทรดแล้วได้กำไร โดยปราศจากการทำ Pyramiding เพิ่มขนาดเมื่อกำไรมากกว่า 5% และลดขนาดเมื่อ ขาดทุนมากกว่า -5% (Example 3: increasing position when profit generated by trade without pyramiding becomes greater than 5% and decreasing position when loss is greater than -5%)

```
// %การลงทุนเปลี่ยนแปลงในการทำ Pyramiding
PyramidThreshold = 5;

// เงื่อนไขการเทรดเบื้องต้น (ไม่ทำ Pyramiding)
Buy = Cross( MACD(), Signal() );
Sell = Cross( Signal(), MACD() );

e = Equity(1); // สร้างเงินลงทุนโดยปราศจากผลกระทบจาก Pyramiding
 PcntProfit = 100 * ( e - ValueWhen( Buy, e ) )/ValueWhen( Buy, e );
InTrade = Flip( Buy, Sell);
// ExRem ถูกใช้เพื่อให้แน่ใจในการเกิด scaling-in/out
// เกิดการเข้าเพียงแค่ครั้งเดียว
DoScaleIn = ExRem( InTrade AND PcntProfit > PyramidThreshold, Sell );
DoScaleOut = ExRem( InTrade AND PcntProfit < -PyramidThreshold, Sell );

// ปรับเปลี่ยนกฎเพื่อจัดการพีระมิด
Buy = Buy + sigScaleIn * DoScaleIn + sigScaleOut * DoScaleOut;
```

```
PositionSize = IIf( DoScaleOut, 500, 1000 ); // ทำการเข้าและ scale-in ขนาด $1000,
scale-out ขนาด: $500
```

ตัวอย่างที่ 4: การออกบางส่วน (scaling out) เมื่อถึงเป้ากำไรที่เรากำหนด (Example 4: partial exit (scaling out) on profit target stops)

ใน code จะเป็นการยกตัวอย่าง ออก 50% ในเป้ากำไรแรก และ 50% ในเป้ากำไรถัดไป และออกทั้งหมดที่เหลือเมื่อหลุด trailing stop

```
Buy = Cross( MA( C, 10 ), MA( C, 50 ) );
Sell = 0;

// ระบบจะออกเมื่อ
// 50% ของ position ถ้าหากถึงเป้ากำไรแรก
// 50% ของ position ถ้าหากถึงเป้ากำไรที่สอง
// 100% ของ position ถ้าหากหลุด TRAILING STOP

FirstProfitTarget = 10; // กำไร SecondProfitTarget = 20; // เป็น %
TrailingStop = 10; // % เช่นกัน

priceatbuy=0; highsincebuy = 0;

exit = 0;

for( i = 0; i < BarCount; i++ )
{
if( priceatbuy == 0 AND Buy[ i ] )
```

```
{
	priceatbuy = BuyPrice[ i ];
}

if( priceatbuy > 0 )
{
	highsincebuy = Max( High[ i ], highsincebuy );

	if( exit == 0 AND
	High[ i ] >= ( 1 + FirstProfitTarget * 0.01 ) * priceatbuy )
	{
		// ชนเป้ากำไรแรก - scale-out exit = 1;
		Buy[ i ] = sigScaleOut;
	}

	if( exit == 1 AND
	High[ i ] >= ( 1 + SecondProfitTarget * 0.01 ) * priceatbuy )
	{
		// ชนเป้ากำไรที่สอง - ให้ออก
		exit = 2;
		SellPrice[ i ] = Max( Open[ i ], ( 1 + SecondProfitTarget * 0.01 ) *
		priceatbuy );
	}

	if( Low[ i ] <= ( 1 - TrailingStop * 0.01 ) * highsincebuy )
	{
		// หลุด trailing stop - ให้ออก
		exit = 3;
```

```
                SellPrice[ i ] = Min( Open[ i ], ( 1 - TrailingStop * 0.01 ) * highsincebuy );
        }



        if( exit >= 2 )
        {
                Buy[ i ] = 0;
                Sell[ i ] = exit + 1; // มาร์คการออกที่เหมาะสม
                exit = 0;
                priceatbuy = 0; //รีเซ็ทราคาใหม่
                highsincebuy = 0;
        }
}
}

SetPositionSize( 50, spsPercentOfEquity );
SetPositionSize( 50, spsPercentOfPosition * ( Buy == sigScaleOut ) ); // scale out 50%
```

ของ position

Mulitple Currency Support

ในการ backt portfolio นั้นผู้ใช้สามารถที่จะ backtest ระบบที่มี สินค้าที่มี สกุลเงินที่แตกต่างกันได้ นั่นรวมถึงความสามารถในการใช้อัตราการแลกเปลี่ยนสกุลเงินในอดีตมาคำนวณได้อีกด้วย ซึ่งอัตราการแลกเปลี่ยน จะถูกกำหนดไว้ในหน้า "Currencies" ที่อยู่ในส่วน preferences อัตราแลกเปลี่ยนที่ใช้ใน สินค้าใดๆก็ตามสามารถที่จะกำหนดได้โดยไปที่

Symbol -> Information page

ในหน้าของ "Currencies" ที่อยู่ใน Preferences - สามารถที่จะกำหนด ค่าสกุลเงินพื้นฐาน และอัตราแลกเปลี่ยน (คงที่ หรือ ไม่คงที่) สำหรับสกุลเงินที่แตกต่างกัน สิ่งนี้จะช่วยให้ผลลัพธ์ของการ backtest มีความถูกต้อง เมื่อเราต้องการที่จะทดสอบ สินค้าที่มี สกุลเงินที่แตกต่างกัน

แล้ว Amibroker จะรู้ได้อย่างไรว่าเมื่อ ไหร่เราต้องการ ให้มัน คงที่ หรือ ไม่คงที่? เงื่อนไขดังต่อไปนี้ที่โปรแกรมต้องการจะเป็นตัวกำหนด ในการใช้ค่าของสกุลเงินต่างๆ

a) Symbol->Information, "Currency" ให้ใส่ค่าของสกุลเงินที่แตกต่างจากฐานข้อมูลเดิมลงไป

b) ค่าสกุลเงินที่เหมาะสม (จะถูกกำหนดใน Symbol) จะถูกจับคู่ โดย Preferences -> ในหน้าของ Currencies

c) การใช้อัตราแลกเปลี่ยนที่ไม่คงที่ "FX SYMBOL" จะถูกกำหนดใน preferences EXISTS ในฐานข้อมูลของคุณ และให้ค่าที่เหมาะสมในแต่ละวันภายใต้ช่วงเวลาของการวิเคราะห์ นั้นๆ

อะไรคือ "INVERSE" check box ใน preferences? เรามาดูตัวอย่างด้วย EURUSD กันดีกว่า

เมื่อ "USD" เป็น สกุลเงินหลักของคุณและอัตราแลกเปลี่ยนของ EUR เป็นแบบ "เส้นตรง" คู่เงิน EURUSD (เช่น 1.3) แต่เมื่อ "EUR" เป็นสกุลเงินหลักของคุณ อัตราแลกเปลี่ยนของ USD จะกลายเป็น INVERSE ของ EURUSD (เช่น 1/1.3) ในตรงกันข้ามมันจะเป็นจริงกับ คู่เงินเช่น USDJPY (ที่เป็น "inverse" เรียบร้อยแล้ว)

การเขียนโค้ดเพื่อตั้งการแจ้งเตือน

(Using formula-based alerts)

บทนำ(Introduction)

บนการใช้งาน AmiBroker นั้น คุณสามารที่จะกำหนดให้ระบบตั้งแจ้งเตือนได้ เมื่อสัญญาณที่เรากำหนดไว้ถูกจับได้เมื่อไหร่ มันสามารถที่จะแสดงการแจ้งเตือนออกมาในรูปแบบที่เป็น ข้อความ , เสียงเตือน , e-mail แจ้งเตือน และ การส่งออกข้อมูลภายนอกต่างๆที่รองรับการทำงานได้

ทั้งหมดนี้จะทำงานบนฟังก์ชัน AlertIF

โดย default แล้ว การแจ้งเตือนทั้งหมดจะสร้างข้อความแจ้งเตือนไว้บน หน้าต่าง Output เพื่อแสดงหน้าต่างดังกล่าว คุณจำเป็นที่จะต้องไปที่ Window -> Alert Output

นอกจากนี้ยังมีหน้าต่าง Easy Alerts ซึ่งจะทำให้คุณกำหนดการตั้งการแจ้งเตือนอย่าง่าย ที่คุณไม่ต้องทำการเขียนโค้ดใดๆเลย (แต่ไม่แนะนำสำหรับ ผู้ที่ต้องการจะใช้ระบบการแจ้งเตือนแบบเต็มรูปแบบด้วย AlertIf ฟังก์ชัน)

การตั้งค่าต่างๆ (Settings)

Alert - สัมพันธ์การตั้งค่าในแถบ "Alerts" ใน Tools-> หน้าต่าง Preferences

เราสามารถที่จะกำหนด e-mail account , เสียงตอนแจ้งเตือน และ กำหนดว่าส่วนใดของ AmiBroker สามารถที่จะสร้างการแจ้งเตือนได้โดย AlertIF ฟังก์ชัน

