HƯỚNG DẪN SỬ DỤNG
AMIBROKER 6.90
BẢN DỊCH TIẾNG VIỆT



MỘT SẢN PHẨM CỦA TEAM AmiX


DNSE
AmiX
PHẦN MỀM KẾT NỐI GIAO DỊCH AMIBROKER
CẬP NHẬT DỮ LIỆU
NHANH VÀ CHÍNH XÁC
KẾT NỐI
GIAO DỊCH TỰ ĐỘNG
LƯU TRỮ
DỮ LIỆU LỊCH SỬ
NHẬN THÔNG BÁO QUA
TIN NHẮN TELEGRAM

GIỚI THIỆU

Ami X

Ami X là phần mềm cung cấp giá chứng khoán Việt Nam realtime miễn phí cho Amibroker và giúp bạn chạy bot đặt lệnh giao dịch tự động tới tài khoản chứng khoán tại Công ty Chứng khoán DNSE.

Phần mềm Ami X được phát triển bởi Công ty Chứng khoán DNSE, và cung cấp hoàn toàn miễn phí cho khách hàng.



Tại sao nên sử dụng Ami X?

DỮ LIỆU NHANH, CHÍNH XÁC

Dữ liệu của Ami X được cung cấp từ nguồn của công ty chứng khoán DNSE. Việc sử dụng 1 nguồn chính thống thay vì lấy Data của 1 bên thứ 3 sẽ giúp Data có tốc độ nhanh nhất, tạo ra ưu thế trong giao dịch và đặt lệnh cho khách hàng.

TIẾT KIỆM CHI PHÍ

Để sử dụng Amibroker thì bạn sẽ cần mua dữ liệu chứng khoán từ một bên thứ ba. Các bên cung cấp Data trên thị trường cho Amibroker hiện nay đều thu phí từ 1.000.000 VNĐ - 3.000.000 VNĐ 1 năm. Việc sử dụng Ami X để lấy Data sẽ giúp bạn tiết kiệm được khoản chi phí này.

SẢN PHẨM ĐƯỢC CẢI TIẾN ĐỂ PHÙ HỢP NHU CẦU KHÁCH HÀNG

Ami X tuy là sản phẩm miễn phí nhưng vẫn được tiếp tục phát triển và cải tiến, và có đội ngũ hỗ trợ khách hàng trong quá trình sử dụng. Với việc phát triển cộng đồng dành cho những người sử dụng Amibroker, team Ami X luôn lắng nghe nhu cầu của khách hàng và cải tiến sản phẩm mỗi ngày.

HỖ TRỢ GIAO DỊCH TỰ ĐỘNG TỚI CÔNG TY CHỨNG KHOÁN

Giờ đây bạn hoàn toàn có thể đưa việc giao dịch lên một tầm cao mới nhờ khả năng tự động hóa giao dịch theo chiến thuật. Ami X có phát triển tính năng nhận tín hiệu từ Amibroker để đặt lệnh giao dịch tới công ty chứng khoán DNSE. Tính năng này mở ra vô vàn khả năng để nâng cao trình độ và hiệu quả giao dịch của nhà đầu tư, tạo lợi thế cạnh tranh cho những người tiên phong.

TẶNG 100 BOT GIAO DỊCH TỰ ĐỘNG

Giờ đây bạn hoàn toàn có thể đưa việc giao dịch lên một tầm cao mới nhờ khả năng tự động hóa giao dịch theo chiến thuật. Ami X có phát triển tính năng nhận tín hiệu từ Amibroker để đặt lệnh giao dịch tới công ty chứng khoán DNSE. Tính năng này mở ra vô vàn khả năng để nâng cao trình độ và hiệu quả giao dịch của nhà đầu tư, tạo lợi thế cạnh tranh cho những người tiên phong.

Sự khác biệt của AmiX



Khả năng kết nối giao dịch tự động tới Amibroker



Data chính thống, nhanh, chính xác, miễn phí



Cộng đồng NĐT giao dịch tự động Amibroker

Chúng tôi quan tâm tới cộng đồng

Team Ami X xây dựng hệ sinh thái toàn diện để hỗ trợ người sử dụng Amibroker và các nhà giao dịch tự động.

Kênh YouTube Ami X

Kết nối giao dịch chứng khoán tự động: Các video Hướng dẫn sử dụng, Chia sẻ kiến thức - chiến thuật, Học code…

Cộng đồng Zalo Ami X

Kết nối Giao dịch Tự động: hỗ trợ khách hàng sử dụng sản phẩm, giải đáp thắc mắc nhanh nhất

Nhóm Facebook Ami X

Giao dịch chứng khoán tự động: Các bài hướng dẫn tính năng mới, các kinh nghiệm giao dịch tự động

Web phaisinh.online

Trang web phaisinh.online chia sẻ các bài viết chuyên sâu về Amibroker và giao dịch tự động

Bản dịch Hướng dẫn sử dụng Amibroker này cũng là một sản phẩm phi lợi nhuận của Team Ami X với mong muốn cộng đồng sử dụng Amibroker tại Việt Nam có thể dễ tiếp cận và sử dụng nhiều hơn các chức năng của Amibroker.

Bản quyền © 1995-2024 AmiBroker.com.

Bảo lưu mọi quyền

Contents

Bản quyền 9
Nội dung 10
Giới thiệu 11
Giới thiệu về các phiên bản AmiBroker 12
Tham quan nhanh 14
Bắt đầu 19
Yêu cầu về phần cứng 19
Hệ điều hành được hỗ trợ 19
Cài đặt và chạy 19
Nhận trợ giúp 20
Biểu đồ tương thích AmiBroker 32-bit so với 64-bit 22
Có gì mới trong phiên bản mới nhất? 24
Nhật ký thay đổi chi tiết 42
Hướng dẫn 56
Các thao tác cơ bản 58
Thêm ký hiệu mới 58
Xóa ký hiệu 58
Chia tách cổ phiếu 58
Xóa báo giá 58
Hợp nhất các trích dẫn của hai ký hiệu 59
Hướng dẫn lập biểu đồ cho người mới bắt đầu 60
Cách sử dụng giao diện biểu đồ kéo thả 66
Giải thích về các hàm đặc biệt: SECTION_BEGIN, _SECTION_END, _SECTION_NAME, _DEFAULT_NAME, _PARAM_VALUES (chỉ dành cho người dùng nâng cao) 70
Chủ đề biểu đồ 75
Tùy chỉnh giao diện người dùng 82
Làm việc với Bảng biểu đồ và Bố cục cửa sổ 91
Sử Dụng Lớp (Layers) 98
Sử dụng cửa sổ Nghiên cứu Web 100
MỞ CỬA SỔ NGHIÊN CỨU WEB MỚI 100
Sử dụng Account Manager 105
Tạo Tài Khoản Mới 105
Sử dụng Dữ liệu Cơ bản 110

Cửa sổ Thông tin 110
Sử dụng Cửa sổ Phân tích Mới 115
Giới thiệu 115
Giao diện người dùng 115
Sử dụng Giới thiệu về cửa sổ Batch 121
Giới thiệu 121
Giao Diện Người Dùng 121
Các thao tác cơ bản 122
Lệnh Thêm Vào 124
Lập lịch 125
Cách Lấy Báo Giá Từ Các Thị Trường Khác Nhau 127
Cách thiết lập AmiBroker với nguồn dữ liệu eSignal (phiên bản RT chỉ) 133
Yêu cầu 133
Cách thiết lập AmiBroker với nguồn dữ liệu myTrack (phiên bản RT chỉ) 136
Yêu cầu 136
Cách sử dụng AmiBroker với nguồn dữ liệu bên ngoài (Quote Tracker) 138
Thiết lập một lần 138
Cách Cài Đặt Amibroker VớI Dữ Liệu Iqfeed (Chỉ Phiên Bản Rt) 142
Cách sử dụng AmiBroker với nguồn dữ liệu DDE bên ngoài 151
Dde Là Gì 151
Cách Làm Việc Với Trình Bổ Sung Dữ Liệu Thời Gian Thực Lần Đầu Thiết Lập 157
Thiết lập một lần 157
Cách Sử Dụng Amibroker Với Nguồn Dữ Liệu Ngoại Vi (Quotes Plus, Tc2000/Tcnet/Tc2005, Fasttrack, Metastock) 164
Thiết Lập Một Lần 164
Hướng Dẫn Tự Động Cập Nhật Trích Dẫn US Bằng Amiquote 168
Hướng Dẫn Tải Dữ Liệu Báo Giá Thủ Công Sử Dụng Amiquote 171
Giới thiệu 171
Cửa sổ nhập khẩu Metastock. 175
Hiểu về cơ sở dữ liệu Amibroker 177
Giới thiệu 177
Cấu trúc cơ sở dữ liệu Amibroker 177
Dữ liệu dạng External? 177
Hiểu về Danh mục 180
Làm việc với Cơ bản về Ngành và Nghiệp vụ 181
Làm việc với danh sách theo dõi 193
Hiểu cách thức hoạt động của AFL 200
Tạo chỉ báo của riêng bạn 206

Sử dụng kiểu dáng, màu sắc, tiêu đề và tham số biểu đồ trong Indicators 210
Cách tạo bản khám phá của riêng bạn 226
Cách viết bình luận biểu đồ của riêng bạn 233
Sử dụng các nghiên cứu trong công thức AFL 239
Kiểm tra ngược các ý tưởng giao dịch của bạn 241
Backtester cấp danh mục đầu tư 256
Đọc báo cáo kiểm tra lại 271
Cách tối ưu hóa hệ thống giao dịch 274
Kiểm tra tiến trình 283
Hệ thống kiểm tra ngược cho hợp đồng tương lai 288
Mô phỏng Monte Carlo cho hệ thống giao dịch của bạn 292
Kim tự tháp (mở rộng quy mô vào/ra) và nhiều loại tiền tệ trong trình kiểm tra ngược danh mục đầu tư 302
Pyramiding / Scaling 302
Sử dụng cảnh báo dựa trên công thức 306
Sử dụng cửa sổ giải thích 310
Hỗ trợ nhiều khung thời gian trong AFL 311
Sử dụng hiệu quả đa luồng 317
Chức năng xếp hạng 325
Sử dụng đoạn mã AFL 331
Cách sử dụng trình gỡ lỗi trực quan AFL 336
Sử dụng các điều khiển GUI trên biểu đồ 342
Video hướng dẫn (trực tuyến) 350
Biểu đồ 350
Hướng dẫn tham khảo Amibroker 352
Windows trong AmiBroker 353
Cửa sổ biểu đồ 355
Cửa sổ tham số 357
Cửa sổ Dữ liệu (Data Window) 360
Công cụ vẽ nghiên cứu của AmiBroker 361
Cửa sổ Thuộc tính Dòng Nghiên cứu 371
Cửa sổ thuộc tính hộp văn bản 373
Trình soạn thảo công thức 375
Cửa sổ Đoạn mã 387
Cửa sổ Bản đồ Rủi Ro - Lợi Nhuận 390
Hộp thoại Đặt Lệnh 391
Cài đặt Cơ sở Dữ liệu 393
Cửa sổ Cài đặt Intraday 396

Cửa sổ Tùy chọn 399
Tùy chỉnh Cửa sổ Công cụ 413
Cửa sổ Cây Biểu tượng 415
Cửa sổ Thông tin 417
Cửa sổ Notepad 421
Cửa sổ Chỉnh sửa Báo giá 422
Cửa sổ Tìm kiếm Biểu tượng (F3) 424
Sử dụng Cửa sổ Nghiên cứu Web 425
Mở Cửa sổ Nghiên cứu Web Mới 425
Cửa sổ Quản lý Phân công 430
Cửa sổ Tính toán lại dữ liệu Composite 431
Cửa sổ Danh mục 433
Trình Wizard Nhập Dữ Liệu ASCII 435
Cửa sổ Nhập Metastock 439
Sử dụng Account Manager 441
Tạo Một Tài Khoản Mới 441
Cửa sổ báo giá thời gian thực 446
Cửa sổ Cảnh báo Dễ Dàng 449
Cửa sổ Phát Lại Thanh (Bar Replay) 453
Trình Soạn Thảo Công Thức 455
Cửa Sổ Đánh Giá Nhanh 466
Cửa Sổ Phân Tích Tự Động 468
Cửa sổ Thiết lập Bộ lọc 471
Cửa sổ Cài đặt Kiểm tra Hệ thống 472
Báo cáo Thử nghiệm 480
Cửa sổ hoa hồng 486
Cửa sổ bình luận 488
Cửa sổ Plugin 490
Cửa sổ Nhật ký 493
Các Menu 496
Menu Tệp 497
Chỉnh sửa menu 500
Menu xem 502
Chèn menu 505
Menu Định dạng 508
Menu Tượng Trưng 509
Menu Phân Tích 512

Menu Công Cụ 513
Cửa sổ menu 515
Menu ngữ cảnh danh sách kết quả Phân tích Tự động 524
Menu ngữ cảnh biểu đồ 526
Menu ngữ cảnh Bố cục 528
Menu ngữ cảnh Công thức (biểu đồ) 529
Menu ngữ cảnh Các lớp 531
Phím tắt 534
Nhập ASCII 536
Nó hoạt động như thế nào? 536
Các lệnh định dạng định dạng 536
Bình luận 564
Ví dụ sử dụng 564
Hành vi mặc định 568
Các loại tệp và định dạng do người dùng xác định 569
Biệt danh mã chứng khoán 570
Phân loại GICS 570
Phân loại ICB 571
Dấu thời gian có độ phân giải cao (mili giây và micro giây) 571
Mô hình đối tượng tự động hóa OLE của AmiBroker 572
Mục lục đối tượng 572
ADQuotation 573
ADQuotations 573
Phân tích 574
AnalysisDoc 576
AnalysisDocs 579
Ứng dụng 580
Cửa sổ 581
Windows 582
Bình luận 582
Tài liệu 582
Tài liệu 583
Market 584
Markets 584
Quotation 584
Quotations 585
Cổ phiếu 585

Stocks 587
Ví dụ thực tế: 588
Chỉ mục đối tượng Mô hình đối tượng tự động hóa OLE của AmiQuote 590
Mục lục đối tượng 590
Hướng dẫn phân tích kỹ thuật 592
Giới thiệu 592
Các công cụ cơ bản 593
Biểu đồ giá 594
Đường xu hướng 596
Đường trung bình động 598
Fibonacci Retracement 601
Múi giờ Fibonacci 602
Dải Bollinger 603
Chỉ báo 604
Tích lũy/Phân phối 605
Đường Advance-Decline (AD-Line) 606
ADX / Chỉ số chuyển động theo hướng 607
CCI - Chỉ số kênh hàng hóa 608
Chaikin Oscillator 609
MACD - Đường trung bình động hội tụ/phân kỳ 610
Chỉ số dòng tiền 611
Chỉ số khối lượng âm 612
OBV - Khối lượng cân bằng 613
Parabolic SAR (Dừng và đảo ngược) 614
RS - Sức mạnh tương đối (so sánh) 615
RSI - Chỉ số sức mạnh tương đối 616
ROC - Tỷ lệ thay đổi giá 618
Stochastic Slow 619
TRIN - Chỉ số Arms 620
TRIX - TRIple eXponential 621
Ultimate Oscillator 622
VAP - Biểu đồ khối lượng theo giá 623
Biểu đồ hiệu suất tương đối 624
Ngôn ngữ công thức Amibroker (Afl) 625
Giới Thiệu Sổ Tay Tham Chiếu AFL 626
Những Yếu Tố Cơ Bản Của Ngôn Ngữ 626
Yếu tố từ vựng 626