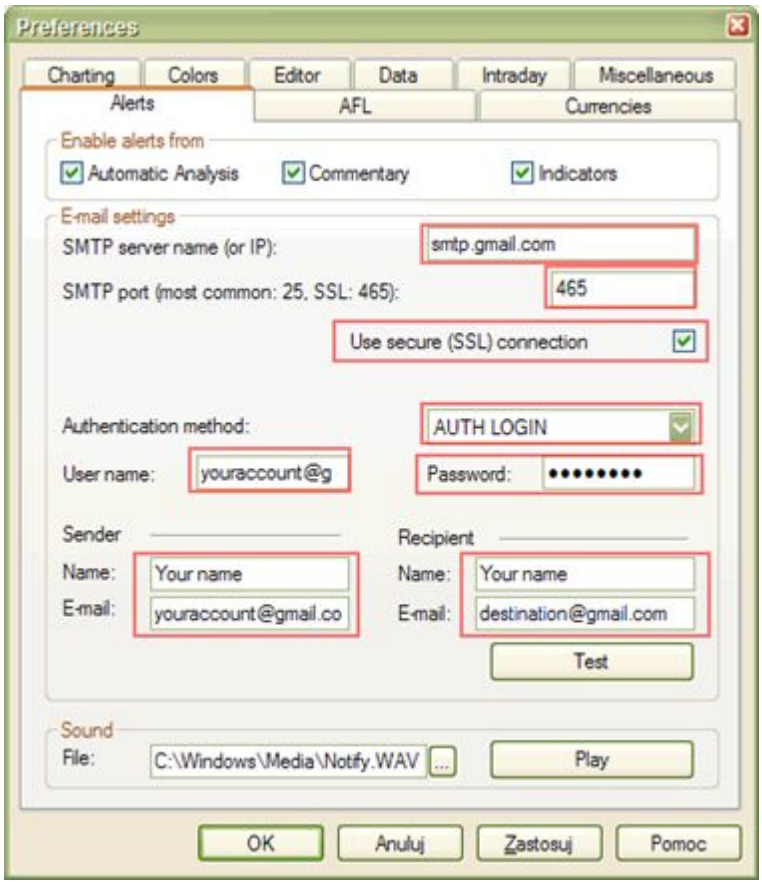
หน้าของการตั้งค่า E-mail ตอนนี้สามารถที่จะเลือกวิธีการต่างที่ได้รับความนิยมได้ อย่างเช่น : AUTH LOGIN (นิยมมากที่สุด), POP3-before-SMPT (ได้รับความนิยม), CRAM-MD5, LOGIN PLAIN

"Enable alerts from" การติ๊กเครื่องหมายถูกในส่วนนี้ จะทำให้คุณ เลือกการตั้งเตือนให้ เปิด/ปิด ตามอัตโนมัติ, หรือตาม ความคิดเห็น/การตีความ และ ตัวชี้วัดที่เรากำหนดเอง

หน้าต่างที่แสดงผลการแจ้งเตือนนั้น รูปแบบตอนนี้จะแสดงเป็น colum ที่แสดงการแจ้งเตือน ถ้ามันเป็นแบบอัตโนมัติ ,ความคิดเห็น/การตีความ หรือ ตัวชี้วัดที่เรากำหนดเอง มันจะง่ายต่อการค้นหาว่าส่วนใดของ Amibroker ที่ทำการแจ้งเตือนนั้นๆ

สำหรับ AmiBroker 5.30 ขึ้นไป - สนับสนุนกับ SSL (secure connection) ซึ่งเอาไว้ใช้กับ GMail ยกตัวอย่างเช่น เพื่อที่จะเปิดการใช้งานการสนับสนุนด้วย SSL support คุณจำเป็นต้องทำตามขั้นตอนดังต่อไปนี้ :

1. Download และ run SSL add-on 0kd http://www.amibroker.com/bin/SSLAddOn.exe
2. กำหนดค่า(Tools->Preferences->Alerts) ด้วยการเปิดใช้ SSL ที่แสดงด่านล่าง



ฟังก์ชัน AlertIF (AlertIF function)

AlertIF ฟังก์ชัน จะมีลักษณะคล้ายคลึงกับ WriteIF แต่แทนที่มันจะเขียนเฉพาะตัวหนังสือในหน้าต่างแสดงการแจ้งเตือนแล้วนั้น มันยังสามารถที่จะ :

- กำหนดข้อความที่เรากำหนดขึ้นเองให้ไปแสดงผลที่ หน้าต่างการแสดงผลลัพธ์การแจ้งเตือนได้
- กำหนดเสียง (เป็นเพียงแค่เสียงปี๊ปจากคอมพิวเตอร์ หรือจากไฟล์สกุล .WAV)
- ส่ง e-mail
- เปิดใช้การเชื่อมต่อกับแอปพลิชั่นอื่นๆ

มี syntax ตามดังต่อไปนี้ :

AlertIf(BOOLEAN_EXPRESSION, command, text, type = 0, flags = 1+2+4+8, lookback = 1);

1. BOOLEAN_EXPRESSION คือการแสดงออกมาว่าถ้ามันวัดผลแล้วเป็น True (ไม่ได้มีค่าเป็นศูนย์) จะทำให้เกิดการแจ้งเตือน และ ถ้าวัดผลออกมาแล้วค่าเป็น False (มีค่าเป็นศูนย์) จะไม่มีการแจ้งเตือนใดๆ เกิดขึ้น หมายเหตุ จะมีเพียงแค่การมองย้อนกลับไปมากที่สุดเพียงแค่ไม่กี่ bar ล่าสุดเท่านั้น ที่จะถูกนำมาพิจารณา

2. command (string) เป็นตัวกำหนดการกระทำเมื่อเกิดการแจ้งเตือนเกิดขึ้น ถ้าหากปล่อยค่าเว้นไว้ การแจ้งเตือนก็จะแสดงไปที่หน้าต่างการแจ้งเตือนตามปกติ (Window->Alert Output) อย่างอื่นที่ช่วยสนับสนุนค่าในการใช้ command มีดังนี้:

SOUND the-path-to-the-WAV-file
EMAIL
EXEC the-path-to-the-file-or-URL <optional args>

คำสั่ง SOUND จะเล่นเสียงจากไฟล์สกุล WAV 1 ไฟล์

คำสั่ง EMAIL จะส่งคำสั่งไปที่ e-mail ที่ทำการระบุไว้ตอนที่เราตั้งค่าก่อนหน้านี้ (Tools -> Preferences -> E-mail) รูปแบบของการตั้ง e-mail มีดังต่อไปนี้:

Subject: Alert type_name (type) Ticker on Date/Time Body: text

คำสั่ง EXEC จะดำเนินการใช้แอปพลิเคชั่นภายนอก , ไฟล์, หรือ URL ที่เฉพาะเจาะจงหลังจากใช้ EXEC <optional args> จะถูกแนบมาหลังจาก ชื่อไฟล์ และ ตัวหนังสือถูกแนบมาด้วยในตอนสุดท้าย

3. Text จะกำหนดว่าตัวหนังสือที่เป็นผลลัพธ์นั้นจะถูกแสดงบนหน้าต่างของการแจ้งเตือน หรือจะถูกส่งไปยัง e-mail หรือ แอปพลิเคชั่นภายนอกที่ถูกระบุไว้โดยคำสั่ง EXEC

4. Type จะเป็นตัวกำหนดประเภทของการแจ้งเตือน ก่อนการกำหนดไว้ค่า types จะเป็น 0 - default, 1 - buy, 2 - sell, 3 - short, 4- cover โดยคุณสามารถที่จะระบุค่าที่สูงมากกว่านี้ (เช่น 5 6 7 8) และ ระบบจะส่งผลลัพธ์ออกมาเป็น other แทน

5. Flags จะเป็นตัวควบคุมพฤติกรรมของ AlertIF ฟังก์ชันได้ มันจะประกอบไปด้วยการทำงานรวมตัวกันของคำสั่งต่างๆ โดยมีค่าดังนี้:

(1 - หมายถึงส่งข้อความไปยังหน้าต่างแสดงการแจ้งเตือน, 2 - สร้างเสียงปี๊ป (บนคอมพิวเตอร์), 4 - ไม่แสดงการแจ้งเตือนใดๆซ้ำอีก หากการแจ้งเตือนมี type ที่เหมือนเดิม, 8 - ไม่แสดงการแจ้งเตือนใดๆซ้ำ หากมี date/time ที่เหมือนเดิม) โดย default ทางเลือกต่างๆจะถูกเปิดไว้ทั้งหมด

6. lookback เป็นพารามิเตอร์ที่คอยคุมว่า จะดู bar ย้อนหลังกี่ bar เพื่อนำมาพิจารณาในการแจ้งเตือน ยกตัวอย่างเช่น :

```
Buy = Cross( MACD(), Signal() );
Sell = Cross( Signal(), MACD() );
Short = Sell;
Cover = Buy;

AlertIF( Buy, "EMAIL", "A sample alert on "+FullName(), 1 );

AlertIF( Sell, "SOUND C:\\Windows\\Media\\Ding.wav", "Audio alert", 2 );
 AlertIF( Short, "EXEC Calc.exe", "Launching external application", 3 );
AlertIF( Cover, "", "Simple text alert", 4 );
```

หมายเหตุ : คำสั่ง EXEC จะใช้ ShellExecute ฟังก์ชัน และใช้ได้ทั้งไฟล์ EXE และ URLs เช่นกัน

หมายเหตุ (Notes)

1. โปรดทราบไว้ว่า โดย default แล้ว AlertIf ฟังก์ชัน จะไม่สร้างสัญญาณที่ซ้ำกัน เมื่อการสแกนรันหลายๆครั้ง ในระหว่างการทดลอง คุณอาจจะชอบที่จะให้เกิดการทำซ้ำของสัญญาณ ในการเกิดสัญญาณที่ตามๆกันมา (เหมือนค่อยเน้นย้ำ เตือนเรา) เพื่อทำแบบนั้นแล้วคุณต้อง เปลี่ยน ค่า default ของ flags เป็น 1 + 2 :

```
AlertIF( condition, "", "Text", 1, 1+2 );
```

2. ถ้าคุณต้องการที่จะสร้างการแจ้งเตือนเฉพาะ bar ที่จบ bar แล้ว คุณจำเป็นที่จะเพิ่ม โค้ดนี่ลงไป :

```
barcomplete = BarIndex() < LastValue(BarIndex());


AlertIF( barcomplete AND condition, "", "Text", 1 );
```

การเปิดใช้หน้าต่างการตีความ

(Using interpretation window)

โปรดทราบ: คุณควรที่จะอ่าน หัวข้อ How to write your own chart commentary มาก่อนบทความนี้

หน้าต่างการตีความ (Interpretation window) (Window->Interpretation) จะแสดง chart-sensitive และ เพื่อที่จะเพิ่ม การการตีความนี้ คุณแค่เพียงใช้ Formula Editor และเพิ่มโค้ดลงไป หลังจาก โค้ดสำหรับตัวบ่งชี้(indicator) โปรดทราบไว้ว่า เพื่อให้ได้ศักยภาพที่ดีที่สุด คุณควรที่จะใช้คำสั่ง เงื่อนไขที่ทำให้มั่นใจว่า การตีความนี้นั้น จะแสดงเฉพาะใน "commentary" โหมด เท่านั้น

```
if( Status("action") == actionCommentary )
{
// printf statements here....
}
```

ตัวอย่าง:

```
Plot( Close, "Price", -1, 64 );
Plot( SAR( Prefs( 50 ), Prefs( 51 ) ), "SAR",-17, 8+16 );

if( Status("action") == actionCommentary )
{
printf("The Parabolic SAR provides excellent exit points. \n");
 printf("You should Close long positions when the price falls below\n");
```

```
 printf("the SAR AND Close Short positions when the price rises above the
SAR.\n");
printf( WriteIf( Graph1 > Close, "SAR is above close", "SAR is below close" ) );
}
```

การสนับสนุนการใช้ Multiple Time Frame ด้วย AFL

(Multiple Time Frame support in AFL)

ปล่อยออกมาตอนเวอร์ชัน 4.41 นำไปพาไปสู่ความสามารถในการใช้ multiple time frames (ช่วงเวลาต่างๆในหน่วย bar) ในสูตรเดียว ฟังก์ชัน time frame สามารถที่แบ่งการทำงานได้ออกเป็น 3 กลุ่มด้วยกัน :

1. เปลี่ยนกรอบเวลาของตัว O, H, L, C, V, OI, Avg arrays: TimeFrameSet, TimeFrameRestore
2. การบีบ/ขยาย arrays ใดๆ ไปยังช่วงเวลาที่เฉพาะเจาะจง: TimeFrameCompress, TimeFrameExpand
3. เข้าถึง price/volume arrays ได้ทันที่ใน time frame ที่แตกต่างกัน: TimeFrameGetPrice

กลุ่มแรก: ถูกใช้เมื่อคุณต้องการดำเนินการคำนวณบางอย่างบน time frame ที่แตกต่างกัน (ไม่ได้ใช้ Time Frame เดียว) ยกตัวอย่างเช่น หากคุณต้องการที่จะคำนวณ 13-bar moving average บนข้อมูลใน TimeFrame 5 นาที และ 9 bar exponential avarage บนข้อมูลใน TimeFrame 1 ชั่วโมง ในขณะที่คุณกำลังแสดงผลกราฟเป็นข้อมูลมูล 1 นาทีอยู่ คุณควรจะเขียนโค้ดดังนี้:

```
TimeFrameSet( in5Minute ); // เพื่อเปลี่ยนเป็น Time Frame 5 นาที

/* ตอนนี้ MA จะทำงานอยู่บนข้อมูล 5 นารที , ma5_13 จะถูกกำหนดให้คำนวณจากข้อมูล 13 bar
และเป็น MA ของข้อมูล 5 นาที */

ma5_13 = MA( C, 13 );
```

```
TimeFrameRestore(); // เพื่อคืนค่าให้กลับไปใช้ Time Frame ดั้งเดิม
 TimeFrameSet( inHourly ); // เพื่อเปลี่ยนเป็น  Time Frame 1 ชั่วโมง
mah_9 = EMA( C, 9 ); // 9 bar moving average จากข้อมูล 1 ชั่วโมง
TimeFrameRestore(); // เพื่อคืนค่าให้กลับไปใช้ Time Frame ดั้งเดิม
Plot( Close, "Price", colorWhite, styleCandle );

//สร้างเส้นค่าเฉลี่ยใน Time Frame ที่กำหนด

Plot( TimeFrameExpand( ma5_13, in5Minute), "13 bar moving average from 5 min
bars", colorRed );
Plot( TimeFrameExpand( mah_9, inHourly), "9 bar moving average from hourly bars",
colorRed );
```

TimeFrameSet(ใส่ช่วงระยะเวลาที่เราต้องการลงไป) - เป็นการแทนที่ price/volume arrays (open, high, low, close, volume, openint, avg) ด้วยการระบบช่วงเวลที่เราต้องการจะเปลี่ยน และเพื่อที่จะเปลี่ยนไปยัง Time Frame ดังเดิมของเรา เราต้องใช้ TimeFrameRestore() ฟังก์ชัน ถ้าหลังจากนั้นคุณต้องการที่จะเปลี่ยน Time Frame อีก คุณก็แค่เพียงเรียกใช้ TimeFrameSet อีกครั้งหนึ่ง แต่ก่อนที่คุณจะเรียกใช้ใหม่คุณจำเป็นต้อง คืนค่าไปยัง Time Frame ดังเดิมเสียก่อน โดยใช้ TimeFrameRestore() หมายเหตุ : ช่วงระยะเวลานั้นจะถูกกำหนดด้วย Time Frame ในหน่วยวินาที

ยกตัวอย่างเช่น: 60 ก็จะหมายถึง 1 นาที คุณควรที่จะใช้ค่าคงที่เพื่อความสะดวก สำหรับช่วงเวลาทั่วๆไป โดยใช้ คำสั่งเหล่านี้: in1Minute, in5Minute, in15Minute, inHourly, inDaily, inWeekly, inMonthly

ตั้งแต่เวอร์ชั่น 4.70 คุณสามารที่จะระบุช่วงเวลาให้เป็น N-tick ได้ ซึ่งเราสามารถทำได้โดยเพียงใส่เครื่องหมายลบไปด้านหน้าระยะเวลาที่เรากำหนด เช่น -5 ก็จะหมายถึง 5-tick bar และ -133 ก็จะหมายถึง 133-tick bar นอกจากนี้ โปรดทราบไว้ว่าการใช้ N-tick นั้นจะใช้ได้ก็ต่อเมื่อ database ของคุณ ใช้ Tick เป็นฐานข้อมูลหลัก สามารตั้งค่าได้ใน File -> Database Settings

```
TimeFrameSet( -133 ); // เปลี่ยนไปใช้ 133-tick
```

TimeFrameRestore() - เพื่อคืนค่าของ arrays ราคา ที่ถูกเปลี่ยนไปก่อนหน้านี้ด้วย TimeFrameSet() หมายเหตุ นี่เป็นค่าเฉพาะ OHLC, V, OI และ Avg เท่านั้น ตัวแปรอย่างอื่นที่ถูกสร้างขึ้นเมื่ออยู่ใน time frame ที่แตกต่างกันก็ยังคงโดนบีบอัดอยู่ เพื่อยกเลิกการบีบอัดนั้น คุณจำเป็นที่จะต้องใช้ TimeFrameExpand (จากผู้แปล : คำว่าบีบอัดและขยายในที่นี้ หมายถึง เวลาที่คุณมี Time Frame ใหญ่ หากคุณต้องการที่จะเห็นรายละเอียดเพิ่มเติมใน Time Frame ที่เล็กลงคุณก็จำเป็นต้อง “ขยาย” Time Frame ที่ใหญ่อยู่ ในตอนแรก เช่นเดียวกับกรณีบีบอัด Time Frame ในการเปลี่ยนจาก Time Frame ที่มีขนาดเล็กไปยัง Time Frame ที่มีขนาดใหญ่)

เมื่อคุณเปลี่ยน time frame ไปครั้งนึงแล้วด้วยการใช้ TimeFrameSet ฟังก์ชัน AFL ทั้งหมดจะทำงานบน time frame ที่คุณเลือก จนกระทั่งคุณเปลี่ยนกลับไปใช้ time frame ดั้งเดิม โดยการใช้ TimeFrameRestore หรือ ใช้การเปลี่ยน time frame อีกครั้งหนึ่งด้วยการใช้ TimeFrameSet ซึ่งเราแนะนำว่า มันจะเป็นการดีมาก หากคุณเรียกใช้ TimeFrameRestore ทุกครั้งหลังจากที่คุณจบการใช้งาน time frame ที่คุณเลือกแล้ว

เมื่อ time frame ถูกเปลี่ยนไปยัง time frame อื่นที่ไม่ใช่ time frame ดังเดิม ผลลัพธ์ของฟังก์ชั่นทุกอันหลังจากที่คุณเรียกใช้ฟังก์ชัน TimeFrameSet ก็จะถูกเปลี่ยนไปด้วยเช่นกัน ถ้าหากคุณต้องการที่จให้มันใช้ Time Frame ดั้งเดิมของมันคุณจำเป็นที่จะต้อง ขยาย (expand) พวกมันในภายหลัง ตัวแปรต่างๆที่สร้างขึ้นก่อนที่จะเรียกใช้ TimeFrameSet() ก็จะยังคงอยู่ใน time frame ที่มันถูกสร้างขึ้น การกระทำต่างๆเหล่านี้สามารถทำได้อย่างไม่มีขีดจำกัดในหนึ่ง formula

หมายเหตุ : การที่คุณจะบีบอัดข้อมูล คุณทำได้แค่จาก Time Frame ที่เล็ก ไป Time Frame ที่เล็กที่ใหญ่กว่าได้เท่านั้น ดังนั้นหากคุณใช้งาน ข้อมูล 1 นาที คุณสามารที่จะบีบอัดข้อมูลเป็น 2, 3, 4, 5, 6, ...N-นาที ได้ แต่เมื่อคุณทำงานบนข้อมูล 15 นาที คุณไม่สามารที่จะใช้ข้อมูล 1 นาทีได้ นอกจากนี้ก็จะมีความคล้ายคลึงกันในการใช้งาน หากข้อมูลของคุณมีแค่เพียง ข้อูล EOD คุณก็จะไม่สามารถเข้าถึงข้อมูล intraday time frames ได้