Cấu trúc ngôn ngữ 627
Từ khóa 642
Tham chiếu hàm AFL - Danh sách chức năng theo thứ tự chữ cái 659
Danh Sách Tham Chiếu Hàm Afl - Danh Sách Chức Năng Nhóm Cơ Bản Nhận Biết Mô Hình Giá 671
Lệnh #include - Lệnh gồm nhiều chức năng 683
#include_once - lệnh bao gồm (một lần) của trình tiền xử lý 684
#pragma - thiết lập tùy chọn tiền xử lý AFL 685
DEFAULT NAME- lấy tên mặc định của biểu đồ 686
_DT- chuyển đổi chuỗi thành datetime 688
_exit - chấm dứt việc thực thi công thức AFL 689
_N- không có đầu ra văn bản 690
_PARAM_VALUES- lấy chuỗi giá trị tham số 699
_SECTION_BEGIN- dấu bắt đầu phần 700
_SECTION_END- đánh dấu phần kết thúc 705
_SECTION_NAME- lấy tên phần hiện tại 709
_TRACE- in văn bản vào trình xem gỡ lỗi hệ thống 710
_TRACEF- in văn bản được định dạng vào trình xem gỡ lỗi hệ thống 711
Danh sách lỗi AFL 712
Lỗi 1. Thao tác Không Được Phép. Loại Toán Tử/Toán Hạng Không Khớp. 715
Tính Toán Thống Kê Cho Nhiều Chứng Khoán Với Hàm Addtocomposite 796
Hàm Equity, Biểu đồ Equity Cá nhân và Danh mục 800
Các hàm chấp nhận các khoảng thời gian biến đổi 805
Hàm định nghĩa bởi người dùng, quy trình. Phạm vi cục bộ/toàn cầu 806
Công cụ AFL 810
Phân tích kỹ thuật tự động 810
Cửa sổ phân tích tự động 811
Trình chỉnh sửa công thức 811
Cửa sổ Bình luận Cố vấn Guru 811
Máy chủ kịch bản AFL 813
Cơ bản 813
Yêu cầu 813
Bật Máy chủ Kịch bản AFL 813
Thông tin thêm 819
Hỗ trợ Component Object Model (COM) trong AFL 820
Giới thiệu 820
Gọi các hàm được định nghĩa trong kịch bản 820
Sử dụng các đối tượng COM/ActiveX bên ngoài trong AFL 822

Plugin trong AFL 824
Giao diện Plugin 824
Những Lỗi Lập Trình Thông Thường trong AFL 825
Hướng Dẫn Tham Khảo Giao Diện Kiểm Tra Danh Mục Đầu Tư 831
Cơ Bản 831
Cách thêm chỉ số định nghĩa bởi người dùng vào báo cáo kiểm tra/ tối ưu hóa 843
Sử dụng các hàm đồ họa mức thấp 851
Ví dụ sử dụng: 854
Thông tin kỹ thuật 861
Hướng dẫn khắc phục sự cố 862
Tệp và thư mục được sử dụng bởi AmiBroker 867
Hệ thống phục hồi sự cố và báo cáo lỗi tự động 869
Mẹo điều chỉnh hiệu suất 874
Giám sát hiệu suất 878
Hiệu suất đa luồng 880
Làm thế nào để mua AmiBroker? 881
GIÁ CẢ 881
LỢI ÍCH KHI MUA AMIBROKER 882
Cảm ơn 884
AmiBroker trên mạng 884

Bản quyền

Hướng dẫn sử dụng AmiBroker 6.90.
Bản quyền © 1995-2024 Tomasz Janeczko/AmiBroker.com. Bảo lưu mọi quyền.
AmiBroker, AmiQuote, AmiFeed, AmiBroker.com và logo AmiBroker là các nhãn hiệu của Tomasz Janeczko/AmiBroker.com.

Microsoft, ActiveX, JScript, VBScript, Internet Explorer, MSDN, Windows, Windows NT, Win32 và Win32s là các nhãn hiệu đã đăng ký hoặc nhãn hiệu của Microsoft Corporation tại Hoa Kỳ và/hoặc các quốc gia khác. Amiga là các nhãn hiệu đã đăng ký hoặc nhãn hiệu của Amiga Technologies, GmbH. Windows, Windows NT là các nhãn hiệu đã đăng ký hoặc nhãn hiệu của Microsoft Corporation. Metastock là nhãn hiệu của Equis International, Inc. Các tên thương hiệu và sản phẩm khác là các nhãn hiệu đã đăng ký hoặc nhãn hiệu của chủ sở hữu tương ứng.

Nội dung

• Giới thiệu
• Tham quan nhanh
• Bắt đầu
• Có gì mới trong AmiBroker?
• Hướng dẫn
• Hướng dẫn tham khảo
• Hướng dẫn phân tích kỹ thuật
• Ngôn ngữ công thức AmiBroker (AFL)
• Thông tin kỹ thuật
• Làm thế nào để mua AmiBroker?
• Tín dụng

Bản quyền © 1995-2024 Tomasz Janeczko/AmiBroker.com. Bảo lưu mọi quyền.
AmiBroker, AmiQuote, AmiFeed, Ngôn ngữ công thức AmiBroker, AFL, AmiBroker.com và logo AmiBroker là các nhãn hiệu của Tomasz Janeczko/AmiBroker.com.

AmiBroker sử dụng công nghệ Microsoft Active Scripting. Microsoft, ActiveX, JScript, VBScript, Internet Explorer, MSDN, Windows, Windows NT, Win32 và Win32s là các nhãn hiệu đã đăng ký hoặc nhãn hiệu của Microsoft Corporation tại Hoa Kỳ và/hoặc các quốc gia khác. Windows, Windows NT là các nhãn hiệu đã đăng ký hoặc nhãn hiệu của Microsoft Corporation. Metastock là nhãn hiệu của Equis International, Inc.
Các tên thương hiệu và sản phẩm khác là các nhãn hiệu đã đăng ký hoặc nhãn hiệu của chủ sở hữu tương ứng.

Giới thiệu

Cảm ơn bạn đã chọn AmiBroker. Hướng dẫn này sẽ giúp bạn bắt đầu và chạy.

AmiBroker là một chương trình phân tích kỹ thuật toàn diện, với khả năng lập biểu đồ, kiểm tra ngược và quét tiên tiến. Nó cung cấp mọi thứ bạn cần để giao dịch thành công. Chỉ cần xem qua **chuyến tham quan các tính năng nhanh** của chúng tôi để tìm hiểu những gì có trong gói phần mềm mạnh mẽ này.

Nếu bạn là **người dùng lần đầu** và vừa cài đặt phần mềm, vui lòng xem phần **Hướng dẫn** sẽ hướng dẫn bạn qua hầu hết các khía cạnh quan trọng khi sử dụng AmiBroker.

Chương tiếp theo - **Hướng dẫn tham khảo** - cung cấp mô tả chi tiết về mọi cửa sổ và nhiều tài liệu kỹ thuật hơn về trình nhập ASCII và giao diện tự động hóa.

Trong **hướng dẫn phân tích kỹ thu**ật, bạn sẽ tìm thấy tài liệu giới thiệu cho bạn về thế giới biểu đồ và chỉ báo kỹ thuật.

Phần tiếp theo của hướng dẫn mô tả Ngôn ngữ công thức AmiBroker - một công cụ mạnh mẽ cho phép bạn tạo hệ thống giao dịch, quét, chỉ báo tùy chỉnh và bình luận của riêng mình. Bạn sẽ tìm thấy mô tả về ngôn ngữ và cú pháp của ngôn ngữ, tài liệu tham khảo đầy đủ về tất cả các chức năng và nhiều hơn nữa.

Phần cuối được cung cấp cho người dùng các phiên bản trước - chương này sẽ giúp họ tìm hiểu những tính năng mới đã được thêm vào mà không cần phải đọc lại toàn bộ tài liệu.

Giới thiệu về các phiên bản AmiBroker

Phần mềm AmiBroker hiện có 2 phiên bản: Standard và Professional.

Bảng sau tóm tắt sự khác biệt giữa hai phiên bản này:

Tính năng	Phiên bản tiêu chuẩn	Phiên bản chuyên nghiệp
Biểu đồ cuối ngày/kiểm tra ngược/quét	Có	Có
Biểu đồ trong ngày/kiểm tra ngược/quét 1, 5, 15 phút, hàng giờ	Có	Có
Thanh phút tùy chỉnh	Có	Có
Biểu đồ tích tắc/kiểm tra ngược/quét	Không	**Có***
Biểu đồ thanh 1 giây, 5 giây, 15 giây/kiểm tra ngược/quét	Không	**Có**
Truyền phát hiển thị báo giá theo thời gian thực	10 ký hiệu	**KHÔNG GIỚI HẠN ký hiệu**
Cửa sổ thời gian và bán hàng	1 ký hiệu	**KHÔNG GIỚI HẠN ký hiệu**
Chức năng GetRTData / GetRTDataForeign AFL	Không	**Có**
Chờ để điền lại trong Phân tích tự động	Không	**Có**
Tự động cập nhật biểu đồ theo thời gian thực	Có	**Có**
Biểu đồ Phân phối biến động bất lợi/thuận lợi tối đa trong báo cáo kiểm tra ngược danh mục đầu tư	Không	**Có**
Phiên bản 64 bit	Không	**Có**
Biểu đồ đa luồng	Có	Có
Cửa sổ Phân tích đa luồng	Có, tối đa 2 luồng	Có, **tối đa 32 luồng ****
Bitness	32 bit	chỉ 64 bit và 32 bit
Yêu cầu đăng ký dữ liệu RT	Không	Không bắt buộc, nhưng tốt nếu có (Phiên bản chuyên nghiệp hoạt động hoàn hảo với dữ liệu EOD, nhưng

Tính năng	Phiên bản tiêu chuẩn	Phiên bản chuyên nghiệp
		tất nhiên các tính năng thời gian thực (như báo giá thời gian thực) yêu cầu nguồn dữ liệu thời gian thực)

* - tính năng này chỉ khả dụng khi sử dụng eSignal RT, Interactive Brokers, nguồn cấp dữ liệu DDE
Trong tương lai, Phiên bản chuyên nghiệp có thể có thêm các tính năng bổ sung không khả dụng trong Phiên bản tiêu chuẩn. Để biết thông tin về giá cả và đặt hàng, hãy xem phần **Cách đặt hàng.**

** - số luồng phụ thuộc vào số bộ xử lý logic trên máy tính của bạn và số ký hiệu đang được thử nghiệm. Để biết chi tiết, hãy xem: **Sử dụng hiệu quả Đa luồng.**

Tham quan nhanh

Biểu đồ mạnh mẽ

• công cụ vẽ hướng đối tượng (đường xu hướng, tia, đường song song, kênh hồi quy, đường thoái lui fibonacci, mở rộng, phần mở rộng thời gian Fibonacci, múi giờ Fibonacci, cung, hình vuông gann, hình vuông gann, chu kỳ, hình tròn, hình chữ nhật, văn bản trên biểu đồ, v.v.)
• tạo chỉ báo kéo và thả - cho phép bạn tạo các chỉ báo phức tạp mà không cần viết một dòng mã nào
• giao diện người dùng hiện đại, có thể tùy chỉnh hoàn toàn
• xem ngay lập tức các biểu đồ trong ngày/hàng ngày/hàng tuần/hàng tháng theo kiểu đường, thanh hoặc nến được phủ lên các đường trung bình động có thể định cấu hình, dải Bollinger, biểu đồ Khối lượng, SAR, v.v.
• khả năng hiển thị hầu hết các biểu đồ trong ngày phổ biến nhất trong 1, 5, 15, 60 phút cũng như **các biểu đồ N phút có thể tùy chỉnh hoàn toàn (trong đó N là 1..1380)**
• biểu đồ thanh 5 giây và 15 giây (phiên bản RT)
• biểu đồ tích tắc, biểu đồ N tích tắc tùy chỉnh (Phiên bản RT)
• biểu đồ nhiều khung thời gian
• nén thời gian tức thời - không cần phải chờ khi chuyển đổi giữa các chu kỳ biểu đồ khác nhau
• biểu đồ hiệu suất tương đối
• hàng chục chỉ báo phổ biến nhất được tích hợp sẵn bao gồm ROC, RSI, MACD, OBV, CCI, MFI, NVI, Stochastics, Ultimate oscillator, DMI, ADX, Parabolic SAR, TRIN, đường Advance/Decline, Accumulation/Distribution, TRIX, Chaikin oscillator, bản đồ rủi ro-lợi nhuận độc đáo và nhiều hơn nữa
• nghiên cứu các công cụ vẽ bao gồm đường xu hướng, đường ngang/dọc, đường thoái lui Fibonacci và múi giờ, hộp văn bản
• nhiều khung biểu đồ, cửa sổ, nhiều chế độ xem và thang thời gian khác nhau có thể thực hiện cùng một lúc
• phóng to cực nhanh và cuộn trực tiếp

Nhiều nguồn cấp dữ liệu

AmiBroker có khả năng xử lý hầu như BẤT KỲ sàn giao dịch nào trên thế giới.

• Báo giá phát trực tiếp theo thời gian thực qua TurboFeed của eSignal với khả năng truy cập vào tất cả các sàn giao dịch của Hoa Kỳ và các sàn giao dịch lớn của Châu Âu.
• Báo giá phát trực tiếp theo thời gian thực qua nguồn cấp dữ liệu myTRACK, IQFeed, QCharts/Quote.com, QuoteTracker, Interactive Brokers, bất kỳ nguồn cấp dữ liệu nào hỗ trợ DDE
• Nguồn cấp **trực tiếp** từ các cơ sở dữ liệu Quotes Plus, TC2000, FastTrack và Metastock (bao gồm cả trong ngày). Đọc thêm...
• Trình hướng dẫn nhập ASCII có thể định cấu hình bởi người dùng - cho phép bạn đọc báo giá theo định dạng bạn có thể xác định (bao gồm cả trong ngày)!
• Trình nhập cơ sở dữ liệu Metastock(R) tích hợp - đọc trực tiếp tất cả các ký hiệu từ cơ sở dữ liệu Metastock của bạn (hoạt động với cả chế độ EOD và trong ngày) chỉ trong vài giây!
• Chương trình tải xuống AmiQuote cung cấp cách nhanh chóng để có được dữ liệu cuối ngày miễn phí từ các sàn giao dịch lớn trên thế giới (tất cả các thị trường Hoa Kỳ, LSE, ASX, Paris, Milan, Frankfurt)
• Dữ liệu FOREX miễn phí có thể tải xuống qua AmiQuote

• Báo giá trễ trong ngày miễn phí từ các sàn giao dịch Hoa Kỳ có thể tải xuống qua AmiQuote
• Trình tải xuống tự động bằng một cú nhấp chuột, được điều khiển bằng tập lệnh có sẵn cho NYSE, Amex, Nasdaq, Sàn giao dịch chứng khoán Úc, Sàn giao dịch chứng khoán Johannesburg, Sàn giao dịch chứng khoán Warsaw

AmiBroker được sử dụng thành công tại các quốc gia sau: Hoa Kỳ, Canada, Vương quốc Anh, Úc, Đức, Ý, Nam Phi, Ba Lan, Hà Lan, Na Uy, Pháp, ...

Để biết thêm thông tin về các nguồn dữ liệu cho AmiBroker, hãy nhấp vào đây.

Cơ sở dữ liệu biểu tượng & báo giá

AmiBroker có hệ thống cơ sở dữ liệu tiên tiến cung cấp các tính năng sau:

• xây dựng và lưu trữ dữ liệu tích tắc lịch sử hoặc thanh 5 hoặc 15 giây cho mục đích kiểm tra ngược (chỉ một số nguồn dữ liệu RT nhất định)
• xây dựng và lưu trữ dữ liệu thanh phút trong ngày hoặc cuối ngày cho mục đích kiểm tra ngược
• số lượng biểu tượng không giới hạn và số lượng báo giá không giới hạn
• hỗ trợ nhiều cơ sở dữ liệu
• lưu trữ báo giá, thông tin công ty, kết quả tài chính, danh mục, thông tin ngành/lĩnh vực
• lọc mạnh mẽ theo lĩnh vực, ngành, nhóm và thị trường
• trình duyệt cây biểu tượng sáng tạo hiển thị các biểu tượng được nhóm theo lĩnh vực, ngành, chỉ số
• xử lý tự động cho các thành phần (số lượng và khối lượng biểu tượng tăng, giảm và không đổi)
• hỗ trợ tự động hóa cho phép bạn kiểm soát cơ sở dữ liệu của mình từ các chương trình bên ngoài được viết bằng bất kỳ ngôn ngữ nào bao gồm Java Script, VBScript

Ngôn ngữ

AFL là ngôn ngữ công thức nâng cao cho phép bạn tạo các chỉ báo, hệ thống giao dịch và bình luận của riêng mình. Ngôn ngữ này được thiết kế đặc biệt cho các nhà giao dịch nên việc viết công thức phân tích dễ dàng và nhanh hơn so với các ngôn ngữ thông dụng.