กลุ่มที่สอง: TimeFrameCompress/TimeFrameExpand สามารที่จะบีบอัด หรือ ขยาย array เดี่ยวๆไปยัง Time Frame ที่แตกต่างกันได้ โดยเฉพาะอย่างยิ่งคุณควรที่จะทราบไว้อย่างยิ่งว่า ใน

TimeFrameExpand ถูกใช้เพื่อคลายการบีบอัดตัวของตัวแปร array ที่ถูกสร้างขึ้นใน time frame ที่แตกต่างกัน การคลายการบีบอัดตัว จำเป็นที่จะต้องบอกถึง array ที่ถูกสร้างขึ้นใน time frame ที่แตกต่างกัน ยกตัวอย่างเช่น หากคุณต้องการที่จะใช้ weekly moving average คุณต้องทำการ ขยาย(expand) มัน ดังนั้นข้อมูลของหนึ่ง weekly จะครอบคลุม bars 5 วัน (Monday-Friday) ของสัปดาห์ที่สอดคล้องกัน

TimeFrameExpand(array, interval, mode = expandLast) การขยาย array ของจาก 'interval' time frame ไปสู่ time frame หลัก ('interval' จะต้องสอดคล้องกับค่าที่ถูกใช้ใน TimeFrameCompress หรือ TimeFrameSet)

mode ที่สามารใช้ได้ :

expandLast - ค่าที่ถูกบีบอัด จะถูกขยายโดยการเริ่มต้นจาก bar สุดท้าย ในช่วงระยะเวลาที่เรากำหนด(ยกตัวอย่างเช่น weekly close/high/low จะใช้บน bar ของวันศุกร์)

expandFirst - ค่าที่ถูกบีบอัด จะถูกขยายโดยการเริ่มต้นจาก bar แรก ในช่วงระยะเวลาที่เรากำหนด(ยกตัวอย่างเช่น weekly close/high/low จะใช้บน bar ของวันจันทร์)

expandPoint - array ผลลัพธ์ที่ได้นั้นจะไม่มีค่าที่ว่างเปล่าเลยโดยนับจากช่วงระยะเวลาที่เรากำหนด (เอา bar ทุก bar ที่ไม่เป็น Null (ไม่มีค่า)).

Caveat: expandFirst ใช้เมื่อ price แตกต่างจากราคาเปิด ซึ่งบางทีอาจจะเป็นการมองไปในอนาคต ยกตัวอย่างเช่น หากคุณสร้าง ราคา High ที่เป็น weekly แล้วทำการขยายมันเป็นระยะเวลาที่เป็น daily โดยใช้ expandFirst จะช่วยให้คุณรู้ในวันจันทร์ว่า high ของทั้งสัปดาห์อยู่ตรงไหน

TimeFrameCompress ถูกออกแบบมาอย่างสมบูรณ์แบบ มันสามารที่จะใช้เมื่อคุณต้องการที่จะขยายช่วงเวลาของ array หนึ่งๆ โดยปราศจากการส่งผลใดๆต่อ OHLC,V arrays หากคุณเรียกใช้ TimeFrameCompress มันจะไม่ส่งผลใดๆต่อผลลัพธ์ของ ฟังก์ชั่นอื่นๆเช่นกัน

wc = TimeFrameCompress(Close, inWeekly);

/* ตอนนี้ time frame ยังคงไม่ถูกเปลี่ยน (ยังเป็น daily อยู่) และ MA จะทำงานบน ข้อมูลที่เป็น daily */

```
dailyma = MA( C, 14 );

/* แต่ถ้าเราเรียก MA บนcompressed array มันจะเป็น MA ที่เป็น time frame อื่น
*/

weeklyma = MA( wc, 14 ); // โปรดทราบว่านี้คือ time-compressed array
Plot( dailyma, "DailyMA", colorRed );
weeklyma = TimeFrameExpand( weeklyma, inWeekly ); // ได้ถูกขยายช่วงเวลเรียบร้อยแล้ว
 Plot( weeklyma, "WeeklyMA", colorBlue );
```

ใน formula นี้ time frame จะยังคงอยู่ที่การตั้งค่าดังเดิม มันจะเปลี่ยน time frame แค่ array เดียว

```
TimeFrameCompress( array, interval, mode = compressLast )
- การบีบอัด array เดียวนั้นสามารทำได้โดยใช้ compression mode ดังต่อไปนี้: compressLast -
ใช้ ค่าสุดท้ายของ array (close) ในช่วงเวลานั้น
compressOpen - ใช้ ค่า open array ในช่วงเวลานั้น
compressHigh - ใช้ ค่าสูงที่สุดของ array ในช่วงเวลานั้น compressLow -ใช้ ค่าต่ำที่สุดของ array
ในช่วงเวลานั้น compressVolume - ใช้ ค่ารวม array ในช่วงเวลานั้น

Graph0 = TimeFrameExpand( TimeFrameCompress( Close, inWeekly, compressLast ),
inWeekly, expandLast );
Graph1 = TimeFrameExpand( TimeFrameCompress( Open, inWeekly, compressOpen
),
inWeekly, expandFirst );
```

กลุ่มที่สาม ประกอบด้วยเพียงแค่หนึ่งฟังก์ชันที่มีประโยชน์มาก: TimeFrameGetPrice จะทำให้การอ้างอิง price และ volume จาก time frames โดยปราศจาก การเปลี่ยน /การบีบอัด/การขยาย time frame ใดๆอีกเลย แค่หนึ่งฟังก์ชันเท่านั้น เพื่อดึง price จาก time frame ที่ใหญ่กว่า และมันยังไม่เพียงแค่อ้างอิงจากปัจจุบัน แต่มันยังสามารถที่จะอ้างอิง bar ในอดีตใน time frame ที่แตกต่างกันได้อีกด้วย

TimeFrameGetPrice(pricefield, interval, shift = 0, mode = expandFirst);

- อ้างอิง OHLCV จาก time frame อื่นๆมันสามารถทำงานได้ทันทีโดยปราศจากการเรียกใช้ TimeFrameSet ฟังก์ชัน

Price field สามารใช้เป็นอันใดอันหนึ่งจาก : "O", "H", "L", "C", "V", "I" (open interest)

Interval คือ ช่วงเวลาในหน่วยวินาที

shift คือการอ้างอิงจากในอดีต (ค่าติดลบ) และค่าในอนาคตคือ future (ค่าเป็นบวก) จากข้อมูลใน time frame ที่ใหญ่ขึ้น เช่น -1 คือใช้ข้อมูลจาก bar ก่อนหน้า (เหมือนกับฟังก์ชัน Ref แต่มันใช้งานใน time frame ที่ใหญ่ขึ้น)

ตัวอย่าง:

TimeFrameGetPrice("O", inWeekly, -1) // เพื่อสร้างราคา Open -ของ 1 week ก่อนหน้า
TimeFrameGetPrice("C", inWeekly, -3) // เพื่อสร้างราคา Close ของ 3 week ก่อนหน้า
TimeFrameGetPrice("H", inWeekly, -2) // เพื่อสร้างราคา High ของ 2 week ก่อนหน้า
TimeFrameGetPrice("O", inWeekly, 0) // เพื่อสร้างราคา Open ของ week ปัจจุบัน
TimeFrameGetPrice("H", inDaily, -1) // เพื่อสร้างราคา High ของ 1 Day ก่อนหน้า
เมื่อใช้ในข้อมูล intraday

Shift ทำงานเหมือน Ref() ฟังก์ชัน แต่มันเป็นการประยุกต์บนการบีบอัด time frame

หมายเหตุ: ฟังก์ชั่นเหล่านี้ทำงานเหมือนกับ3 ฟังก์ชั่นที่ซ้อนกันเหล่านี้

TimeFrameExpand(Ref(TimeFrameCompress(array, interval, compress(ขึ้นอยู่กับ field ที่ใช้)), shift), interval, expandFirst)

ดังนั้นหาก shift = 0 การบีบอัดข้อมูลอาจจะเป็นการมองไปในอนาคต (high ของ week สามารถที่จะรู้ได้ในวันจันทร์) หากคุณต้องการที่จะเขียนระบบการเทรด โดยใช้ฟังก์ชั่นนี้โปรดทำให้แน่ใจว่าคุณอ้างอิงข้อมูลในอดีต ด้วยการใช้ ค่า shift ที่ติดลบ

สิ่งเดียวที่แตกต่างกันของของการใช้ TimeFrameGetPrice คือมันจะเร็วกว่า 2 เท่า แทนการใช้ Expand/Compress ปกติ

หมายเหตุเกี่ยวกับศักยภาพของ TimeFrame ฟังก์ชัน :

a) การวัดค่าจะทำบน Athlon 1.46GHz, 18500 daily bars จะถูกบีบอัดให้เป็น weekly time frame

TimeFrameGetPrice("C", inWeekly, 0) - 0.0098 sec (9.8 milliseconds)

TimeFrameSet(inWeekly) - 0.012 sec (12 milliseconds)

TimeFrameRestore( ) - 0.006 sec (6 milliseconds)

TimeFrameCompress(Close, inWeekly, compressLast); - 0.0097 sec (9.7 milliseconds)

TimeFrameExpand(array, inWeekly, expandLast); - 0.0098 sec (9.8 milliseconds)

b) การวัดค่าจะทำบน 1.46GHz, 1000 daily bars จะถูกบีบอัดให้เป็น weekly time frameall functions below 0.0007 sec (0.7 millisecond)

มันทำงานได้อย่างไร ? (How does it work internally ?)

Time-frame ฟังก์ชัน ไม่ได้ทำการเปลี่ยน BarCount - พวกมันแค่ บีบ arrays ดังนั้น คุณจะมี N-bars อันแรกที่เป็น NULL (แท่งแรกไม่มีค่า) และจากนั้น ส่วนสุดท้ายของ array จะยังคงใช้ time frame ในช่วงระยะเวลาที่ถูกบีบอัดไว้อยู่ ดังนั้นนี้จึงเป็นเหตุผล หลักที่เราจะต้องขยายข้อมูลให้กลับไปยังข้อมูลที่เป็นระยะเวลาดั้งเดิมของมันด้วย TimeFrameExpand นี่คือการตรวจสอบอย่างง่ายๆ (ใช้การ exploration) โดย

แสดงว่ามันจะเกิดอะไรขึ้นหลังจากที่คุณเปลี่ยนไปใช้ time frame ที่ใหญ่ขึ้น โดยการรัน Exploration บน current symbol, all quotations,ตั้งค่าช่วงเวลาเป็น daily และคุณจะเห็นว่า "ราคา close ที่ถูกบีบอัดใน weekly เป็นอย่างไร" column จะประกอบด้วยค่าที่ไม่มีค่า จากจุดเริ่มต้น และ ถูกบีบอัดข้อมูลเป็น weekly ในตอนสุดท้ายของ array

```
Filter = 1;
AddColumn(Close, "Daily close");

TimeFrameSet(inWeekly);
AddColumn(wc = Close, "weekly close compressed"); TimeFrameRestore();

AddColumn( TimeFrameExpand(wc, inWeekly), "weekly close expanded");
```

ตัวอย่าง

ตัวอย่างที่ 1: สร้างกราฟ weekly MACD และสร้างลูกศรเมื่อเกิดการ cross กัน จากข้อมูล daily

```
TimeFrameSet( inWeekly );
m = MACD(12, 26 ); // MACD จากข้อมูล  WEEKLY
TimeFrameRestore();
m1 = TimeFrameExpand( m, inWeekly ); Plot( m1, "Weekly MACD", colorRed );
PlotShapes( Cross( m1, 0 ) * shapeUpArrow, colorGreen );
PlotShapes( Cross( 0, m1 ) * shapeDownArrow, colorGreen );
```

ตัวอย่างที่ 2: สร้างกราฟ weekly candlestick วางทับบนเส้นราคาของข้อมูล daily

```
wo = TimeFrameGetPrice( "O", inWeekly, 0, expandPoint );
wh = TimeFrameGetPrice( "H", inWeekly, 0, expandPoint );
```

```
wl = TimeFrameGetPrice( "L", inWeekly, 0, expandPoint );
wc = TimeFrameGetPrice( "C", inWeekly, 0, expandPoint );

PlotOHLC( wo, wh, wl, wc, "Weekly Close", colorWhite, styleCandle ); Plot( Close,
"Daily Close", colorBlue );
```

ตัวอย่างที่ 3: ระบบแบบ 3 หน้าจอ

```
/* เปลี่ยนไปใช้ weekly time frame */ TimeFrameSet( inWeekly );
whist = MACD( 12, 26 ) - Signal( 12, 26, 9 );
wtrend = ROC( whist, 1 ); // weekly trend - การเปลี่ยนแปลงของ macd histogram ใน 1
week
TimeFrameRestore();

/* ขยายการคำนวณ  MACD ไปสู่ daily เพื่อที่เราจะสามารที่จะใช้สัญญาณต่างที่เกิดขึ้นจากข้อมูล
daily ได้ */

wtrend = TimeFrameExpand( wtrend, inWeekly );

/* elder ray */
bullpower= High - EMA(Close,13); bearpower= Low - EMA(Close,13);

Buy = wtrend > 0 /* หน้าจอแรก: weekly trend ที่เป็นบวก */
AND
bearpower < 0 AND bearpower > Ref( bearpower, -1 ) /* หน้าจอที่สอง : bear power
negative แต่เพิ่มขึ้น */
AND
H > Ref( H, -1 ); /* หน้าจอที่สาม: ถ้าหาก price ทำ new high*/
```

```
BuyPrice = Ref( H, -1 ); // buy stop level;

Sell = 0 ; // exit เฉพาะ by stops
ApplyStop( stopTypeProfit, stopModePercent, 30, True );
ApplyStop( stopTypeTrailing, stopModePercent, 20, True );
```

ฟังก์ชั่นการจัดอันดับ

(Ranking functionality)

การจัดอันดับนั้น คือ ความสัมพันธ์ระหว่างชุดของข้อมูล สำหรับข้อมูลสองสิ่งขึ้นไป

แบบแรกคือการ “จัดอันดับที่สูงกว่า”, “จัดอันดับที่ต่ำกว่า” , “จัดอันดับที่สูงกว่า” และ “จัดอันดับที่เท่ากัน”

แบบที่สองคือ วิธีที่ง่ายที่สุด คือการจัดอันดับโดยการเรียงลำดับ 'value' หรือ 'score' ยกตัวอย่างเช่น คุณสามารถที่จะใช้ 100-bar rate of change ของ symbol ใดๆ เพื่อเป็น 'score' หรือ 'value ของคุณ จากนั้นทำการเรียงลำดับผลลัพธ์เหล่านั้น คุณก็จะได้ symbol list ที่ตัวแรกคือ ตัวที่มีศักยภาพที่ดีที่สุด (rate of change สูงที่สุด) และ ตัวสุดท้ายของ list คือ ตัวที่มีศักยภาพต่ำที่สุด

AmiBroker มีรูปแบบให้ผู้ใช้เลือกใช้ในการจัดอันดับ อยู่ 3 รูปแบบด้วยกัน

1.ใช้การจัดอันดับจากสัญญาณการเทรด (buy/short) เพื่อตัดสินใจ ว่าตัวไหนน่าสนใจกว่าตัวอื่นๆ ในขณะที่ทำการ backtest portfolios หรือ optimization
2.ทำการจัดอันดับ โดยเลือกได้หลายๆตัวแปรในการจัดอันดับด้วยการใช้ Exploration
3.สร้างการจัดอันดับเป็นตัวเลขเพื่อใช้ในภายหลังขึ้นมา(ฟังก์ชั่นเพื่อใช้งานทั่วไป)

การจัดอันดับแบบแรก นั้นจะถูกคำนวณอัตโนมัติ หากระบบเทรดของคุณมีการกำหนดตัวแปร PositionScore คุณสามารถที่จะใช้ตัวแปร PositionScore ในการตัดสินใจว่าการเทรดครั้งไหน ควรที่จะเข้ามากกว่ากันเมื่อ เราไม่สามารถทำการเข้าซื้อได้ในทุกๆสัญญาณที่เข้ามา ในกรณีนี้ AmiBroker จะใช้ค่า absolute ของตัวแปร PositionScore ในการตัดสินใจ สำหรับรายละเอียดของ ฟังก์ชันในการจัดอันดับ คุณสามารอ่านได้ใน Portfolio Backtester tutorial

การจัดอันดับแบบที่สอง เป็นเพียงแค่การกำหนดตัวเลขขึ้นมาตัวเลขนึงเพื่อการจัดอันดับ โดยการใช้ผลลัพธ์จากการ explore colum ที่มีการจัดอันดับนั้นสามารถทำได้โดยใช้ฟังก์ชัน AddRankColumn หลัง

จากนั้นใช้ SetSortColumns ฟังก์ชันเพื่อการเรียงลำดับ คุณสามารถที่จะเรียกใช้ SetSortColumns ได้หลายครั้ง และคุณก็สามารถที่จะใช้ AddRankColumn เพื่อสร้างหลายๆ colum เพื่อการเรียงลำดับที่หลากหลายได้มากยิ่งขึ้นอีกด้วย สามารถดูตัวอย่างได้จากด้านล่าง:

```
Filter = 1;
AddColumn( Close, "Close" );
 AddColumn( Volume, "BI" );
AddSummaryRows( 31 + 32, 1.5 );


AddRankColumn(); // ไม่ต้องทำการเรียงลำดับก่อน AddRankColumn แค่เพียงเพิ่ม line
number
เท่านั้น

SetSortColumns( -4 );
AddRankColumn(); // การจัดอันดับตาม Colum ที่ 4 (จากน้อยไปมาก)
SetSortColumns( -3 );
AddRankColumn(); // การจัดอันดับตาม Colum ที่ 3 (จากมากไปน้อย)
```

การจัดอันดับแบบที่สาม คือการจัดอันด้วยด้วยวัตถุประสงค์โดยทั่วๆไป โดยใช้การจัดอันดับแบบ bar ต่อ bar ด้วยข้อมูลจาก ตัวแปรแบบตัว(static variables) ซึ่งมันตัวแปรที่ต้องการข้อมูลมากที่สุด แต่ก็ให้ความเป็นไปได้มากที่สุดเช่นกัน โดยทั่วไปแล้วการที่จะสร้างตัวแปรแบบที่คงตัวจะถูกใช้เพื่อ เรียงลำดับหรือจัดอันดับ (sorting/ranking) ยกตัวอย่างเช่น การสร้างตัวแปรที่เป็น "scores" และหลังจากนั้นก็เรียงมันด้วย ฟังก์ชันพิเศษ (StaticVarGenerateRanks) ซึ่งมันจะ ทำการสร้างชุดของตัวแปรแบบคงตัวใหม่ขึ้นมา เพื่อใช้ในการจัดอันดับ

หมายเหตุ: ฟังก์ชันนี้ไม่ได้มีจุดประสงค์ในการแทนที่การจัดอันดับผ่าน PositionScore

ในทางตรงกันข้ามเลย หากคุณสามารที่จะใช้ PositionScore คุณก็ควรที่จะเลือกใช้ PositionScore เนื่องจากความเร็วในการประมวลผลของมันมากกว่า และใช้เนื้อที่ความจำที่น้อยกว่าอีกด้วย มันเป็นหนทางที่จะทำให้การจัดอันดับในการทำ backtest ของคุณมีประสิทธิภาพอย่างมาก