AFL có hơn 200 hàm AFL tích hợp để sử dụng làm khối xây dựng cho các công thức của bạn. AFL bao gồm các hàm lượng giác, trung bình, thống kê, thao tác dữ liệu, có điều kiện, phát hiện mẫu và các hàm chỉ báo được xác định trước.

AFL hỗ trợ các biến không giới hạn, lồng dấu ngoặc đơn không giới hạn, gọi hàm lồng nhau không giới hạn và nhiều toán tử logic. Phiên bản 4.40 mang đến công cụ được viết lại hoàn toàn với điều khiển luồng và vòng lặp gốc (if-else, while), các hàm và quy trình do người dùng xác định với phạm vi biến cục bộ và toàn cục.

Phiên bản mới 4.50 cung cấp hỗ trợ nhiều khung thời gian gốc, do đó bạn có thể kết hợp các khoảng thời gian thanh khác nhau trong một công thức.

Công thức Editor / Biểu đồ kéo-thả

Công thức Editor cho phép bạn nhanh chóng tạo lại bất kỳ chỉ báo/nghiên cứu nào được tìm thấy trong tài liệu. Biểu đồ kéo và thả cho phép tạo lớp phủ phức tạp, chỉ báo trên chỉ báo và nhiều hơn nữa. Trong số những thứ khác, có thể:

• bất kỳ số lượng biểu đồ nào có thể được phủ lên cùng một khung biểu đồ
• sửa đổi các chỉ báo tích hợp
• tùy chỉnh hoặc tự động mở rộng
• lưới linh hoạt
• truy cập vào dữ liệu tổng hợp (số lượng/khối lượng các vấn đề tăng, giảm, không đổi)

Cảnh báo dựa trên công thức

• Khả năng viết cảnh báo dựa trên công thức phức tạp có thể hiển thị trên màn hình, gửi cho bạn qua e-mail, cộng với phát tệp WAV do người dùng xác định.
• Khả năng chạy các ứng dụng bên ngoài thông qua cảnh báo - điều này cho phép thực hiện giao dịch tự động

Kiểm tra ngược, tối ưu hóa, khám phá và sàng lọc hệ thống CẤP ĐỘ DANH MỤC

Sàng lọc: Cửa sổ phân tích tự động cho phép bạn quét cơ sở dữ liệu của mình để tìm các ký hiệu khớp với các quy tắc mua/bán đã xác định của bạn. AmiBroker tự động tạo báo cáo cho bạn biết liệu các tín hiệu mua/bán có xảy ra trên ký hiệu nhất định trong khoảng thời gian đã chỉ định hay không.

Khám phá: tìm kiếm trong cơ sở dữ liệu của bạn các ký hiệu phù hợp với tiêu chí của bạn và tạo báo cáo hiển thị dữ liệu bạn muốn xem: giá trị chỉ báo, hiệu suất trong quá khứ, v.v. Sau đó sắp xếp kết quả theo bất kỳ giá trị nào được liệt kê.

Kiểm tra ngược: AmiBroker cũng có thể thực hiện kiểm tra ngược đầy đủ tính năng cho chiến lược giao dịch của bạn, giúp bạn biết được hiệu suất của hệ thống.

Công cụ kiểm tra ngược nêu bật:

• KIỂM TRA/TỐI ƯU HÓA CẤP ĐỘ DANH MỤC
• Biểu đồ ba chiều (3D), hoạt hình đầy đủ về kết quả tối ưu hóa
• Giao diện kiểm tra ngược tùy chỉnh nâng cao
• Số liệu kiểm tra ngược do người dùng xác định
• Các kỹ thuật quản lý tiền/quy mô vị thế khác nhau dựa trên Vốn chủ sở hữu danh mục đầu tư
• Thực hiện siêu nhanh - AmiBroker có thể kiểm tra ngược 10000 ký hiệu (mỗi ký hiệu 3000 thanh dữ liệu) = 30 triệu điểm dữ liệu trong NĂM phút!
• Hỗ trợ tích hợp cho NHIỀU khung thời gian trong một công thức
• Report Explorer MỚI cung cấp cách tuyệt vời để sắp xếp/so sánh/xem tất cả kết quả kiểm tra ngược
• Quét/Khám phá/Kiểm tra ngược/Tối ưu hóa trên dữ liệu Thời gian thực (tích tắc và tăng dần) (chỉ dành cho phiên bản RT)
• Quét/Khám phá/Kiểm tra ngược/Tối ưu hóa trên dữ liệu trong ngày (thanh 1 phút trở lên)
• Kiểm tra ngược toàn bộ sàn giao dịch hoặc chỉ một tập hợp giới hạn, do người dùng xác định phù hợp với lựa chọn thị trường, nhóm, ngành, lĩnh vực của bạn
• Vẽ đường cong vốn chủ sở hữu, cầu vồng vốn chủ sở hữu, đường cong vốn chủ sở hữu tổng hợp
• Kiểm tra các giao dịch mua, bán hoặc cả mua và bán

• Dừng lỗ tối đa, dừng mục tiêu lợi nhuận, dừng theo sau, dừng N-bar (thời gian)
• Kiểm tra ngược thực tế
• Khả năng kiểm soát quy mô vị thế từ công thức của bạn (Đọc thêm...)
• Tạo các hợp chất của riêng bạn và quét/kiểm tra ngược chúng
• Báo cáo chi tiết cung cấp cho bạn số liệu thống kê quan trọng về hệ thống.

Tối ưu hóa: AmiBroker cho phép bạn tối ưu hóa hệ thống giao dịch của mình với tối đa 10 biến tối ưu hóa trên một hoặc **NHIỀU** chứng khoán cùng một lúc!

Bình luận và diễn giải biểu đồ tự động

• Mô tả đầy đủ, dạng văn bản về tình hình thực tế trên thị trường
• mũi tên mua-bán tự động hiển thị trên biểu đồ
• diễn giải dạng văn bản tự động của các chỉ báo và biểu đồ giá (Cửa sổ->Diễn giải)

Hỗ trợ tập lệnh/COM/DLL

• Công cụ AFL cho phép nhúng mã VBScript/JScript vào các công thức AFL cung cấp khả năng KHÔNG GIỚI HẠN
• khả năng gọi các đối tượng COM (ActiveX) bên ngoài từ công thức AFL
• SDK miễn phí (bộ phát triển phần mềm) cho người dùng đã đăng ký, cho phép viết DLL chỉ báo (phần bổ trợ)
• nhiều phần bổ trợ của bên thứ 3 đã có sẵn

Trình quản lý danh mục đầu tư

Trình quản lý danh mục đầu tư tích hợp giúp bạn theo dõi các khoản đầu tư của mình. Nó cho phép bạn đăng ký các giao dịch mua/bán, tính toán hoa hồng môi giới, cổ tức (có thuế cổ tức có thể thiết lập), tiền gửi/rút tiền mặt. Bạn nhận được tính toán tức thời về giá trị vốn chủ sở hữu, tỷ lệ phần trăm và lợi nhuận điểm của mình.

Hỗ trợ tập lệnh

AmiBroker có giao diện tự động hóa hiển thị các đối tượng và phương pháp có thể truy cập từ bất kỳ ngôn ngữ lập trình nào bao gồm các phương ngữ tập lệnh như JScript (JavaScript) và VBScript. Khả năng tập lệnh của AmiBroker cho phép bạn tự động hóa các tác vụ quản lý cơ sở dữ liệu tốn thời gian. Sử dụng tập lệnh, bạn sẽ có thể tạo trình tải xuống tự động, công cụ bảo trì, trình xuất tùy chỉnh theo nhu cầu cụ thể của mình.

Tích hợp Internet

AmiBroker có trình duyệt web tích hợp cho phép bạn nhanh chóng xem hồ sơ công ty. Trình xem hồ sơ có thể định cấu hình hoàn toàn để bạn có thể thiết lập cho sàn giao dịch cụ thể của mình.

Các cài đặt dựa trên thị trường để bạn có thể tự động truy cập các trang web khác nhau cho từng thị trường. Bạn sẽ không còn phải tốn thời gian duyệt thủ công để có được tin tức mới nhất và thông tin liên quan đến biểu tượng nữa.

Khả năng cấu hình

AmiBroker được thiết kế để có thể cấu hình và tùy chỉnh trong hầu hết mọi lĩnh vực. Nó không bị ràng buộc với sàn giao dịch hoặc nhà cung cấp dữ liệu cụ thể. Nhờ các phương pháp nhập và tập lệnh linh hoạt, bạn sẽ có thể dễ dàng áp dụng nó vào thị trường yêu thích của mình. Ngoài ra, các công cụ phân tích kỹ thuật được tích hợp trong AmiBroker cho phép bạn thay đổi mọi thông số một cách dễ dàng và nếu bạn muốn nhiều hơn nữa, bạn có thể tạo các chỉ báo của riêng mình bằng ngôn ngữ công thức linh hoạt.

Bắt đầu

Yêu cầu về phần cứng
Hệ điều hành được hỗ trợ
Cài đặt và chạy
Nhận trợ giúp

Yêu cầu về phần cứng

Để chạy AmiBroker, bạn cần máy tính tương thích với PC đáp ứng các yêu cầu tối thiểu sau

- Pentium 450 MHz trở lên
- RAM 256 MB
- Dung lượng ổ cứng 20 MB
- Card đồ họa 256 màu (khuyến nghị màu cao) Độ phân giải màn hình tối thiểu 800x600

Cấu hình máy được khuyến nghị

- CPU: 1 GHz trở lên, nhiều lõi
- RAM 4 GB trở lên

Hệ điều hành được hỗ trợ

AmiBroker hoạt động trên các hệ điều hành sau:

- Windows 11 (bất kỳ phiên bản nào) 32 bit
- Windows 11 (bất kỳ phiên bản nào) 64 bit
- Windows 10 (bất kỳ phiên bản nào) 32 bit
- Windows 10 (bất kỳ phiên bản nào) 64 bit
- Windows 8 (bất kỳ phiên bản nào) 32-bit
- Windows 8 (bất kỳ phiên bản nào) 64-bit
- Windows 7 (bất kỳ phiên bản nào) 32-bit
- Windows 7 (bất kỳ phiên bản nào) 64-bit
- Windows Vista (bất kỳ phiên bản nào) 32-bit
- Windows Vista (bất kỳ phiên bản nào) 64-bit
- Windows Server 2012 (bất kỳ phiên bản nào) 32-bit
- Windows Server 2012 (bất kỳ phiên bản nào) 64-bit
- Windows Server 2008 R2 (bất kỳ phiên bản nào) 64-bit
- Windows Server 2008 (bất kỳ phiên bản nào) 32-bit
- Windows Server 2008 (bất kỳ phiên bản nào) 32-bit
- Windows XP (bất kỳ phiên bản nào)
- Windows XP x64 (64-bit)
- Windows 2000 (bất kỳ phiên bản nào)

Để biết thêm thông tin về khả năng tương thích của hệ điều hành, hãy xem: Khả năng tương thích của phiên bản 32-bit/64-bit biểu đồ

Cài đặt và chạy

Cài đặt AmiBroker bằng chương trình thiết lập của nó - có thể tải xuống từ http://www.amibroker.com/download.html. Sau khi tải xuống, hãy nhấp đúp vào biểu tượng của chương trình. Thao tác này sẽ khởi chạy chương trình thiết lập - bạn có thể chấp nhận

an toàn tất cả các giá trị mặc định bằng cách nhấp vào "Tiếp theo" trên mỗi trang và "Cài đặt" trên trang cuối cùng. Theo mặc định, AmiBroker được cài đặt vào thư mục "C:\Program Files\AmiBroker" và vị trí này được gọi là "thư mục AmiBroker chính".

Nếu chương trình thiết lập yêu cầu bạn khởi động lại máy, vui lòng thực hiện để cho phép thay thế các thành phần hệ thống.

Sau khi cài đặt, bạn có thể khởi động AmiBroker từ menu Start->Programs->AmiBroker->AmiBroker chuẩn của Windows.

Ngay sau khi khởi động AmiBroker, cửa sổ chào mừng sẽ xuất hiện, sau đó AmiBroker sẽ tải cơ sở dữ liệu báo giá của nó trong vài giây. Tiếp theo, màn hình AmiBroker chính sẽ xuất hiện.

Màn hình chính của AmiBroker với biểu đồ giá,
các chỉ báo MACD và RSI và chế độ xem hồ sơ mở ra. (Phiên bản Windows)

Trong thiết lập mặc định, bạn có thể thấy thanh công cụ, cửa sổ không gian làm việc với danh sách ký hiệu ở bên trái và cửa sổ biểu đồ ở bên phải.

Thanh công cụ cung cấp quyền truy cập nhanh vào các chức năng chương trình thường dùng nhất. Với chế độ xem danh sách ký hiệu, bạn có thể chọn ký hiệu đang hoạt động. Việc thay đổi lựa chọn sẽ khiến biểu đồ được vẽ lại và cập nhật trong một số cửa sổ thông tin nếu chúng đang mở. Cửa sổ biểu đồ cho phép bạn phân tích xu hướng giá hiện tại và hành vi của các chỉ báo kỹ thuật.

Bạn có thể thoát AmiBroker bằng mục menu Tệp/Thoát.

Nhận trợ giúp

AmiBroker có hệ thống trợ giúp theo ngữ cảnh mới, có thể sử dụng bằng cách nhấn phím F1 ở bất kỳ đâu trong chương trình.

Khi bạn nhấn phím F1 trong khi bất kỳ cửa sổ và menu nào được hiển thị, AmiBroker sẽ mở trang tệp trợ giúp có liên quan mô tả cửa sổ hoặc menu đang đề cập. Không cần tìm kiếm trong tệp trợ giúp nữa.

Ngoài việc sử dụng trợ giúp theo ngữ cảnh F1, bạn nên đọc TẤT CẢ các bài viết Hướng dẫn trước. Câu trả lời cho hầu hết các vấn đề thường gặp đều có ở đó. Trong trường hợp có vấn đề lớn, hãy kiểm tra Hướng dẫn khắc phục sự cố. Ngoài ra, còn có một tab "Tìm kiếm" ở bên trái cửa sổ trợ giúp trực tuyến này cho phép nhanh chóng tìm thông tin theo từ khóa. Chỉ cần nhập từ bạn đang tìm kiếm và nhấp vào "Hiển thị".

Trong trường hợp có thêm câu hỏi/vấn đề, bạn có thể kiểm tra các tài nguyên sau:

• Trang web AmiBroker - có thể tìm kiếm bằng hộp "Tìm kiếm" ở góc trên cùng bên trái của trang. Trang này cung cấp cho bạn quyền truy cập vào:
    • Bản tin AmiBroker Tips có hướng dẫn từng bước hữu ích về cách sử dụng nhiều khía cạnh khác nhau của AmiBroker
    • Khu vực hỗ trợ - có tài liệu bổ sung
    • Câu hỏi thường gặp - danh sách các câu hỏi thường gặp nhất kèm theo câu trả lời
    • Thư viện AFL - có các công thức AFL sẵn sàng sử dụng cho các chỉ báo tùy chỉnh, bình luận và hệ thống giao dịch

• Khu vực thành viên - có tài liệu chỉ dành cho người dùng đã đăng ký
• Diễn đàn AmiBroker nơi bạn có thể gặp gỡ những người dùng AmiBroker khác, đặt câu hỏi và chia sẻ ý tưởng (có kho lưu trữ có thể tìm kiếm).

Kiểm tra những nơi này trước sẽ giúp tôi tập trung vào việc phát triển các tính năng mới trong AmiBroker. Trong trường hợp có vấn đề không được đề cập trong các tài nguyên trên, vui lòng sử dụng trang hỗ trợ khách hàng tại http://www.amibroker.com/support.html

Biểu đồ tương thích AmiBroker 32-bit so với 64-bit

TÓM TẮT

Bảng sau đây cho thấy rõ ràng rằng AmiBroker Professional 32 bit chạy trên MỌI phiên bản Windows (CẢ 32 và 64-bit) và với mọi plugin dữ liệu. Phiên bản 64-bit của AmiBroker chỉ chạy trên các phiên bản Windows 64-bit và chỉ với số lượng nguồn dữ liệu hạn chế do thiếu hỗ trợ API của nhà cung cấp dữ liệu cho công nghệ 64-bit.

	AmiBroker Standard AmiBroker Professional 32-bit	AmiBroker Professional 64-bit
Hệ điều hành		
Windows 11 32-bit	Có	Không
Windows 11 64-bit	Có	Có
Windows 10 32-bit	Có	Không
Windows 10 64-bit	Có	Có
Windows 8 32-bit	Có	Không
Windows 8 64-bit	Có	Có
Windows 7 32-bit	Có	Không
Windows 7 64-bit	Có	Có
Windows Vista	Có	Không
Windows Vista x64	Có	Có
Windows Server 2008-2022 (32-bit)	Có	Không
Windows Server 2008-2022 x64 (64-bit)	Có	Có
Windows XP	Có	Không
Windows XP x64 (64-bit)	Có	Có
Windows 2000	Có	Không
Windows NT 4	Không4)	Không
Windows Millenium Edition (ME)	Không4)	Không
Windows 98	Không4)	Không
Windows 95	Không4)	Không

Không gian bộ nhớ có thể định địa chỉ		
Hệ điều hành 32 bit	2 GB hoặc 3 GB3)	Không gian bộ nhớ có thể định địa chỉ
Hệ điều hành 64 bit	4 GB	1000 GB

Nguồn dữ liệu	Có	Có
AmiQuote (Yahoo, MSN, Google Finance)	Có	Có
Nhập Metastock	Có	Có
Plugin Metastock	Có	Có
eSignal	Có	Có
RT	Có	Có
IQFeed RT	Có	Có
Interactive Brokers	Có	Có
Dữ liệu cao cấp (qua plugin Metastock)	Có	Có
TC2000/TCNet	Có	Không2)
FastTrack	Có	Không2)
DDE	Có	Có
Cơ sở dữ liệu ODBC	Có	Có
bất kỳ tệp ASCII nào (qua nhập)	Có	Có

Ghi chú:

1) Phiên bản gốc 64 bit của plugin này có thể thực hiện được và đang được phát triển
2) Phiên bản gốc 64 bit của plugin này về mặt kỹ thuật là không khả thi vì hiện tại nhà cung cấp dữ liệu không có API 64 bit
3) Bộ nhớ có thể định địa chỉ 3 GB chỉ khả thi khi chuyển đổi /3G trong Tệp BOOT.INI của Windows 32-bit
4) Windows 95, 98, Me, NT4 chỉ được hỗ trợ bởi phiên bản AmiBroker cũ 5.xx

Có gì mới trong phiên bản mới nhất?

Điểm nổi bật của phiên bản 6.90

- Cải tiến trình soạn thảo AFL
  - Điều hướng đa con trỏ: Cải tiến chỉnh sửa với hỗ trợ đa con trỏ, chỉnh sửa nhiều lựa chọn và chỉnh sửa cửa sổ chia đôi.
  - Công cụ thông tin hiển thị mô tả và tham số của từng chức năng khi bạn nhập chúng
  - Tin nhắn "profiler/checker" nội tuyến: Đã thêm để hiển thị các dòng/chức năng nào gây ra việc tìm kiếm trong tương lai.
  - Tỷ lệ HiDPI tốt hơn
  - Dữ liệu tổng hợp để gỡ lỗi: Khi một ký hiệu không có dấu ngoặc kép, dữ liệu tổng hợp sẽ được tạo để cho phép xác minh mã và gỡ lỗi.
  - Khi phát hiện lỗi, nút "Giải thích" sẽ được thêm vào thanh chú thích hiển thị trang trợ giúp với giải thích lỗi
  - Phát hiện mã ChatGPT: Đã thêm thông báo để phát hiện và thông báo cho người dùng về mã có khả năng không hợp lệ do ChatGPT tạo ra.
  - Cải tiến giao diện người dùng: Đã sửa lỗi đánh máy trong danh sách thả xuống tự động hoàn thành, tỷ lệ chiều rộng lề gấp với phông chữ/DPI và cửa sổ xem hiển thị toàn bộ các mảng nhỏ.

- Trình duyệt Web Research mới
  - Trình duyệt Chromium / Edge (WebView2) hoàn toàn mới, tuân thủ tiêu chuẩn hiện có sẵn như một giải pháp thay thế cho trình duyệt cũ

    Để sử dụng công cụ trình duyệt web mới trong AmiBroker, bạn phải sử dụng AmiBroker 64 bit và thực hiện như sau:

    1. Tải xuống thành phần x64 WebView2 từ Microsoft https://developer.microsoft.com/en-us/microsoft-edge/webview2/

    2. Trong AmiBroker, hãy vào Tools->Preferences, tab "Miscellaneous" và bật tùy chọn "Sử dụng Chromium/Edge (WebView2) cho Web Research"

    3. Khởi động lại AmiBroker
- Các tính năng và cải tiến mới của AFL
  - thêm hỗ trợ cho bản đồ / từ điển (cặp khóa-giá trị) - hàm MapCreate
  - thêm hỗ trợ cho chuỗi thô C++ và chuỗi thô kiểu Python
  - thêm chuẩn hóa tự động AddToComposite thông qua atcFlagNormalize
  - thêm hàm TrimResultRows
  - đã thêm hàm SparseInterpolate
  - đã thêm GetOption("BHSymbol")
  - Nz, SafeDivide, Matrix, StaticVarInfo, SumSince được cải tiến
  - đã thêm các khai báo biến tĩnh với giá trị khởi tạo tùy chỉnh (thay vì Null)
  - kiểm tra chặt chẽ hơn chuỗi định dạng printf/StrFormat
  - tự động lưu một lần các biến tĩnh liên tục thông qua SetOption("StaticVarAutoSave", -1)
  - Khởi tạo ma trận: Tăng tham số mới cho phép tạo ma trận với các phần tử tăng đơn điệu.
  - Cải tiến bộ nhớ và hiệu suất: Tăng kích thước khối phân bổ vi mô lên 64 byte và cải thiện khả năng xử lý các biến tĩnh.

    - Xử lý lỗi: Mã lỗi mới, kiểm tra chặt chẽ hơn đối với chuỗi định dạng và cảnh báo nâng cao đối với các ký tự không hợp lệ.
    - Chức năng điều khiển GUI*: Đã thêm tham số 'style' bổ sung cho tất cả các chức năng Gui*.

- Cải tiến biểu đồ/tính năng mới
    - Công cụ văn bản cho phép chỉ định kiểu phông chữ/kích thước
    - tăng giới hạn khoảng cách lưới X tối thiểu cho màn hình HiDPI
    - nhiều bản sửa lỗi, cải tiến và tăng tốc QuickGFX

- Cải tiến cửa sổ phân tích/tính năng mới
    - thêm thông tin khi vốn chủ sở hữu thu được từ lãi suất lớn hơn từ giao dịch riêng lẻ
    - thêm thông tin về số lần đạt đến giới hạn thanh khoản
    - thêm hỗ trợ cho "Hiển thị mũi tên giao dịch hiện tại" sau khi chạy EXPLORATION
    - Thông tin phân tích mới: Tóm tắt các ký hiệu có dấu ngoặc kép bị lọc do phần đệm hiển thị trong tab INFO.
    - Tối ưu hóa riêng lẻ: Đã khắc phục sự cố và vi phạm chia sẻ khi GenerateReports được bật.
    - Độ phân giải dấu thời gian: Tăng độ phân giải cho dấu thời gian của thư mục báo cáo bằng cách sử dụng bộ đếm duy nhất.
    - Quản lý danh mục: Hộp thoại bộ lọc hiển thị hướng dẫn nếu không có ký hiệu nào đáp ứng tiêu chí đã chọn và hiển thị danh mục có tên trống.

- Report Explorer
    - Cải thiện hiệu suất: Đọc lại/làm mới nhanh hơn 10 lần và hiệu suất hiển thị được cải thiện 20 lần khi sử dụng triển khai dạng xem danh sách siêu nhanh.
    - Xử lý lỗi: Thông báo vi phạm chia sẻ được hiển thị lặng lẽ trên thanh trạng thái thay vì hộp thoại chặn chế độ modal.

- Cải thiện bảo trì cơ sở dữ liệu/nguồn cấp dữ liệu
    - Chỉnh sửa->Xóa phạm vi hỗ trợ nhiều lựa chọn ký hiệu, do đó bạn có thể xóa phạm vi dấu ngoặc kép khỏi nhiều ký hiệu cùng một lúc
    - Plugin IQFeed hiện có tùy chọn hợp nhất các dấu tích chưa đóng gói và tra cứu ký hiệu trong menu ngữ cảnh
    - Nhập ASCII: Cải thiện hiệu suất bằng cách xóa sớm các dòng trùng lặp trong các tệp lớn.

- Cải thiện giao diện người dùng
    - hộp thoại chế độ modal nhấp nháy và di chuyển đến vị trí của chuột để tránh nhầm lẫn cho người dùng khi hộp thoại chế độ modal mở trên màn hình khác chặn chương trình tiếp tục
    - Hỗ trợ DPI cao: Nhiều thành phần giao diện người dùng, chẳng hạn như nút bật ra, hộp thoại chế độ modal, nút mở rộng lưới thuộc tính và đơn vị đánh dấu XAML, hiện được chia tỷ lệ chính xác với màn hình DPI cao.
    - thêm trang Bắt đầu để cải thiện định hướng cho người dùng mới
    - thêm nút tạm dừng vào Trình lập lịch hàng loạt

    - Cải tiến chế độ tối: Tăng khả năng đọc các mục chỉ đọc trong trang Tóm tắt tài khoản ở chế độ tối.
    - Thanh biểu tượng: Thêm hỗ trợ phím menu ngữ cảnh (VK_APPS) ngoài chức năng nhấp chuột phải.
    - Hộp thoại danh mục: Ngăn chặn việc đặt tên danh mục thành trống, quay lại tên mặc định nếu cố gắng.

- Các bản sửa lỗi khác
    - Tệp tham số không bị hỏng khi tham số sử dụng nhầm dòng mới
    - Trình hướng dẫn bảo trì chỉ báo: Tự động tạo tệp sao lưu có dấu thời gian để dễ dàng khôi phục.
    - Cảnh báo dễ dàng: Bây giờ hiển thị nội dung trường bình luận trong cửa sổ Đầu ra cảnh báo.

Điểm nổi bật của phiên bản 6.40

Phiên bản 6.40 mang đến nhiều chức năng mới, đặc biệt là về ngôn ngữ công thức và hiệu suất. Có hàng trăm tính năng mới và thay đổi đối với chức năng hiện có so với phiên bản 6.30, được liệt kê chi tiết trong tài liệu "Ghi chú phát hành" trong thư mục AmiBroker. Dưới đây chỉ là danh sách ngắn về một số tính năng trong số đó:

• QuickGFX - công cụ kết xuất biểu đồ hoàn toàn mới mang lại hiệu suất tăng lên tới 100 lần so với kết xuất GDI. Tốc độ kết xuất GDI cũng được cải thiện tới 2 lần.

Công nghệ dựng hình trực tiếp QuickGFX do công ty phát triển hoàn toàn bỏ qua Windows GDI và mang lại hiệu suất tăng đáng kinh ngạc từ 10x-100x so với dựng hình GDI

Để bật công nghệ thử nghiệm QuickGFX, hãy vào Tools->Preferences, tab "Miscellaneous", nhấp vào "Experimental: Use QuickGFX render" và nhấn "Apply" hoặc "OK"
Bạn có thể thấy khi công nghệ này được bật, chú thích thời gian biểu đồ (khi bật) sẽ ghi là "QuickGFX render...ms" thay vì "GDI render ...ms"

Đây chỉ là tính năng của phiên bản PROFESSIONAL. Người dùng phiên bản Standard sẽ cần mua bản nâng cấp để sử dụng tính năng này.

- Cải tiến Phân tích tự động
    - Sắp xếp dễ dàng nhiều hành động Phân tích thông qua chuỗi #pragma (quét, khám phá) và nút "Chạy chuỗi" mới mà không cần phải viết hàng loạt. Ví dụ, cho phép chạy phân tích hai bước bằng một cú nhấp chuột tạo ra các biến hợp thành hoặc biến tĩnh ở bước đầu tiên và sử dụng chúng ở bước thứ hai
    - Backtester - thêm tùy chọn "Giá trị vị trí tối đa" trong cài đặt cho phép chỉ định giá trị đô la tối đa của vị trí. Số không (0) nghĩa là không có giá trị tối đa. Các vị trí lớn hơn sẽ bị thu hẹp (nếu bật thu hẹp) hoặc sẽ không được nhập (nếu tắt thu hẹp)
    - tăng tốc đáng kể khi chạy các khám phá lớn, chủ đề tối và tăng tốc gấp 5 lần khi hiển thị các chế độ xem danh sách lớnnhư (so với 6.31)

- Các chức năng/tính năng mới của Ngôn ngữ công thức AmiBroker
    - FindIndex(array, value, start_from = 0, dir = 1) - tìm chỉ mục của mục mảng khớp với giá trị đã chỉ định

- BarsSinceCompare(past, comparison, current) - đối với mọi thanh, hãy so sánh các giá trị mảng trước đó với giá trị thanh hiện tại và trả về số thanh kể từ lần so sánh cuối cùng là đúng
- InternetSetHeaders("headers") - đặt tiêu đề HTTP tùy chỉnh cho các yêu cầu web tiếp theo
- InternetSetOption(option, value) - đặt tùy chọn HTTP cho phiên internet
- InternetPostRequest("http://url_to_your_web_resource", data, flags = 0) - gửi yêu cầu HTTP Post đến tài nguyên web Internet (URL)
- MxCopy() - sao chép tại chỗ các khối hình chữ nhật từ ma trận này sang ma trận kia (sao chép các phần của một ma trận sang ma trận kia)
- _exit() - kết thúc nhẹ nhàng việc thực thi AFL tại thời điểm gọi
- InternetGetStatusCode() - trả về mã trạng thái HTTP của lệnh gọi InternetOpenURL hoặc InternetPostRequest cuối cùng
- Chr(code) trả về chuỗi biểu diễn một ký tự của mã ASCII đã cho
- GetObject(path, class) cung cấp chức năng tương đương với JScript GetObject và VBScript GetObject
- GuiSendKeyEvents("ED") - đăng ký các ký tự đã cho để gửi dưới dạng sự kiện Gui khi nhấn phím bàn phím. GuiGetEvent sẽ trả về code == notifyKeyDown và id của ký tự đã cho, ví dụ id == 'D' cho chữ D
- inverf(x) - hàm nghịch đảo của hàm erf
- erf(x) - tính toán hàm Error https://en.wikipedia.org/wiki/Error_function
- SafeDivide(x, y, valueifzerodiv) - phép chia an toàn xử lý phép chia cho số không bằng cách xử lý đặc biệt (thay thế kết quả bằng giá trị do người dùng xác định)
- công cụ Javascript mới (Chakra) có hỗ trợ JSON gốc (EnableScript("chakra"))

- Cải tiến, nâng cao và sửa lỗi Ngôn ngữ công thức AmiBroker
  - toán tử mũ được thực hiện nhanh hơn 50-100 lần trong trường hợp mũ số nguyên nhỏ là 2, 3, 4 và 5
  - Hàm GuiSetText() tránh gửi thông báo thay đổi để ngăn ngừa vòng lặp cập nhật
  - Hàm ADX được vector hóa (nhanh hơn 2 lần so với trước đây)
  - thêm tham số bổ sung cho Hàm Error() để dừng thực thi
  - chữ ký hàm nội bộ được thay đổi để cho phép sắp xếp linh hoạt các đối số bất kể loại và varargs của chúng với ít chi phí hơn
  - GfxDrawImage với hình ảnh PNG khóa tệp độc quyền đã bị xóa
  - trình phân tích cảnh báo nếu sử dụng phần thân rỗng trong các câu lệnh 'for' hoặc 'while'
  - hỗ trợ nén gzip và deflate trong các hàm Internet
    giao diện plugin tương thích ngược với việc thay đổi chữ ký hàm
  - nhiều lần kiểm tra thời gian chạy hơn để ngăn ngừa lỗi của người dùng và các bản sửa lỗi khác

- Các tính năng/cải tiến mới của cửa sổ hàng loạt
  - Clipboard Cut/Copy/Paste được triển khai trong trình chỉnh sửa hàng loạt
  - Edit->Delete (từ menu chính) và Edit-Delete All được triển khai để hoàn thiện
  - chế độ xem danh sách hiện sử dụng chế độ ảo (dữ liệu chủ sở hữu)
  - thêm hành động "thêm kết quả vào danh sách theo dõi" / WatchlistAddResults
  - thêm hành động "xóa danh sách theo dõi" / WatchlistClear
  - thêm hành động "bình luận"

    - thêm tham số tùy chọn vào lệnh Data Import ASCII để cho phép chỉ định tệp định nghĩa định dạng
    - thêm tham số tùy chọn vào lệnh Execute and Wait để chỉ định thư mục làm việc hiện tại cho lệnh
    - thêm tham số tùy chọn vào Export to File / Export walk-forward to file để chỉ định dấu phân cách cột trong tệp CSV
    - -tham số tùy chọn xác định dấu phân cách cột được sử dụng để xuất. Chỉ sử dụng một ký tự. Khi không cung cấp, dấu phẩy sẽ được sử dụng.