StaticVarGenerateRanks
โดยทั่วไปแล้วมันถูกสร้างขึ้นมาเพื่อใช้ในงานอื่นๆมากกว่าเอาไว้ใช้เพื่อการ backtest เช่นใช้ในการ explorations หรือใน indicators ที่อาจจะต้องการฟังก์ชันของการจัดอันดับ แต่แน่นอนมันสามารถที่จะใช้กับการ backtest ได้เช่นกัน เมื่อ PositionScore เพียงอย่างเดียวไม่เพียงพอต่อสิ่งที่คุณต้องการในการสร้างระบบเทรดของคุณ

โปรดระวัง: ฟังก์ชันนี้ใช้การประมวลผลและหน่วยความจำที่ค่อนข้างมาก มันใช้เวลาถึง 20ms สำหรับ 15K bars และ 7 symbol พยายามที่จะเรียกใช้มันเพียงครั้งเดียวต่อการ scan/exploration/backtest การใช้ if(Status("stocknum")==0) หรือจะดีกว่าในใช้การทำงานที่แยกกันโดยการสแกนข้อมูลก่อนการ คำนวณ แล้วค่อยนำมาจัดลำลับในภายหลัง แต่ถ้าคุณทำมันไม่ได้ คุณสามารถที่จะเรียกใช้ฟังก์ชัน
StaticVarGenerateRanks ศักยภาพในการทำงานของโปรแกรมจะลดลงอย่างมีนัยยะสำคัญSเพราะไม่เพียงแต่ฟังก์ชันนี้จะใช้เวลามากในการทำงาน มันยังปิดกันการเข้าถึงการใช้งานฐานข้อมูลร่วมกันอีกด้วย นั่นหมายถึงการที่คุณจะเข้าถึงการใช้ตัวแปรแบบที่คงตัวนั้น คุณจำเป็นที่จะรอจนกระทั่งฟังก์ชันนี้ทำงาน เสร็จ

StaticVarGenerateRanks ฟังก์ชัน (StaticVarGenerateRanks function)

StaticVarGenarateRanks("outputprefix", "inputprefix", topranks, tiemode)
นี่เป็นองค์ประกอบหลักของฟังก์ชัน มันต้องใช้ 4 พารามิเตอร์ด้วยกัน :
ซึ่งเป็นแกนหลักของการใช้การจัดอันดับแบบที่ 3

"outputprefix" - คำนำหน้าที่จะนำไปต่อท้ายผลลัพธ์ตัวแปรแบบคงตัวของเรา(output)
ซึ่งนำไปใช้ในการจัดอันดับ
"inputprefix" คำนำหน้าของตัวแปรแบบคงตัวที่ใช้ในการทำ score (input)

topranks ใช้ในการกำหนดว่าจะมีจำนวนเท่าไร่ในการจัดอันดับ top/bottom
ซึ่งรวมถึงจำนวนในการจัดอันดับชุดข้อมูลด้วย
tiemode ใช้ในการกำหนด ความสัมพันธ์ (equal ranks) แบบไหนควรได้รับการแก้ไข

"inputprefix" คือตัวนำหน้าที่จะกำหนดชื่อของตัวแปรแบบคงตัว ที่จะถูกใช้ในการนำไปจัดอันดับ (ตัวที่จะนำไป input) AmiBroker จะทำการค้นหา ตัวแปรแบบคงตัวทั้งหมดที่เริ่มด้วยคำนำหน้านั้นๆ และสมมุติให้ว่ามันเป็นส่วนหนึ่งของ ชื่อหุ้นนั้นๆ ดังนั้น หากคุณต้องการที่จะ จัดอันดับโดยใช้ ROC (rate of change) สิ่งที่คุณต้องทำคือ การเก็บค่าของมันลงในตัวแปรแบบคงที่ ให้เราบอกว่าเราจะใช้ชื่อตัวแปรแบบคงตัว เช่น "ItemScoreAPPL", "ItemScoreMSFT" และอื่น ๆ

เพื่อที่จะสร้างตัวแปรแบบคงตัวคุณสามารใช้ loop นี้ได้

```
for ( i = 0; ( sym = StrExtract( symlist, i ) ) != ""; i++ )
{
SetForeign( sym ); Value = ROC( C, 10 );
RestorePriceArrays();
StaticVarSet( "ItemScore" + sym, Value );
}
```

ตอนนี้คุณสามารถที่จะเข้าสู่การ การเรียงลำดับ/การจัดอันดับได้แล้ว (sorting/ranking) พวกมันมี 2 โหมดคือ normal ranking mode และ Top/Bottom Rank mode วิธีแบบ Normal ranking mode จะถูกใช้เมื่อตัว toprank มีค่าตั้งแต่ 0

```
StaticVarGenerateRanks( "rank", "ItemScore", 0, 1224 );
```

ในกรณีนี้ StaticVarGenerateRanks จะถูกเรียกเพื่อสร้าง ชุดของตัวแปรแบบคงตัว ที่ขึ้นต้นด้วยคำนำหน้าซึ่ง กำหนดมาจากสอง argument แรก ที่เรากำหนดไว้ ดังนั้นในกรณีนี้ RankItemScoreMSFT

จะใช้ในการจัดอันดับของ MSFT, RankItemScoreAAPL จะใช้ในการจัดอันดับของ AAPL หมายเหตุ AmiBroker จะจัดอันดับโดยการเริ่มต้นจากเลขหนึ่ง

argument ที่สาม (topranks) ถ้าเราให้ค่าเป็น 0 หมายถึงเลือกใช้ normal ranking mode

argument ที่สี่ (tiemode) กำหนดความสัมพันธ์ในการจัดอันดับ

Supported modes คือ 1234 และ 1224 (1224 mode จะให้ความสัมพันธ์ของตัวแปรที่มีค่าเท่ากันด้วย)

code ตัวอย่าง สำหรับ normal ranking mode (ทุกอย่างที่ทำเสร็จแล้ว มันจะจบในการรันครั้งเดียว(สามารที่จะใช้ใน indicator ได้) :

```
 symlist = "C,CAT,DD,GE,IBM,INTC,MSFT";

// ลบตัวแปรแบบคงตัว
StaticVarRemove( "ItemScore*" );

// ป้อนข้อมูล  arrays ที่คงที่

for ( i = 0; ( sym = StrExtract( symlist, i ) ) != ""; i++ )
{
SetForeign( sym ); Value = ROC( C, 10 );
RestorePriceArrays();
StaticVarSet( "ItemScore" + sym, Value );
}

// ทำการจัดอันดับ
StaticVarGenerateRanks( "rank", "ItemScore", 0, 1224 ); // normal rank mode
```

```
// อ่านค่าการจัดอันดับ
for ( i = 0; ( sym = StrExtract( symlist, i ) ) != ""; i++ )
{
Plot( StaticVarGet( "RankItemScore" + sym ), sym, colorCustom10 + i );
}
```

Top/bottom ranking mode (คือ การสร้างตารางการจัดอันดับจาก top/bottom วิ่งตรึงค่าดัชนีไปไว้ด้านบนสุด เมื่อ topranks > 0 ค่าของ top ranked จะถูกใช้ และเมื่อ topranks < 0 ค่าของ bottom ranked จะถูกใช้ ค่าที่ถูกเก็บไว้ในตัวแปรมีรูปแบบดังนี้

OutputprefixInputprefixN เมื่อ N คือตัวเลข 1, 2, 3 ซึ่งเป็นตัวเลขที่แสดงถึงอันดับของ top/bottom ranks

ลองสมมุติว่า OutputPrefix นี้มีพารามิเตอร์คือ "Top" และ Inputprefix พารามิเตอร์ คือ ROC ในแต่ละอันตัวแปร ตามนี้ TopROC1 จะถูกตึงด้วยอันดับของค่า Top rated นอกจากนี้ TopROC2 จะถูกตรึงด้วย อันดับที่สอง ของค่าและ อื่นๆ

ฟังก์ชั่น StaticVarGenerateRanks ใช้เลขตำแหน่งที่เริ่มต้นจากที่หนึ่ง ใน top ranking mode StaticVarGenerateRanks

จะเตรียมตัวแปรแบบคงตัว ที่คั่นด้วยเครื่องหมาย “ , ” เพื่อแยก list ของชื่อตัวแปร ซึ่งมันสามารถที่จะใช้เพื่อค้นหาว่า ดัชนีตัวไหนอ้างอิงจาก symbol ตัวไหน ดังนั้นถ้า TopROC1 ถูกตึงด้วย 1 ที่คุณจะหา substring แรกในตัวแปร ในตัวแปร TopROCSymbols เพื่อหาว่าตัวแปรตัวไหน (symbol) ถูกจัดอันดับไว้บนสุด

นอกจากนี้ StaticVarGetRankedSymbols ให้วิธีการที่ง่ายต่อการใช้งาน เพื่อที่จะแยก list ของการจัดอันดับด้วยเครื่องหมาย “,” โดยเฉพาะอย่างยิ่งกับการใช้กับ วันที่ (datetime)

ตัวอย่างโค้ดใน top ranking mode :

```
symlist = "C,CAT,DD,GE,IBM,INTC,MSFT";

// ลบตัวแปรแบบคงตัว
 StaticVarRemove( "ItemScore*" );

// ป้อนข้อมูล  arrays ที่คงที่

for ( i = 0; ( sym = StrExtract( symlist, i ) ) != ""; i++ )
{
SetForeign( sym ); Value = ROC( C, 10 );
RestorePriceArrays();
StaticVarSet( "ItemScore" + sym, Value );



}

// ทำการจัดอันดับ
StaticVarGenerateRanks( "rank", "ItemScore", 0, 1224 ); // normal rank mode
StaticVarGenerateRanks( "top", "ItemScore", 3, 1224 ); // top-N mode
StaticVarGenerateRanks( "bot", "ItemScore", -3, 1224 ); // bottom-N mode
// อ่านค่าการจัดอันดับ
for ( i = 0; ( sym = StrExtract( symlist, i ) ) != ""; i++ )
{
Plot( StaticVarGet( "RankItemScore" + sym ), sym, colorCustom10 + i );
}

sdt = SelectedValue( DateTime() );
```

```
Title = "{{NAME}} -{{DATE}} - {{VALUES}} TOP: " + StaticVarGetRankedSymbols( "top",
"ItemScore", sdt ) +
" BOT: " + StaticVarGetRankedSymbols( "bot", "ItemScore", sdt ) ;
```