- Cải tiến trình soạn thảo AFL
    - mẹo gọi (mẹo công cụ thông tin tham số) hiện cung cấp mô tả bổ sung về hàm và các tham số của hàm (tính đến v6.39, chỉ có 20 hàm có thông tin bổ sung này)
    - phiên gỡ lỗi sẽ tự động kết thúc với thông báo phù hợp khi người dùng cố gắng chỉnh sửa mã trong quá trình gỡ lỗi (điều này giúp tiết kiệm các lần nhấp chuột cần thiết để dừng phiên gỡ lỗi để chỉnh sửa)
    - trước đây khi không có văn bản nào được chọn và Mã Prettify được chọn, hộp thông báo khiến khung chính được lấy tiêu điểm thay vì giữ tiêu điểm trong khung AFL. Đã sửa
    - khi phát hiện lỗi thời gian chạy, thanh thông báo sẽ hiển thị số lỗi được phát hiện (giống như trước đây trong quá trình "xác minh" bình thường
    - khi sử dụng chế độ xem danh sách chế độ tối, màu văn bản mặc định của cửa sổ xem là màu đen khiến nó hầu như không nhìn thấy được, đã đổi thành màu trắng

- Các tính năng và cải tiến mới của Giao diện người dùng:
    - Chủ đề tối cho tất cả các chế độ xem danh sách do chủ sở hữu vẽ (KHÔNG có trong hộp thoại) hiện có chủ đề tùy chỉnh (hiện có "chủ đề hệ thống (sáng)" và "chủ đề đen") - vào Công cụ->Tùy chỉnh, tab "Giao diện", hộp kiểm "Chế độ tối cho chế độ xem danh sách"
    - Tăng tốc gấp 5 lần để hiển thị các chế độ xem danh sách lớn (bỏ qua kết xuất Windows)
    - Tùy chọn: thêm cài đặt phông chữ Công cụ văn bản độc lập với phông chữ Trục (Tùy chọn->Trang khác)
    - Biểu đồ: công cụ văn bản được cải tiến với kích thước phông chữ có thể chọn theo từng nghiên cứu
    - Tên tham số không bị cắt bớt trong màn hình HighDPI

- Cải tiến cơ sở dữ liệu
    - Cấu trúc ICB 8 chữ số mới được triển khai: https://www.ftserussell.com/data/industry-classification-benchmark-icb
    - Database Purify nhanh hơn 10 lần
    - OLE: thêm hàm Stock.Quotations.Adjust để thực hiện điều chỉnh theo chương trình OLE: thêm hàm Stock.Quotations.Adjust(). long Adjust(BSTR pszFieldList, float fMultiplier,float fOffset, DATE dDateTime, boolean bBefore)

Điểm nổi bật của phiên bản 6.30

Phiên bản 6.30 mang đến nhiều chức năng mới, đặc biệt là về ngôn ngữ công thức và hiệu suất. Có hàng trăm tính năng mới và thay đổi đối với chức năng hiện có so với phiên bản

6.20, được liệt kê chi tiết trong tài liệu "Ghi chú phát hành" trong thư mục AmiBroker. Dưới đây chỉ là danh sách ngắn một số trong số chúng:

- ĐẶC ĐIỂM NỔI BẬT: Các điều khiển GUI trên biểu đồ do người dùng định nghĩa

  18 chức năng AFL mới đã được thêm vào để cho phép tạo giao diện người dùng đồ họa trên biểu đồ do người dùng định nghĩa.

  - GuiButton - tạo nút điều khiển trên biểu đồ (AFL 4.30)
  - GuiCheckBox - tạo hộp kiểm điều khiển trên biểu đồ (AFL 4.30)
  - GuiDateTime - tạo nút điều khiển chọn ngày giờ trên biểu đồ (AFL 4.30)
  - GuiEdit - tạo nút điều khiển chỉnh sửa trên biểu đồ (AFL 4.30)
  - GuiEnable - bật hoặc tắt nút điều khiển trên biểu đồ (AFL 4.30)
  - GuiGetCheck - lấy trạng thái đã kiểm tra của nút điều khiển (AFL 4.30)
  - GuiGetEvent - lấy sự kiện GUI (AFL 4.30)
  - GuiGetText - lấy văn bản từ nút điều khiển trên biểu đồ (AFL 4.30)
  - GuiGetValue - lấy giá trị số của nút điều khiển trên biểu đồ (AFL 4.30)
  - GuiRadio - tạo nút điều khiển radio trên biểu đồ (AFL 4.30)
  - GuiSetCheck - thiết lập trạng thái đã kiểm tra của điều khiển trên biểu đồ (AFL 4.30)
  - GuiSetFont - thiết lập phông chữ cho điều khiển trên biểu đồ (AFL 4.30)
  - GuiSetRange - thiết lập phạm vi điều khiển thanh trượt (AFL 4.30)
  - GuiSetText - thiết lập giá trị văn bản của điều khiển trên biểu đồ (AFL 4.30)
  - GuiSetValue - thiết lập giá trị số của điều khiển trên biểu đồ (AFL 4.30)
  - GuiSetVisible - hiển thị hoặc ẩn điều khiển trên biểu đồ (AFL 4.30)
  - GuiSlider - tạo điều khiển thanh trượt trên biểu đồ (AFL 4.30)
  - GuiToggle - tạo điều khiển nút chuyển đổi trên biểu đồ (AFL 4.30)

- Các tính năng mới trong AmiBroker Formula Language

  - khai báo biến tĩnh
  - truyền biến theo tham chiếu
  - hàm giọng nói mới
    - VoiceSetRate - đặt tốc độ giọng nói (AFL 4.30)
    - VoiceSetVolume - đặt âm lượng giọng nói (AFL 4.30)
    - VoiceWaitUntilDone - đợi cho đến khi giọng nói TTS nói xong (AFL 4.30)
  - xử lý lỗi hệ thống thông qua GetLastOSError
  - hỗ trợ bút hình học trong GfxSelectPen

- Cải thiện hiệu suất đáng kể trong phiên bản 64 bit với việc di chuyển sang trình biên dịch VC++2017 mới

  Tại sao chúng tôi di chuyển sang trình biên dịch mới với phiên bản 64 bit?
  - Trình biên dịch mới hỗ trợ các lệnh CPU mới (SSE3/AVX) mà chúng ta có thể sử dụng để cung cấp hiệu suất tốt hơn
  - Theo các thử nghiệm của chúng tôi, trình biên dịch mới hỗ trợ tự động tạo ra mã nhanh hơn (tối ưu hóa tốt hơn, tự động vector hóa, v.v.)
  - Trình biên dịch mới tốt hơn với chức năng kiểm tra lỗi (ít lỗi lọt qua hơn)

- Chúng ta không cần quan tâm đến khả năng tương thích với các hệ thống trước Vista ở phiên bản 64 bit và tất cả các CPU có khả năng 64 bit đều đủ "hiện đại".

Tại sao chúng ta vẫn sử dụng trình biên dịch cũ ở phiên bản 32 bit?

- Trình biên dịch mới không tạo ra mã tương thích với các hệ điều hành cũ hơn (XP hoặc phiên bản cũ hơn). Trình biên dịch cũ cung cấp khả năng tương thích 100% với tất cả các phiên bản Windows
- Trình biên dịch mới yêu cầu CPU hiện đại

Cải thiện hiệu suất chính xác phụ thuộc vào chức năng và phần cứng. Nhiều chức năng nhanh hơn 30-50% nhưng trong một số trường hợp như các hàm Min()/Max() có thể tăng tốc tới 8 lần trong phiên bản 64 bit.

- Các cải tiến quan trọng khác
  - Khung tối ưu hóa tự động
  - Khả năng tương thích HTML5 trong cửa sổ Nghiên cứu Web
  - Thu gọn bình luận trong trình soạn thảo AFL
  - Liên kết có thể nhấp trong danh sách kết quả Phân tích

Điểm nổi bật của phiên bản 6.20

Phiên bản 6.20 mang đến nhiều chức năng mới, đặc biệt là liên quan đến thử nghiệm hệ thống. Có hàng trăm tính năng mới và thay đổi đối với chức năng hiện có so với phiên bản 6.10, được liệt kê chi tiết trong tài liệu "Ghi chú phát hành" trong thư mục AmiBroker. Dưới đây chỉ là danh sách ngắn một số ít trong số chúng:

- ĐẶC ĐIỂM NỔI BẬT: Bộ xử lý hàng loạt tích hợp - cho phép bạn tự động hóa các tác vụ phức tạp, tuần tự, có
  - giao diện dễ sử dụng
  - trình lập lịch thời gian tích hợp
  - giao diện dòng lệnh
  - kích hoạt AFL
  - ghi nhật ký

- Các tính năng mới trong Ngôn ngữ công thức AmiBroker:
  - các hàm truy cập internet trực tiếp (IntenetOpenURL/InternetReadString/InternetClose) cho phép giao tiếp với các dịch vụ web / REST từ AFL
  - hàm phân phối mục đích chung: PriceVolDisribution
  - nhiều hàm thống kê hơn: NormDist/Kurtosis/Skewness
  - chu kỳ biến đổi PercentRank và StDev
  - các hàm tích lũy CumProd/Prod/ProdSince
  - hàm vẽ ảnh bitmap GfxDrawImage
  - khả năng chọn giọng nói API giọng nói qua VoiceSelect/VoiceCount
  - các cột khám phá nhiều văn bản qua AddMultiTextColumn

- Cải thiện hiệu suất
  - khởi động nhanh hơn

- Cửa sổ Phân tích phản hồi nhanh hơn khi đang bận xử lý (cộng với chức năng Tạm dừng mới)

- Cải thiện giao diện người dùng và khả năng sử dụng
  - nhiều điều chỉnh trong việc mở rộng giao diện người dùng để hỗ trợ tốt hơn màn hình HiDPI/4K
  - cài đặt chuyển tiếp hiện là cho từng dự án, không phải toàn cục
  - cửa sổ theo dõi của trình gỡ lỗi hỗ trợ hiển thị các biến tĩnh và biểu thức có biến tĩnh

Điểm nổi bật của phiên bản 6.10

Phiên bản 6.10 mang đến nhiều chức năng mới, đặc biệt là về thử nghiệm hệ thống. Có hàng trăm tính năng mới và thay đổi đối với chức năng hiện có so với phiên bản 6.00, được liệt kê chi tiết trong tài liệu "Ghi chú phát hành" trong thư mục AmiBroker. Dưới đây chỉ là danh sách ngắn một số trong số chúng:

- ĐẶC ĐIỂM: Trình gỡ lỗi trực quan tích hợp - có
  - thực thi một bước
  - điểm dừng
  - đồng hồ do người dùng xác định với trình đánh giá biểu thức
  - chế độ xem mảng với tính năng tô sáng thay đổi cho từng mục
  - chú giải công cụ kiểm tra giá trị
  - cửa sổ đầu ra (cho đầu ra printf)

- ĐẶC ĐIỂM: Hỗ trợ Ma trận nâng cao
  - đảo ngược ma trận (thông qua phép loại bỏ Gauss-Jordan)
  - giải hệ phương trình tuyến tính (bao gồm mã phù hợp đa thức ví dụ)
  - định thức
  - truy cập khối nhanh
  - sắp xếp cột/hàng ma trận
  - chuyển đổi ma trận từ/sang chuỗi
  - tính bền bỉ (lưu/tải ma trận từ/vào đĩa thông qua cơ chế biến tĩnh)

- Các tính năng mới trong Trình chỉnh sửa AFL:
  - Dấu trang
  - Tìm trong Tệp
  - Chặn bình luận/bỏ bình luận
  - Liên kết có thể nhấp vào đến tài liệu/tệp bên ngoài trong bình luận

- Các tính năng mới trong AmiBroker Formula Language
  - tổng hợp nhanh hơn bằng StaticVarAdd
  - nén các biến tĩnh liên tục (có thể tiết kiệm tới 90% dung lượng đĩa và bộ nhớ)
  - kiểm tra lỗi nghiêm ngặt hơn (kiểm tra chuỗi định dạng printf/StrFormat)
  - các hàm biến tĩnh và động hiện hỗ trợ ma trận
  - các hàm mới GfxFillSolidRect, SumSince, MxInverse, MxSolve, MxDet, MxSetBlock, MxGetBlock, MxSort, MxSortRows, MxToString, MxFromString, StaticVarAdd, fgetcwd
  - các tham số mới trong các hàm: DateTimeToStr, StaticVarSet

    - cải thiện hiệu suất trong printf(), biến period Sum

- Cải thiện biểu đồ, giao diện người dùng và các cải tiến bên trong
    - cửa sổ danh mục được cải tiến (cho phép sắp xếp lại các thị trường/nhóm/ngành/ngành công nghiệp lên/xuống và xem danh sách)
    - độ dày bấc nến do người dùng định nghĩa
    - số thập phân do người dùng định nghĩa trong mức giá đường ngang
    - công cụ chu kỳ xử lý các năm sau năm 2038
    - các biểu tượng lớn được sử dụng trên màn hình HighDPI cho chế độ xem cây và hộp kiểm chế độ xem danh sách để có khả năng truy cập tốt hơn nhiều
    - tự động thay đổi kích thước các cột Phân tích thành nội dung
    - cải tiến phục hồi sự cố
    - hàng tấn cải tiến khác (xem Ghi chú phát hành để biết chi tiết)

Điểm nổi bật của phiên bản 6.00

Phiên bản 6.00 mang đến nhiều chức năng mới, đặc biệt là liên quan đến thử nghiệm hệ thống. Có hàng trăm tính năng mới và thay đổi đối với chức năng hiện có so với phiên bản 5.90, được liệt kê chi tiết trong tài liệu "Ghi chú phát hành" trong thư mục AmiBroker. Dưới đây chỉ là danh sách ngắn về một số tính năng trong số đó:

• Trình mô phỏng Monte Carlo hiệu suất cao tích hợp - với biểu đồ phân phối tích lũy của vốn chủ sở hữu, mức giảm tối đa, hỗ trợ các số liệu tùy chỉnh do người dùng định nghĩa và khả năng thực hiện tối ưu hóa do trình mô phỏng MC điều khiển.
• Hỗ trợ Ma trận đầy đủ (mảng hai chiều) trong AFL với phép tính số học ma trận gốc trực tiếp (các phép toán ma trận như cộng, trừ, nhân, chia, chuyển vị, v.v.), hãy xem Ma trận, MxIdentity, MxTranspose, MxGetSize
• Thống kê Mua và nắm giữ chi tiết (chuẩn) được tự động thêm vào báo cáo kiểm tra ngược
• Thứ tự dừng có thể xác định của người dùng (hàm SetStopPrecedence) và tính hợp lệ dừng
(các tham số ValidFrom/ValidTo trong hàm ApplyStop)
• Hỗ trợ mảng thưa thớt: SparseCompress, SparseExpand
• Hàm lọc Phản hồi xung vô hạn (IIR) để triển khai hiệu quả các thuật toán làm mịn bậc cao
• Đầu ra văn bản thô trong các khám phá thông qua hàm AddRow
• Các kiểu mới được Exploration XYCharts hỗ trợ
• Hàm Phần trăm chu kỳ biến đổi
• Hỗ trợ Unicode (UCN) trong PlotText, PlotTextSetFont, GfxDrawText, GfxTextOut, biểu đồ tiêu đề, diễn giải và cửa sổ chú thích (cho phép nhiều chú thích đồ họa / quanh co)
• Các hàm đồ họa cấp thấp mới: GfxSelectHatchBrush, GfxSelectStockObject
• Hàm so khớp ký tự đại diện StrMatch
• Assignment Organizer được cải tiến
• Chức năng ngắt dòng trong trình soạn thảo AFL và hàm "Code Prettify" được cải tiến