จะสามารถใช้ StaticVarGenerateRanks ใน Analysis window ได้อย่างไร (How to use StaticVarGenerateRanks in Analysis window)

เนื่องจากการจัดอันดับ เป็นกระบวนการที่มีความความต้องการของข้อมูลเป็นอย่างมาก ดังนั้นมันควรจึงทำงานแค่หนึ่งครั้งต่อการรัน Analysis หนึ่งครั้ง ไม่ใช่บนทุกๆ symbol คุณสามารถ ซึ่งคุณสามารถทำมันได้โดยการ รันแยกกันของ การใช้สูตรในการให้มันจัดอันดับก่อนที่จะให้มัน รัน Analysis หรือการใช้Status("stocknum") == 0 ที่เป็นคำสั่งที่จะให้คุณมันใจว่ากระบวนการจัดอันดับนั้นทำอยู่บนแค่ symbol แรกจาก watch list ภายใต้ analysis เท่านั้น

และนี่คือตัวอย่างของโค้ดสำหรับการ exploration ที่ใช้กับ active watch list หรือ symbol list ทั้งหมด และดำเนินการจัดอันดับ

```
 if ( GetOption( "ApplyTo" ) == 2 )
{
wlnum = GetOption( "FilterIncludeWatchlist" );
List = CategoryGetSymbols( categoryWatchlist, wlnum ) ;
}
else

if ( GetOption( "ApplyTo" ) == 0 )
{
List = CategoryGetSymbols( categoryAll, 0 );
}
else
{
```

```
        Error( "The formula works fine if your ApplyTo setting is 'Filter' or 'All'" );
}

if ( Status("stocknum") == 0 ) // สร้างการอันดับ เมื่ออยู่บน SYMBOL แรก

{
StaticVarRemove( "values*" );

for ( n = 0; ( Symbol = StrExtract( List, n ) )  != "";  n++         )
{
        SetForeign ( symbol ); values = RSI(); RestorePriceArrays();
        StaticVarSet (  "values"  +  symbol, values );
        _TRACE( symbol );
}

StaticVarGenerateRanks( "rank", "values", 0, 1224 );
}

symbol = Name();

values = StaticVarGet ( "values" +  symbol ); rank = StaticVarGet ( "rankvalues" +
symbol );

AddColumn ( values, "values" ); AddColumn ( rank, "rank" );


Filter = 1;
```

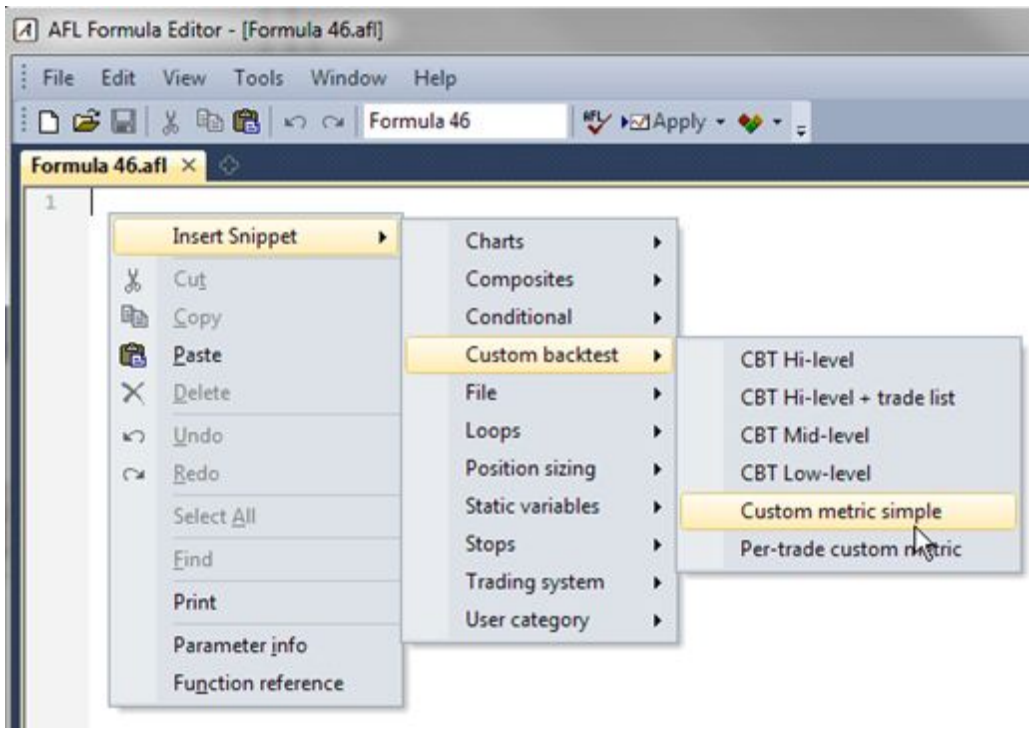
SetSortColumns(2, 4);

คีย์ลัดในโค้ด AFL

(Using AFL Code snippets)

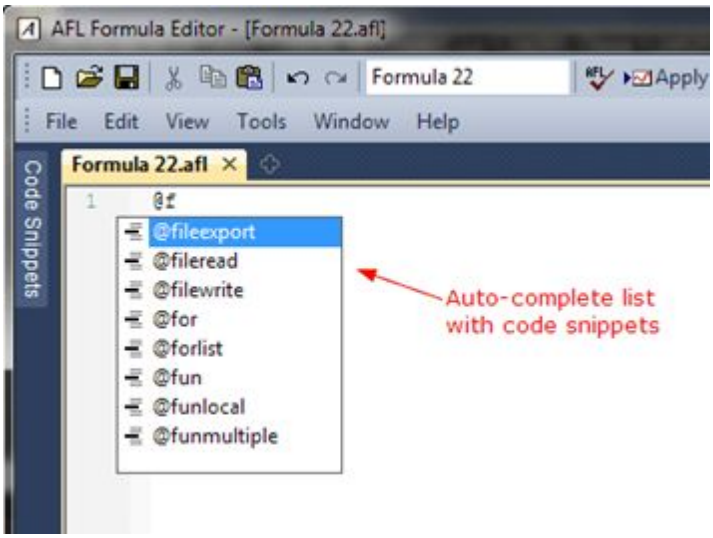
คีย์ลัดโค้ด นั้นเป็นส่วนเล็กๆส่วนหนึ่งของการใช้งานใหม่ๆบน โค้ด AFL มันสามารถที่จะนำเข้าได้โดย

- คลิกขวาบนหน้าต่าง AFL editorและเลือกเมนู "Insert Snippet" หรือ
- ลากคีย์ลัดจาก Snippet window
- หรือคีย์ลัดบนคีย์บอร์ด (เช่น @for)



ในเวอร์ชั่น 5.90 โค้ดคีย์ลัดสามารถทำงานได้อัตโนมัติบน list ใน the AFL Editor แค่เพียง ใช้ @ แล้วเขียนตัวอักษรแรกลงไป มันก็จะแสดง list ของคีย์ลัดที่คุณสามารใช้ได้

การเปลี่ยนคีย์บอร์ดเป็นอีกทางหนึ่งที่จะช่วยให้คุณใช้งานมันได้ โดยที่ไม่ต้องใช้งานฟังก์ชันด้านบน ดังนั้นแค่ @keytrigger และแทนทีมันด้วยคัยร์ลัด