Điểm nổi bật của phiên bản 5.90

Ngoài chức năng hoàn toàn mới, phiên bản này tập trung vào cải thiện tốc độ và nâng cao chức năng hiện có. Có hàng trăm tính năng mới và thay đổi đối với chức năng hiện có so

với phiên bản 5.80, được liệt kê chi tiết trong tài liệu "Ghi chú phát hành" trong thư mục AmiBroker. Dưới đây chỉ là danh sách ngắn về một số tính năng trong số đó:

- Cải thiện hiệu suất
    - AFL Engine: bộ phân bổ bộ nhớ tùy chỉnh không sử dụng thư viện thời gian chạy của Microsoft để theo dõi tham chiếu nữa. Kết quả - các công thức phức tạp với nhiều vòng lặp và OLE (đặc biệt là các thử nghiệm ngược tùy chỉnh cấp thấp) chạy nhanh hơn tới 3 lần trong 32 bit và nhanh hơn tới 4 lần trong 64 bit
    - tốc độ thực thi được cải thiện theo hệ số > 2x đối với các hàm AFL: MACD, Signal, CCI, Sum (giai đoạn biến đổi)

- Cửa sổ Code Snippets hoàn toàn mới và trình kích hoạt bàn phím
    - thêm cửa sổ Code Snippets - cho phép chèn/xóa/lưu các phần đã chọn của công thức dưới dạng đoạn mã. Ngoài ra còn có chức năng kéo thả đoạn mã tiện lợi vào cửa sổ chỉnh sửa công thức
    - Đoạn mã có sẵn trong danh sách tự động hoàn thành (nhập @ cộng với chữ cái đầu tiên của phím kích hoạt đoạn mã) và ngay cả khi không kích hoạt tự động hoàn thành thì @keytrigger vẫn được thay thế bằng văn bản đoạn mã

- Report Explorer được thiết kế lại và Report Viewer được cải thiện (HTMLView)
    - Bố cục cột (thứ tự và kích thước) hiện được lưu và khôi phục giữa các lần chạy
    - Hiệu suất tải và làm mới được cải thiện đáng kể (gấp 5 lần) khi sử dụng chế độ vẽ/ảo của chủ sở hữu
    - Sắp xếp nhiều cột được triển khai
    - Các cột số hiện được căn phải để dễ đọc hơn
    - Hình ảnh được cải thiện đáng kể (danh sách sử dụng kiểu hiện đại, đường lưới, thay đổi kích thước cột ngay lập tức, đệm đôi để không nhấp nháy, dấu phân cách nghìn, giá trị âm được hiển thị bằng màu đỏ đậm, nhận biết HighDPI, thanh công cụ đã thay đổi)
    - HTMLView - Trình xem báo cáo kiểm tra ngược - đã thêm Chỉnh sửa/Sao chép, Chỉnh sửa/Chọn tất cả và Chỉnh sửa/Sao chép BẢNG. Lệnh cuối cùng chuyển đổi các bảng HTML sang định dạng CSV và sao chép vào bảng tạm để có thể dễ dàng dán các bảng vào Excel. Ngoài ra, nó chia các cột Nhập/Thoát thành các cột Ngày nhập/thoát/Giá riêng biệt

- Chỉ báo xu hướng Giá thầu/Giá chào bán mới trong cửa sổ Báo giá theo thời gian thực - một chỉ báo đồ họa hiển thị hướng của 10 thay đổi gần đây nhất trong giá thầu/giá chào bán theo thời gian thực. Hộp ngoài cùng bên phải là hộp gần đây nhất và khi báo giá thầu/giá chào bán mới đến, chúng sẽ được chuyển sang bên trái.
- Báo cáo kiểm tra ngược HTML do người dùng định nghĩa
    - giờ đây có thể xuất HTML thay vì đồ họa trong công thức biểu đồ báo cáo bằng AFL: EnableTextOutput(3) - Đầu ra HTML vào báo cáo kiểm tra ngược
    - viết lại 3. Profit Table.afl sử dụng các tính năng nhúng HTML, bố cục tự động mở rộng (do đó, nó sẽ mở rộng khi số lớn hơn), các cột tóm tắt in đậm, các giá trị âm màu đỏ, ngày ranh giới được thay đổi thành ngày cuối cùng của năm/tháng

- Cải tiến biểu đồ
    - Các đường xu hướng mở rộng trái/phải và Tia hiện sử dụng Hệ số mở rộng do người dùng định nghĩa (trường mới trong Thuộc tính nghiên cứu) thay vì luôn mở rộng vô hạn. Mở rộng Hệ số bằng ZERO có nghĩa là VÔ HẠN, các giá trị khác 0,1 ... 26 xác định độ dài của đường bên trái/phải được mở rộng
    - Có thể đạt được mức thu phóng tối đa thông qua View->Zoom Out lên đến 5 triệu thanh, cũng như giới hạn thu phóng Pref/Charting/Default được đặt thành 5 triệu.
    - Bản vẽ đường thẳng hiện có chiều rộng đường thẳng do người dùng xác định theo pixel (trường "Chiều rộng đường thẳng" mới trong hộp thoại Thuộc tính nghiên cứu). Ngoài ra, hộp "Đường thẳng dày" làm cho đường thẳng rộng gấp đôi (do đó chiều rộng thực tế của đường thẳng dày là 2 * lineWidth thay vì thêm 1 pixel vào chiều rộng)
    - Thêm khả năng kiểm soát số thập phân trong nhãn giá trị biểu đồ thông qua biến GraphLabelDecimals (ví dụ, thêm GraphLabelDecimals = 2; vào công thức sẽ cung cấp cho bạn nhãn giá trị có 2 chữ số thập phân)

- Cải tiến Giao diện người dùng
    - Cải thiện giao diện cửa sổ tham số. Chiều cao mục được tăng lên và thanh trượt ngón tay cái được làm rộng hơn để dễ sử dụng hơn trên màn hình có kích thước nhỏ/DPI cao
    - Giao diện Phân tích mới làm mới nhanh hơn
    - Màu sắc, kiểu in đậm và in nghiêng hiện được thêm vào cửa sổ Diễn giải và Bình luận
    - Hộp thoại Bộ lọc hiện hiển thị số ký hiệu khớp theo thời gian thực
    - Hộp thoại Thiết lập cột có các nút Đánh dấu tất cả/Chuyển đổi tất cả mới

- Các tính năng/cải tiến mới của AFL:
    - Các hàm AFL mới: GetFormulaPath, NullCount, Sort, Reverse, StrSort, StrTrim, SendEmail
    - Chức năng mở rộng của các hàm AFL: StrExtract, StrMid, RestorePriceArrays, PlotGrid, EnableTextOutput, GetOption
    - Các ký tự đơn được thêm vào AFL

- Cải thiện tính ổn định và gỡ lỗi
    - Thêm nhiều kiểm tra tham số
    - Phiên bản 64 bit hiện có theo dõi ngăn xếp cuộc gọi trong báo cáo lỗi để gỡ lỗi tốt hơn
    - Thêm nhiều kiểm tra bộ nhớ hơn, cảnh báo sớm và thông báo lỗi khi hết bộ nhớ
    - đã thêm các kiểm tra cho các tình huống bất thường nhưng có khả năng 'gây rắc rối'

Điểm nổi bật của phiên bản 5.80

Ngoài chức năng hoàn toàn mới, phiên bản này tập trung vào các cải tiến và nâng cao gia tăng của chức năng hiện có. Có hàng trăm tính năng mới và thay đổi đối với chức năng hiện có so với phiên bản 5.70, được liệt kê chi tiết trong tài liệu "Ghi chú phát hành" trong thư mục AmiBroker. Dưới đây chỉ là danh sách ngắn một số trong số chúng:

- AFL Formula Editor được viết lại hoàn toàn mới, hỗ trợ các tính năng sau:
    - Cải thiện tính năng tô sáng cú pháp
    - Tự động khớp/tô sáng dấu ngoặc nhọn (MỚI)
    - Tự động thụt lề (MỚI)
    - Đánh dấu thụt lề (MỚI)
    - Tự động hoàn thành ở hai chế độ (ngay lập tức (MỚI) và theo yêu cầu)
    - Thông tin tham số
    - Lề đánh số dòng và lề lựa chọn (MỚI)
    - Gấp mã (MỚI)
    - Báo cáo lỗi trong dòng (MỚI)
    - Giao diện người dùng có tab mới với khả năng hoạt động ở cả chế độ MDI và khung nổi riêng biệt, có thể di chuyển ra sau màn hình AmiBroker chính và đưa trở lại (Cửa sổ->Chuyển đổi khung) (MỚI) hoặc giữ ở trên cùng (Cửa sổ->Giữ ở trên cùng)
    - Sao chép/dán/xóa khối hình chữ nhật (Sử dụng chuột và giữ phím Alt bên trái để đánh dấu khối hình chữ nhật)
    - In ấn được cải tiến (có tô sáng cú pháp và tiêu đề/chân trang)

- Mã snippets - đây là những đoạn mã AFL nhỏ có thể tái sử dụng. Chúng có thể được chèn bằng cách nhấp chuột phải vào cửa sổ trình soạn thảo AFL và chọn menu "Insert Snippet". Các đoạn mã có thể được người dùng xác định.
- Các tính năng mới trong Đồ họa cấp thấp
    - nhiều lớp Z-order GfxSetZOrder
    - tọa độ hiện có thể được cung cấp ở cả chế độ pixel và chế độ giá thanh GfxSetCoordsMode
    - cải thiện tốc độ (lên đến 3x)

- Biến tĩnh liên tục - StaticVarSet/StaticVarSetText (thêm tham số 'liên tục')
- Cải tiến phân tích (Kiểm tra ngược/Tối ưu hóa)
    - cài đặt "Giao dịch bằng chuyển đổi tiền mặt FX" mới
    - Sửa lỗi công cụ SPSO/Tribes 64 bit

- Chức năng AFL mới/được cải tiến
    - PlotTextSetFont (MỚI)
    - GfxSetCoordsMode (MỚI)
    - GfxSetZOrder (MỚI)
    - GfxGetTextWidth (MỚI)
    - fopen (thêm tham số 'chia sẻ')
    - StaticVarSet/StaticVarSetText (thêm Tham số 'persistent')
    - SetOption (tùy chọn mới "StaticVarAutoSave")
    - SetChartOptions (cờ mới chartDisableYAxisCursor, chartDisableTooltips)
    - PlotText (tham số mới yoffset)

Điểm nổi bật của phiên bản 5.70

Ngoài chức năng hoàn toàn mới, phiên bản này tập trung vào các cải tiến và nâng cao gia tăng của chức năng hiện có. Có 116 tính năng mới và thay đổi đối với chức năng hiện có so với phiên bản 5.60, được liệt kê chi tiết trong tài liệu "Ghi chú phát hành" trong thư mục AmiBroker. Dưới đây chỉ là danh sách ngắn một số trong số chúng:

- Cải tiến phân tích:
    - Tối ưu hóa cá nhân đa luồng mới
    - Các hàm xếp hạng mục đích chung mới

    - StaticVarGenerateRanks/StaticVarGetRankedSymbols
    - Các cột xếp hạng do người dùng định nghĩa (thông qua hàm AddRankColumn)
    - Nhiều cải tiến nội bộ trong công cụ kiểm tra ngược/tối ưu hóa
    - Các công cụ tối ưu hóa SPSO, Tribes hiện cũng có sẵn ở phiên bản 64 bit

- Cải tiến Time&Sales: lọc do người dùng định nghĩa, màu do người dùng định nghĩa, 2 chế độ hiển thị do người dùng lựa chọn
- Cải tiến cơ sở dữ liệu:
    - Phiên bản 64 bit hỗ trợ các tệp lớn hơn 2 GB cho mỗi ký hiệu
    - bộ nhớ đệm trong bộ nhớ có thể chứa tới 100.000 ký hiệu (tăng từ 20K)
    - plugin DDE và ODBC 64 bit mới

- Cải tiến biểu đồ:
    - Hiệu suất được cải thiện đáng kể Công nghệ QuickData được triển khai giúp giảm mức sử dụng CPU cho biểu đồ
    - Edit->'Paste Special' cho phép sao chép-dán toàn bộ khung biểu đồ với nhiều tùy chọn khác nhau
    - Đo khoảng cách khi vẽ đường xu hướng (khoảng cách X, Y trên thanh trạng thái)
    - Ràng buộc X/Y cho các công cụ vẽ (nhấn phím X và/hoặc Y để ràng buộc chuyển động theo hướng X hoặc Y khi vẽ)
    - Trình nhập ASCII bổ sung hỗ trợ cho dấu thời gian mili giây

- Cải tiến AFL:
    - các chức năng mới:
        - StaticVarGenerateRanks, StaticVarGetRankedSymbols - xếp hạng do người dùng định nghĩa chung
        - Error - hiển thị thông báo lỗi do người dùng định nghĩa (cũng hữu ích cho các nhà phát triển plugin)
        - fdir - danh sách thư mục
        - CategoryCreate - tạo danh sách theo dõi theo chương trình
        - AddRankColumn - xếp hạng các cột trong khám phá

    - hiệu suất được cải thiện cho Percentile() (thứ tự độ lớn)
    - các trường mới được hỗ trợ trong GetFnData
    - Hỗ trợ XShift được thêm vào PlotShapes
    - tăng tốc các hàm toán học siêu việt (sqrt, sin, asin, cos, acos, tan, atan, ln, log10, v.v.)
    - cải thiện SetSortColumns

- trong Report Explorer là mục cuối cùng và có kiểu "PS".

Điểm nổi bật của phiên bản 5.60

• Kết xuất GDI (đồ họa) đa luồng - giờ đây mọi bản vẽ (kết xuất đồ họa) đều được thực hiện trong các luồng công việc riêng biệt nên giao diện người dùng phản hồi nhanh hơn nhiều và biểu đồ được cập nhật nhanh hơn và hoàn toàn độc lập với nhau.
• Báo cáo tóm tắt tự động chuyển tiếp ngoài mẫu - giờ đây mỗi bước ngoài mẫu sẽ tạo ra báo cáo riêng lẻ, cộng với báo cáo tóm tắt mới bao gồm tất cả các bước ngoài mẫu. Có thể nhìn thấy Báo cáo kiểm tra ngược được mã hóa màu nâng cao

• Biểu đồ XY (phân tán) trong các cuộc khám phá
• Chủ đề biểu đồ và giao diện biểu đồ được cải thiện (đặc biệt là lưới)
• Thiết lập và cập nhật tự động chỉ bằng một cú nhấp chuột cho danh sách cổ phiếu, phân công ngành và lĩnh vực cho tất cả các sàn giao dịch lớn của Hoa Kỳ
• Khoảng thời gian biểu đồ tùy chỉnh không giới hạn bằng hộp kết hợp Khoảng thời gian mới chấp nhận bất kỳ khoảng thời gian nào được nhập thủ công
• hỗ trợ các danh mục ICB (Điểm chuẩn phân loại ngành) trong AFL, UI, trình nhập ASCII và giao diện OLE
• Biểu đồ vùng Gradient gốc
• Đường siêu dày trong Plot, PlotOHLC, PlotForeign
• Các hàm AFL mới: GetAsyncKeyState, InIcb, IcbID, StaticVarInfo, SetGradientFill, XYChartAddPoint, XYChartSetAxis
• Các hàm AFL được cập nhật với chức năng mới: Status, CategoryGetSymbols, CategoryGetName, CategorySetName, CategoryAddSymbol, CategoryRemoveSymbol, CategoryFind, Plot, PlotOHLC, PlotForeign
• plugin eSignal 64-bit mới

• Giao diện người dùng được cập nhật ở nhiều nơi
• nhiều cải tiến khác (xem Ghi chú phát hành để biết chi tiết)

Điểm nổi bật của phiên bản 5.50

• Cửa sổ Phân tích mới được giới thiệu trong phiên bản 5.50 mang đến những cải tiến sau so với Phân tích tự động cũ
  - hoạt động đa luồng = tốc độ - cửa sổ Phân tích mới sử dụng tất cả CPU/lõi có sẵn để thực thi các công thức trong nhiều luồng song song, giúp tăng tốc đáng kể. Ví dụ, trên Intel i7 4 lõi có thể chạy tới 8 luồng, nó có thể chạy nhanh hơn tới 8 lần so với cửa sổ Phân tích cũ. Tốc độ tăng chính xác phụ thuộc vào độ phức tạp của công thức (công thức càng phức tạp thì tốc độ càng cao), lượng dữ liệu được xử lý (truy cập RAM có thể không nhanh bằng CPU do đó hạn chế khả năng tăng tốc).
  - hoạt động không chặn - giờ đây bạn có thể xem, cuộn và sắp xếp kết quả phân tích trong khi chúng vẫn được tạo, ngoài ra vì luồng giao diện người dùng không được sử dụng để xử lý cho hầu hết các phần, biểu đồ và các phần chương trình điều khiển GUI khác phản hồi nhanh hơn nhiều so với phân tích tự động cũ
  - nhiều phiên bản - bạn có thể chạy nhiều phiên bản Phân tích mới cùng một lúc, do đó bạn có thể chạy nhiều lần quét/kiểm tra ngược/khám phá/tối ưu hóa song song mà không cần chờ một lần hoàn tất
  - giao diện người dùng mượt mà hơn - Cửa sổ Phân tích mới có thể hoạt động như tài liệu có tab, có thể nổi,
  - các nút có thể được sắp xếp lại để quy trình làm việc tốt hơn. Có nhiều không gian hơn cho danh sách kết quả, thông tin bổ sung về quá trình thực hiện được cung cấp trên tab "Thông tin" mới. Ngoài ra, kết quả chuyển tiếp hiện được hiển thị trong cửa sổ Phân tích mới để ít lộn xộn hơn.
• Biểu đồ thứ hạng cao-thấp nhỏ trong cửa sổ báo giá theo thời gian thực
• Biểu đồ thanh nhỏ do người dùng xác định trong Explorations (xem hàm AddColumn)
• Tính năng Thêm cột thứ hạng - nhấp chuột phải vào danh sách kết quả phân tích và chọn "Thêm cột thứ hạng" - tính năng này sẽ thêm một cột có thứ hạng theo thứ tự dựa trên thứ hạng hiện tại hoặc chỉ cột số hàng khi danh sách không được sắp xếp
• Kiểm tra ngược tài khoản IRA thông qua tính năng SettlementDelay (xem hàm SetOption)
• Thuật toán thanh phạm vi được cải thiện đáng kể
• Các hàm AFL mới: ThreadSleep, StaticVarCompareExchange

• Các hàm AFL được cập nhật với chức năng mới: AddColumn, SetOption, GetOption, CategoryGetSymbols, PopupWindow, GetFnData, ClipboardSet
• Giao diện OLE được cập nhật để hỗ trợ cửa sổ Phân tích mới
• Giao diện kiểm tra ngược tùy chỉnh được cập nhật để hỗ trợ truy cập vào thuộc tính EquityArray cục bộ, theo từng phân tích
• Giao diện người dùng được cập nhật ở nhiều nơi
• Chế độ xem danh sách do chủ sở hữu vẽ cho Cải thiện tốc độ gấp 10 lần khi hiển thị hàng triệu hàng
• nhiều cải tiến khác (xem Ghi chú phát hành để biết chi tiết)

Điểm nổi bật của phiên bản 5.40

• Biểu đồ đa luồng hoàn toàn. Thực thi AFL song song hàng loạt (mỗi khung biểu đồ chạy trong luồng riêng biệt) cho phép tối đa hóa tốc độ và khả năng sử dụng máy tính đa lõi/đa CPU hiện đại. Ví dụ, trên CPU Intel i7 8 lõi, biểu đồ của bạn sẽ chạy nhanh hơn tới 8 lần so với phiên bản
5.30. Công cụ AFL đã được viết lại hoàn toàn từ đầu để cho phép nhiều phiên bản của công cụ chạy đồng thời. Điều này không chỉ cho phép đa luồng mà còn tăng cường khả năng phản hồi của toàn bộ ứng dụng, vì ngay cả công thức người dùng viết kém được sử dụng trong biểu đồ cũng không thể khóa hoặc làm chậm phần còn lại của chương trình. Đa luồng được BẬT theo mặc định. Có thể tắt bằng cách bỏ chọn hộp "Biểu đồ đa luồng" trong Công cụ->Tùy chọn, tab "AFL" nhưng điều này không được khuyến khích. Đa luồng phải được BẬT nếu bạn muốn AmiBroker hoạt động ở tốc độ tối đa.
• 12 hàm AFL mới
  - DateTimeAdd - thêm số giây/phút/giờ/ngày đã chỉ định vào datetime
  - HMA - Trung bình động Hull
  - FIR - Bộ lọc phản hồi xung hữu hạn
  - PercentRank - tính thứ hạng phần trăm
  - Lookup - tìm kiếm mảng cho thanh có ngày/giờ đã chỉ định
  - FirstVisibleValue - lấy giá trị hiển thị đầu tiên của mảng
  - LastVisibleValue - lấy giá trị hiển thị cuối cùng của mảng
  - InGICS - kiểm tra xem ký hiệu đã cho có thuộc danh mục GICS đã chỉ định hay không
  - GicsID - lấy thông tin về danh mục GICS
  - PlaySound - phát tệp âm thanh .WAV
  - ShellExecute - thực thi chương trình/tệp bên ngoài
  - _DT - từ đồng nghĩa của StrToDateTime
• Cải tiến và sửa lỗi của Quote Editor: cho phép người dùng bật/tắt dịch chuyển thời gian và chỉnh sửa dấu thời gian xuống đến mili giây, xử lý cố định múi giờ 12 giờ (AM/PM) thiết lập
• Cải tiến biểu đồ: nhãn giá trị đẹp hơn, các hàm đồ họa cấp thấp được tăng tốc gấp 4 lần.
• Thay đổi liên quan đến biểu đồ
  - Cửa sổ dữ liệu và công cụ đọc chú giải dữ liệu ngay lập tức và không yêu cầu thực thi AFL bổ sung (các giá trị cần thiết để hiển thị chúng được lưu trữ trong RAM trong quá trình làm mới biểu đồ bình thường và có sẵn mà không cần chạy lại công thức). Biến chú giải hiện đã lỗi thời. Để hiển thị các giá trị tùy chỉnh trong chú giải mà không cần vẽ đường, bạn có thể sử dụng Plot() với cờ styleHidden.
  - Hiển thị giải thích không yêu cầu thực thi AFL
  - Chèn chỉ báo và đặt lại tham số nhanh hơn gấp bội
  - cài đặt thu phóng biểu đồ hiện được lưu trong tệp bố cục và được khôi phục khi tải bố cục
• Cải tiến giao diện OLE (phương thức IsBusy mới của đối tượng Analysis và phương thức Import tự động làm mới UI)
• Cải tiến công cụ AFL:

- thêm cảnh báo phát hiện lỗi tiềm ẩn của người dùng - chẳng hạn như gán trong biểu thức điều kiện hoặc gọi hàm Plot() dư thừa
- thêm kiểm tra bổ sung cho các giá trị tham số không hợp lệ cho nhiều hàm và chỉ số mảng == Null - thông báo lỗi thích hợp được hiển thị
- lotText được tối ưu hóa để tiết kiệm bộ nhớ và giảm thời gian thực thi bằng cách bỏ qua các phần vô hình

• Sửa lỗi trình quản lý tài khoản
• cải thiện khả năng tương thích với Windows 7 (bản kê khai nhận biết DPI cao, khả năng tương thích với Internet Explorer 9)
• Sửa lỗi dành riêng cho 64 bit (bao gồm sửa lỗi với DLL của bên thứ 3)
• chủ đề giao diện mới và nhiều cải tiến và sửa lỗi khác

Điểm nổi bật của phiên bản 5.30

• thay đổi định dạng cơ sở dữ liệu để hỗ trợ độ chi tiết của dấu thời gian xuống còn một micro giây (0,000001 giây) và nhiều trường dữ liệu hơn
• biến mảng tĩnh
• biểu đồ báo cáo kiểm tra ngược do người dùng xác định (xem ví dụ trong cửa sổ Biểu đồ, thư mục "Biểu đồ báo cáo")
• Cửa sổ dữ liệu mới (Cửa sổ->Cửa sổ dữ liệu)
• công cụ Giám sát hiệu suất mới
• thống kê tích tắc được thêm vào cửa sổ Thời gian & Doanh số
• phần mở rộng vùng trống biểu đồ bằng phím END (trên bàn phím), để khôi phục cài đặt gốc, hãy nhấn phím HOME
• thêm tùy chọn để yêu cầu khai báo biến (SetOption("RequireDeclarations", True));
• trạng thái cột cố định (chiều rộng/thứ tự/khả năng hiển thị) trong Phân tích tự động và tất cả các chế độ xem danh sách khác.
• khả năng biểu đồ vùng gradient (xem Biểu đồ - Biểu đồ cơ bản - Biểu đồ giá gradient)
• toán tử AFL typeof() mới
• hàm AFL mới
  - ColorBlend
  - DateTimeDiff
  - HighestVisibleValue
  - LowestVisibleValue
  - StaticVarCount
• Thứ tự Z do người dùng xác định của các bản vẽ và biểu đồ chỉ báo
• đệm dữ liệu tùy chọn cho những ngày không giao dịch
• Các công cụ vẽ hình chữ nhật và hình elip hiện là khối theo mặc định
• Thêm nhãn tọa độ X-Y (sử dụng menu View->X-Y Labels để hiển thị/ẩn).
• Hỗ trợ SSL (kết nối an toàn) và lựa chọn cổng TCP/IP cho cảnh báo qua email đã được thêm vào
• cửa sổ Biểu tượng mới với tìm kiếm và sắp xếp toàn văn cực nhanh
• hỗ trợ hệ thống danh mục 4 cấp GICS
• đặt lệnh trực tiếp từ biểu đồ (Interactive Brokers)
• nhiều cải tiến và bản sửa lỗi khác (xem tài liệu Ghi chú phát hành để biết chi tiết)

Điểm nổi bật của phiên bản 5.20

• Tối ưu hóa hệ thống giao dịch thông minh (không đầy đủ)

AmiBroker hiện được cung cấp với 3 thuật toán tối ưu hóa tiến hóa không đầy đủ:
SPSO (Standard Particle Swarm Optimizer)
TRIBES (Advances Particle Swarm)
CMA-ES (Chiến lược tiến hóa thích ứng ma trận hiệp phương sai

• Hỗ trợ cho các chiến lược cân bằng dài-ngắn trung lập với thị trường thông qua kiểm soát MaxOpenLong/MaxOpenShort và xếp hạng dài/ngắn riêng biệt trong backtester
• Tối ưu hóa hiệu suất trong công cụ vẽ biểu đồ - biểu đồ nhanh hơn gấp bội khi số thanh hiển thị lớn hơn nhiều so với số pixel.
• Cửa sổ nhật ký được triển khai - cho phép theo dõi và báo cáo lỗi thời gian chạy
• QuickAFL được triển khai trong Automatic Analysis - tăng tốc backtest, tối ưu hóa và khám phá gấp 2 lần (nếu phạm vi nhỏ hơn tất cả các báo giá). (Lưu ý: để bật tính năng này, bạn cần đánh dấu vào ô "Sử dụng QuickAFL" trong cài đặt Automatic Analysis).
• Biểu đồ Volume-At-Price nhiều đoạn (thông qua hàm PlotVAPOverlayA)
• AmiBroker 32 bit hiện là LARGEADDRESSAWARE, tức là hiện có thể sử dụng tới 4GB RAM
• Khoảng thời gian theo Quý và Năm tích hợp

• Tự động tóm tắt các hàng trong các cuộc khám phá (thông qua hàm AFL AddSummaryRows)
• Cải tiến và nâng cao biểu đồ
- Xử lý tốt hơn các công cụ vẽ
- Chế độ nam châm tốt hơn
- Múi giờ Fibonacci hiện bao gồm các dòng 144 và 233
- Cải thiện khả năng phóng to thông qua thanh cuộn

• Thanh phạm vi hiện sử dụng TickSize theo ký hiệu làm đơn vị
• Các hàm AFL mới:
GetChartBkColor
CategorySetName
PlotVAPOverlayA
AddSummaryRows
DaysSince1900
OptimizerSetEngine
OptimizerSetOption
StrCount

• Cải tiến hiệu suất AFL trong LinearReg, LinRegSlope, LinRegIntercept, TSF và StdErr, các hàm Day(), Month(), Year(), DaysSince1900(), DayOfWeek(), DayOfYear() (nhanh hơn theo cấp số nhân)
• Các hàm AFL được cải thiện: lệnh Say() được xếp hàng (chuyển văn bản thành giọng nói), cải thiện StrExtract() - giờ có thể tham chiếu đến các mục đếm từ cuối
• Các plugin dữ liệu thời gian thực được cập nhật (phiên bản IB 1.2.4, phiên bản eSignal 1.9.0), IBController được cập nhật để hỗ trợ các thay đổi mới nhất trong các nguồn dữ liệu

Điểm nổi bật của phiên bản 5.10

• Kiểm tra Walk-Forward tự động (kỹ thuật xác thực và tối ưu hóa hệ thống giao dịch)

• Cửa sổ nổi (khả năng lập biểu đồ đa màn hình THỰC SỰ)
khả năng "gỡ neo" (hoặc "nổi") cửa sổ biểu đồ và di chuyển đến màn hình riêng biệt.
Tất cả mã bố cục cũng được cập nhật để lưu và khôi phục chính xác các thiết lập biểu đồ đa màn hình http://www.amibroker.com/video/FloatAndLink.html

• Liên kết biểu tượng và khoảng thời gian
nhiều biểu đồ hiện có thể được liên kết theo biểu tượng và/hoặc theo khoảng thời gian bằng các liên kết được mã hóa màu dễ sử dụng

• AFL Code Profiler - hiển thị phân tích mã với báo cáo thời gian chi tiết theo từng chức năng (AFL Editor: Tools->Code Check & Profile menu)

• Cải tiến cửa sổ báo giá theo thời gian thực
- sắp xếp lại các biểu tượng trong báo giá RT bằng cách kéo và thả
- nhập trực tiếp các biểu tượng vào cửa sổ báo giá RT
- khả năng tách các nhóm biểu tượng bằng cách chèn dòng trống
- làm mới nhanh hơn và làm mờ màu nền nhiều giai đoạn khi thay đổi báo giá

• các chức năng AFL mới/được cải thiện
- (mới) SetBarFillColor
- (đã cải thiện) GetCursorXPosition
- (đã cải thiện) GetCursorYPosition
- (đã cải thiện) GetCursorMouseButtons
- (đã cải thiện) SetChartOptions
- (đã cải thiện) SetOption
- (đã cải thiện) Status

• Cải thiện tốc độ kiểm tra ngược/tối ưu hóa (lên đến 2 lần trong một số trường hợp so với v5.00)

• cải thiện đường ngắm biểu đồ - không nhấp nháy, hoạt động nhanh hơn và có thể bật/tắt trên toàn cầu

• theo dõi nhiều thị trường nước ngoài hơn: giờ đây bạn có thể xác định tỷ giá cho tối đa 20 loại tiền tệ (khác với loại tiền tệ cơ sở) để kiểm tra ngược nhiều loại tiền tệ trong cửa sổ tùy chọn.