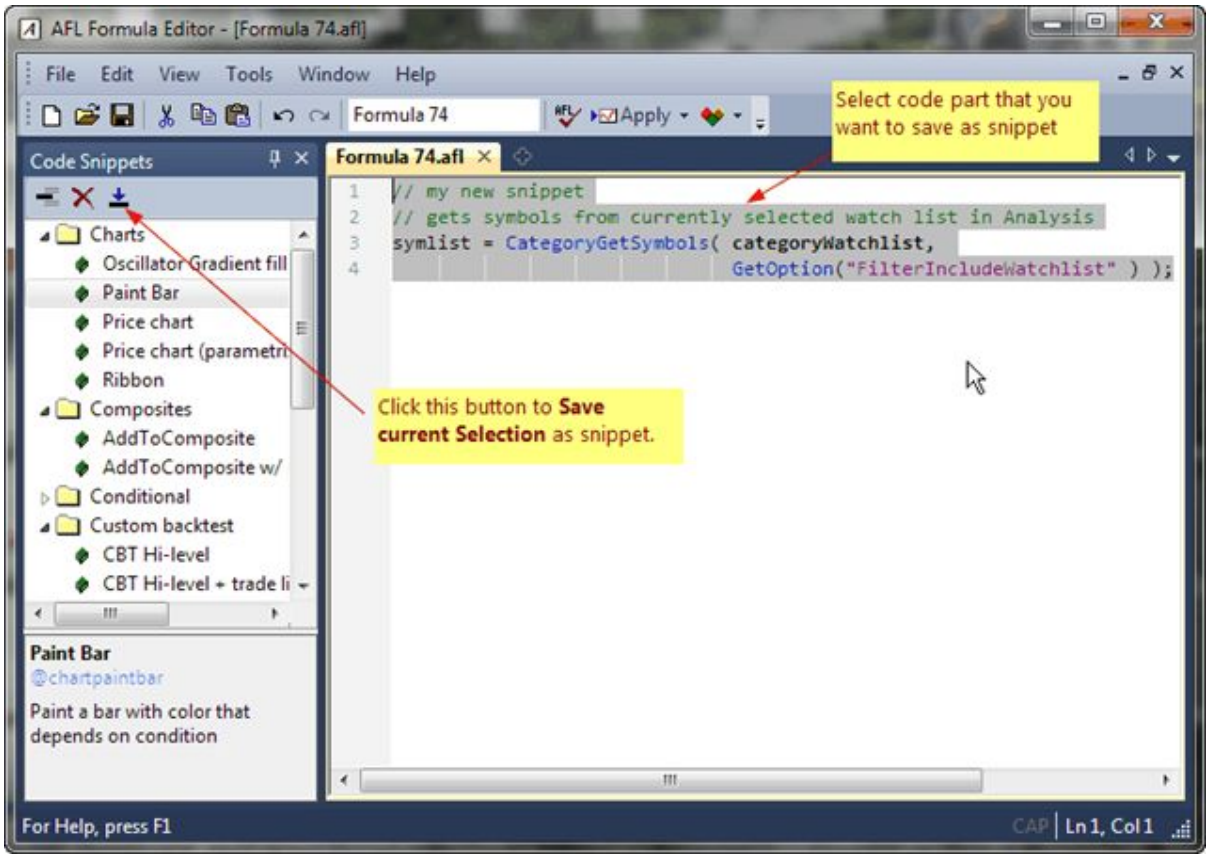


การกำหนดคีย์ลัดของตัวคุณเอง (DEFINING YOUR OWN SNIPPETS)

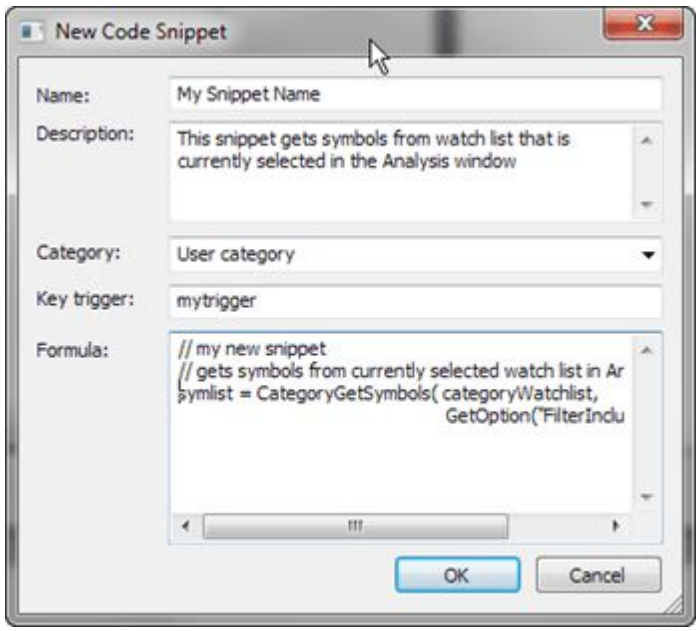
คุณสามารถที่จะเพิ่มคีย์ลัดของตัวคุณเองได้ง่ายๆ ด้วยการเขียนด้วยหน้าต่าง Code Snippet ซึ่ง Code Snippets window นั้นจะทำงานบนหน้า AFL editor มันสามารถที่จะแสดงบนหรือซ่อนไว้ ด้วยการใช้ Window menu

เพื่อที่จะสร้างคีย์ลัดด้วยตัวคุณเอง สิ่งที่คุณต้องทำก็คือ :

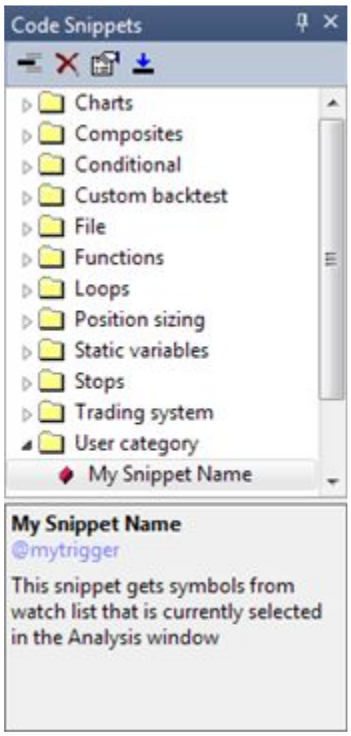
1. ประเภทของโค้ดที่คุณต้องการ
2. เลือก (mark) โค้ดที่คุณต้องการจะให้เป็นคีย์ลัด
3. กด Save selection ให้เป็นปุ่ม ในหน้าต่าง Code Snippets



ถ้าคุณทำตามขั้นตอนข้างต้นโต้แล้วกล่องโต้ตอบต่อนี้จะปรากฏขึ้น :



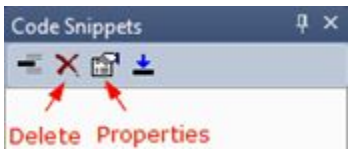
ตอนนี้คุณต้องใส่คำอธิบายของคีย์ต่างๆลัดลงไปในช่อง Name, Description และ Category ในส่วนของ Category สามารถเลือกได้จากรายการที่มีอยู่แล้ว (ใช้ drop down box) หรือ สร้าง category ขึ้นมาใหม่ โดยการพิมพ์ลงไปในช่อง category เลย Key trigger คือพื้นที่สำหรับตัวเลือกที่ประกอบด้วยข้อมูลโค้ดคีย์ลัดที่เป็นแบบอัตโนมัติ (ตามที่อธิบายไว้ข้างต้น) Formula คือพื้นที่ที่เอาไว้ใส่โค้ดคีย์ลัดของตัวมันเอง หลังจากที่เติมข้อมูลครบทุกช่องแล้วก็กด OK คีย์ลัดใหม่ของคุณก็จะปรากฏขึ้นมา



จากนั้นคุณสามารถใช้คีย์ลัดโค้ดของคุณเองแบบเดียวกับตัวอย่างที่มีอยู่ตามที่ได้อธิบายไว้แล้ว บางทีวิธีที่สะดวกที่สุดอาจจะเป็นการใช้การลากวางจาก list ไปยัง AFL editor

บางทีคุณอาจจะสังเกตเห็น user-defined snippets (ตัวคีย์ลัดที่เรากำหนดเอง) ถูกมาร์คไว้ด้วยกล่องสีแดงใน Code Snippets list มีแค่เพียง user-defined snippets ที่สามารถที่จะเขียนทับ หรือ ลบได้เท่านั้น

หากต้องการแก้ไขข้อมูล user-defined snippet ที่มีอยู่ที่ คุณสามารถทำตามขั้นตอนด้านบน และใช้ชื่อที่มีอยู่เดิมแล้ว Amibroker จะถามคุณหาก คุณต้องการที่จะ เขียนทับ คีย์ลัดที่มีอยู่เดิม หรือคุณแค่เพียงคลิกบนปุ่ม Properties และแก้ไขมัน โดยตรงได้เลย



เพื่อที่จะลบคีย์ลัด ให้เลือกคีย์ลัดที่คุณต้องการจะลบจาก list และกดปุ่ม Delete (X) ในหน้าต่าง Code Snippet

TECHNICAL INFO (สำหรับผู้ใช้ขั้นสูง) (TECHNICAL INFO (advanced users only))

มีสองไฟล์อยู่ใน Amibroker directory ที่เป็นคีย์ลัด :

CodeSnippets.xml - คีย์ลัดเหล่านี้จะมาพร้อมกับการติดตั้ง AmiBroker (และสามารถถูกทับได้โดยการติดตั้งใหม่ในภายหลัง ดังนั้น โปรดอย่าแก้ไขมัน)

UserSnippets.xml - นี่คือ user-definable snippets ไฟล์นี้ไม่ได้มาพร้อมในการติดตั้ง และผู้ใช้งานสามารถสร้างได้ด้วยตัวของพวกเขาเอง

โครงร่าง XML สำหรับไฟล์คีย์ลัด อย่างง่าย (ตามด้านล่าง) ฟังก์ชั่นทริกเกอร์ที่ยังไม่ได้ดำเนินการ

อย่างไรก็ตาม Keytrigger ควรรวมอยู่ในคำนิยามสำหรับใช้งานในอนาคต มันจะทำงานเหมือนกับการทำให้สมบูรณ์อัตโนมัติ แค่เพียงคุณใช้คีย์ลัด มันจะทำการ unfold formula นั้นๆ

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<AmiBroker-CodeSnippets CompactMode="0">

<Snippet>
<Name>First Snippet</Name>
```

```
<Description>Description of the snippet</Description>
<Category>User category</Category>
<KeyTrigger>?trigger1</KeyTrigger>
<Formula>
<![CDATA[

// the formula itself

]]>
</Formula>
</Snippet>

<Snippet>
<Name>Second Snippet</Name>
<Description>Description of the snippet</Description>
<Category>User category</Category>
<KeyTrigger>?trigger2</KeyTrigger>
<Formula>
<![CDATA[

// the formula itself

]]>
</Formula>
</Snippet>
</AmiBroker-CodeSnippets>
```

คู่มือการใช้งานแบบวิดีโอ (on-line)

(Video Tutorials (on-line))

เพื่อความสะดวกของคุณ เราได้เตรียมบทเรียนวิดีโอต่อไปนี้ (ในรูปแบบโปรแกรม Macromedia Flash) บนหน้าเว็บของเรา:

· How to install AmiBroker
· How to use drag-and-drop charting interface
· How to setup new database with eSignal RT feed (RT version)
· How to setup new database with IQFeed RT feed (RT version)
How to setup new database with Interactive Brokers (RT version)
· How to use AmiQuote in 'manual' mode
· How to use chart sheets and layouts
· How to use layers
· How to use AFL Code Wizard

สำหรับวิดีโอเพิ่มเติมโปรดเช็คที่: http://www.amibroker.com/support.html