• chế độ backtester mới: backtestRegularRaw2 và backtestRegularRaw2Multi

• phương pháp FindSignal mới của đối tượng backtester

• Hoạt ảnh biểu đồ tối ưu hóa 3D hiện mượt mà hơn (100fps)

• lồng nhau không giới hạn các câu lệnh #include và #include_once

• Cải thiện tỷ lệ biểu đồ bán logarit, thứ tự tab MDI được lưu trong bố cục, báo cáo lỗi được cải thiện, thêm biểu tượng Vista có độ phân giải cao, các bản sửa lỗi và cải tiến khác

Điểm nổi bật của phiên bản 5.00

• Hệ thống danh sách theo dõi mới có các tính năng:
- số lượng danh sách theo dõi không giới hạn
- danh sách giữ nguyên thứ tự ban đầu mà các ký hiệu được thêm vào (vẫn có thể sắp xếp theo thứ tự bảng chữ cái theo yêu cầu)
- chức năng AFL mới để tham chiếu đến danh sách theo dõi theo tên

• Hỗ trợ AFL Code Wizard - chương trình tạo công thức tự động hoàn toàn mới dành cho những người không có bất kỳ kinh nghiệm lập trình nào. Để biết thêm thông tin về trình hướng dẫn AFL Code, hãy xem video giới thiệu này:
http://www.amibroker.com/video/amiwiz/AFLWiz1.html
• Cải tiến công cụ AFL
- các câu lệnh điều khiển luồng mới: switch /case / break / continue
- các toán tử gán hợp chất mới: +=, -=, *=, /=, %=, &=, |=
- các hàm mới: GetPlaybackDateTime(), PopupWindow(), Mersene Twister Random Number Generator mtRandom() và các hàm khác

• Bộ phân bổ heap bộ nhớ chuyên dụng mới cho các báo giá và tín hiệu hệ thống giao dịch, giúp chạy tối ưu hóa lâu hơn bao giờ hết mà không nhận được thông báo hết bộ nhớ
• Hai chế độ kiểm thử ngược mới (có sẵn khi sử dụng hàm SetBacktestMode) cho phép xử lý các tín hiệu nhập chưa lọc (thô)
• Lịch trình hoa hồng 5 tầng do người dùng xác định trong kiểm thử ngược (Phân tích / Cài đặt tự động)
• Chia sẻ mẫu biểu đồ
bây giờ bạn có thể lưu biểu đồ dưới dạng "Biểu đồ Mẫu, Hoàn chỉnh (*.chart)" lưu trữ tất cả các bố cục VÀ công thức được tham chiếu trong MỘT tệp có thể gửi cho bạn bè của bạn và toàn bộ biểu đồ sẽ được khôi phục trên bất kỳ máy tính nào một cách dễ dàng, mà không cần phải sao chép từng công thức.
• Biểu đồ New-Look - các đường phân chia giữa các khung giờ là một pixel và không có đường viền xung quanh biểu đồ
giúp hiển thị biểu đồ và bản in sạch hơn, lớn hơn và dễ đọc hơn
• Thanh phạm vi tùy chỉnh (được hỗ trợ trong biểu đồ và thông qua TimeFrameSet())
• Giao diện đồ họa cấp thấp mới (23 hàm AFL mới)
• Nhập HTML trong Phân tích tự động
• Chống răng cưa toàn màn hình trong trình xem biểu đồ tối ưu hóa 3D (biểu đồ 3D mượt mà tuyệt đẹp và khả năng đọc được cải thiện)
• Hiển thị cửa sổ Báo giá theo thời gian thực được cải thiện (cập nhật nhanh hơn, đánh dấu thay đổi hai màu)
• Kiểm soát Chuyển thời gian trong trình nhập ASCII

Nhật ký thay đổi chi tiết

THAY ĐỔI CHO PHIÊN BẢN 6.10.0 (so với 6.09.0)

• SetOption các trường mới cho Monte Carlo
MCUseEquityChanges - sử dụng thay đổi vốn chủ sở hữu thay vì danh sách giao dịch
MCChartEquityScale - 1 cho thang logarit, 0 cho thang tuyến tính
MCLogScaleFinalEquity - 1 cho thang logarit, 0 cho thang tuyến tính
MCLogScaleDrawdown - 1 cho thang logarit, 0 cho thang tuyến tính
MCNegativeDrawdown - 1 - sử dụng số âm cho drawdown (CDF drawdown ngược)
• Bộ nhớ đệm QuickData không được xóa khi nhấp đúp vào danh sách Kết quả phân tích để hiển thị biểu đồ. Đã sửa.
• MonteCarlo: thêm tùy chọn sử dụng số âm cho drawdown (bật theo mặc định). Điều này đảo ngược thứ tự của cột "drawdown" trong bảng MC và đảo ngược ý nghĩa (tức là giá trị phần trăm 10% có nghĩa là có 10% khả năng drawdown bằng hoặc tệ hơn (tiêu cực hơn) so với giá trị được trình bày). Khi tùy chọn này bị tắt (như trong các phiên bản cũ), drawdown được báo cáo là các số lớn hơn 0 và giá trị phần trăm 10% có nghĩa là có 10% khả năng drawdown bằng hoặc tốt hơn (nhỏ hơn) so với số tiền được trình bày.
• AFL: Gfx: đồ họa cấp thấp được ghi trên lớp 128 đã được phát trước thời hạn. Đã sửa.
• Thêm thông báo lỗi khi người dùng cố gắng ghi giá trị vào BarCount (biểu tượng chỉ đọc)
• Cập nhật thêm tài liệu

THAY ĐỔI CHO PHIÊN BẢN 6.09.0 (so với 6.08.0)

• AFL: chức năng mới fgetcwd - lấy thư mục làm việc hiện tại
• Trình chỉnh sửa AFL: hộp thông báo lỗi hiển thị khi không tìm thấy tệp @link khiến khung chính lấy tiêu điểm. Đã sửa.

• Trình chỉnh sửa AFL: tự động hoàn thành và các chú giải công cụ tham số bật lên khi thêm/chỉnh sửa phần chú thích Doxygen/JavaDoc. Đã sửa.
• Tài liệu cập nhật

THAY ĐỔI CHO PHIÊN BẢN 6.08.0 (so với 6.07.0)

• AFL Editor: Đã triển khai các liên kết theo kiểu JavaDoc/Doxygen có thể nhấp trong các chú thích tài liệu
Để sử dụng các liên kết trong các chú thích, bạn phải đặt lệnh @link theo sau là đường dẫn đến tệp hoặc URL bên trong chú thích theo kiểu JavaDoc/Doxygen:
a) chú thích nhiều dòng bắt đầu bằng
/**
@link readme.html
@link http://www.amibroker.com
*/
(lưu ý dấu hoa thị kép sau dấu gạch chéo đầu tiên)
b) chú thích một dòng bắt đầu bằng dấu gạch chéo ba
/// @link readme.html
/// @link c:\program files\amibroker\readme.html
/// @link http://www.amibroker.com

Bây giờ khi bạn di chuột qua lệnh @link, bạn sẽ thấy dấu gạch chân cho biết lệnh đó có thể nhấp được.
Lệnh này phản ứng với NHẤP ĐÔI (không phải nhấp một lần). Khi bạn nhấp đúp vào tài liệu được liên kết sẽ được mở

Lệnh @link có thể mở các trang web, tệp cục bộ (hỗ trợ cả đường dẫn tương đối và tuyệt đối) bằng chương trình đã đăng ký với Windows
để mở loại tệp đã cho. Vì vậy, nếu bạn sử dụng
/// @link something.doc

thì MS Word sẽ được sử dụng.

/// @link test.xls

sẽ mở test.xls trong Excel.

Đường dẫn tương đối tham chiếu đến thư mục làm việc của AmiBroker.
Các tệp Html được mở bằng trình duyệt mặc định, các tệp txt thường được mở bằng Notepad (hoặc bất kỳ ứng dụng nào bạn sử dụng).

Nếu tệp không tồn tại thì bạn sẽ nhận được thông báo lỗi.
• AFL: Toán tử trừ một ngôi và toán tử NOT có thể gây ra sự cố khi áp dụng cho kết quả của hàm hoặc biến không xác định có kiểu không phải số. Đã sửa.
• Stack walker 64 bit không sử dụng máy chủ biểu tượng Microsoft nữa nên có thể khởi tạo nhanh hơn nhiều. Ngoài ra, việc khởi tạo stack walker được chuyển sang trình tự khởi động vì thực hiện sau khi ngoại lệ là rủi ro và không đáng tin cậy
• 64 bit: vi phạm quyền truy cập trong quá trình chạy Phân tích không dẫn đến tình trạng đóng băng 'ứng dụng không phản hồi'. Thay vào đó, báo cáo lỗi được hiển thị với ngăn xếp cuộc gọi thích hợp.
• Sau khi thay đổi trong hộp thoại ngoại lệ 6.02, lỗi đầu tiên được hiển thị trong công thức ngay cả khi ngoại lệ thực sự không phải do lỗi đầu tiên mà là lỗi cuối cùng. Đã sửa.
• Thông tin ngoại lệ có thể bị nhầm lẫn khi nhiều luồng tạo ra chúng cùng một lúc. Đã sửa.

• Phân tích mới: Chức năng tự động thay đổi kích thước cột đã hiển thị lại các cột ẩn. Đã sửa.

• Giao diện người dùng: AmiBroker hiện ngăn chặn việc chồng chéo hộp thoại "Khôi phục lỗi" khi nhiều ngoại lệ được ném ra từ nhiều luồng. Hộp thoại đầu tiên được hiển thị, các ngoại lệ khác được ghi vào DebugView miễn là "Khôi phục lỗi" không bị loại bỏ.
• "Đổi tên" và các chức năng sao lưu ở một số nơi có thể không thành công nếu tên tệp mới đã tồn tại. Đã sửa.
• Không còn hỗ trợ cho Windows 95

THAY ĐỔI CHO PHIÊN BẢN 6.07.0 (so với 6.06.0)

• AFL: nén cho mảng Biến tĩnh được triển khai để tiết kiệm bộ nhớ và kích thước tệp:
StaticVarSet có tham số bổ sung kiểm soát nén
Tham số compressMode quyết định khi nào biến đã cho sẽ được nén hay không. Theo mặc định, chỉ các biến tĩnh liên tục mới được nén (cmDefault).
Bạn có thể tắt hoàn toàn compressionMode = cmNever hoặc bật nó cho các biến liên tục và không liên tục bằng cách sử dụng
compressionMode = cmAlways

Nén được thực hiện bằng cách xóa các giá trị lặp lại khỏi chuỗi khi các giá trị lặp lại được khôi phục khi thực hiện StaticVarGet.
Nén KHÔNG tương thích với chế độ không căn chỉnh của StaticVarGet.
Nếu mảng nén được StaticVarGet truy xuất với align=False, thì các giá trị lặp lại được tìm thấy trong mảng gốc sẽ không được truy xuất.

Bật nén sẽ làm chậm StaticVarSet (vì nó cần thực hiện một số xử lý bổ sung), nhưng không ảnh hưởng đến hiệu suất của các hàm khác,
do đó StaticVarGet nhanh như nhau khi có hoặc không có nén.
• AFL: printf/StrFormat kiểm tra các chuỗi %s, %c, %d, %x, %i, %u, %x, %p hiện không được hỗ trợ và in Lỗi 62 khi tìm thấy chúng.
• AFL: printf/StrFormat hiện triển khai kiểm tra chuỗi định dạng đúng vì đôi khi người dùng truyền chuỗi bằng % là dấu hiệu đặc biệt để định dạng chuỗi thay vì %% để in dấu phần trăm thực tế

Khi kiểm tra không thành công, "Lỗi 61. Số lượng chỉ định định dạng % không khớp với số lượng đối số được truyền." sẽ hiển thị
• AFL: Một số hàm chu kỳ biến đổi như Ref/Sum/MA đã chấp nhận sai ma trận là 'chu kỳ'. Đã sửa (thông báo lỗi thích hợp hiện được hiển thị).

• AFL: Các hàm StaticVarGet/StaticVarSet hiện hỗ trợ ma trận bao gồm tính bền bỉ (do đó ma trận có thể được ghi vào đĩa và tải lại giữa các lần chạy AmiBroker)
• AFL: VarSet/VarGet chính thức chấp nhận ma trận (cho phép các biến động của loại ma trận). (FWIW chúng đã được chấp nhận trong các phiên bản trước nhưng việc triển khai chưa hoàn thiện.)
• Biểu đồ: Độ dày bấc nến hiện có thể được người dùng xác định là % chiều rộng nến với giá trị tối đa. chiều rộng pixel (Công cụ->Tùy chọn, tab "Thanh và Nến"). Ngoài ra, theo mặc định, độ dày bấc hiện là 25%/5px tối đa thay vì hằng số 1 pixel.
• Biểu đồ: Công cụ chu kỳ trong biểu đồ hàng tuần tạo ra các đường không chính xác khi vượt quá 2038. Đã sửa.
• Biểu đồ: Mức công cụ đường ngang hiện được hiển thị với số thập phân như được xác định trong Tùy chọn (tab Khác, "chữ thập phân trong tiêu đề/công cụ biểu đồ"). Lưu ý rằng cài đặt này được sử dụng khi bạn vẽ một đường mới hoặc sửa đổi đường cũ.
• Trình gỡ lỗi: Đã thêm phím F5 làm phím tắt cho "Gỡ lỗi / Đi"

THAY ĐỔI CHO PHIÊN BẢN 6.06.0 (so với 6.05.0)

• AFL Editor: Đã triển khai dấu trang (Ctrl+F2 - chuyển đổi dấu trang, F2 - chuyển đến dấu trang tiếp theo, Shift+F2 - chuyển đến dấu trang trước đó
• AFL Editor: Tìm trong Tệp - khi tên tệp chứa dấu ngoặc nhọn (), nhấp đúp vào tệp sẽ không mở tệp. Đã sửa.
• AFL Editor: Bình luận dòng đặt chú thích // kép vào các dòng xuất hiện sau dòng trống. Đã sửa. Bây giờ các dòng trống được giữ nguyên.
• AFL: tham số bổ sung cho chế độ hàm DateTimeToStr (chế độ = 0 - chuyển đổi cả phần ngày và giờ, 1 - chỉ ngày, 2 - chỉ giờ) Lưu ý rằng chế độ 2 sẽ cung cấp cho bạn chuỗi rỗng khi áp dụng trên biểu đồ có khoảng thời gian hàng ngày hoặc dài hơn
• AFL: hàm mới GfxFillSolidRect(x1, y1, x2, y2, color) - đây là phương pháp nhanh nhất để tô hình chữ nhật đặc bằng một màu (nhanh hơn GfxRectangle)
pw = Status("pxwidth"); ph = Status("pxheight");
GfxFillSolidRect(0, 0, pw, ph, colorBlack);
• AFL: hàm mới SumSince - tổng nhanh các phần tử mảng vì điều kiện là đúng, hoạt động giống như Cum(array) - ValueWhen(condition, Cum(array)) hoặc Sum(array, BarsSince(condition), nhưng nhanh hơn nhiều Cú pháp: SumSince(condition, array)
• AFL: Hiệu suất Sum() của chu kỳ thay đổi được cải thiện đáng kể khi chu kỳ thay đổi +/-1 trên cơ sở từng thanh
• Hộp thoại danh mục: thêm khả năng sắp xếp lại thứ tự của thị trường/nhóm/lĩnh vực/ngành công nghiệp và danh sách theo dõi bằng cách sử dụng các nút "Di chuyển lên" / "Di chuyển xuống"
Đây là nhiệm vụ không hề đơn giản vì tất cả các ký hiệu phải được đồng bộ hóa với sự thay đổi vì khi thứ tự của các danh mục thay đổi thì tất cả dữ liệu ký hiệu phải được lập chỉ mục lại để phản ánh thứ tự mới vì các ký hiệu tham chiếu đến vị trí ban đầu của category.
• Phân tích mới: Thêm tùy chọn "Tự động thay đổi kích thước cột để vừa với nội dung" - TẮT theo mặc định. Khi bật,
chiều rộng cột được điều chỉnh để vừa với nội dung sau khi phân tích hoàn tất. Lưu ý rằng điều này có nghĩa là bất kỳ chiều rộng nào được chỉ định trong tham số width trong AddColumn width sẽ bị bỏ qua khi tùy chọn này được BẬT.
• Phân tích mới: khi "Chờ để điền lại" được BẬT và một số ký hiệu trong danh sách "Áp dụng cho" bị sai (không tồn tại) thì Phân tích đã cố gắng chạy mã cho ký hiệu không có dấu ngoặc kép gây ra sự cố. Đã sửa.

THAY ĐỔI CHO PHIÊN BẢN 6.05.0 (so với 6.04.0)

• Trình soạn thảo AFL Tìm trong Tệp đã được triển khai - trên danh sách đầu ra, bạn có thể nhấp đúp vào dòng để mở tệp trong trình soạn thảo
• Trình soạn thảo/Gỡ lỗi AFL: Kiểm tra mã và Hồ sơ ngừng hoạt động khi chạy sau phiên gỡ lỗi. Đã sửa.
• AFL: Null hiện được MxFromString chấp nhận
• AFL: hiệu suất của printf() cho đầu ra khối lượng lớn (>60000 ký tự) được cải thiện ~60 lần.
• Biểu đồ: giảm hiện tượng rách khi thay đổi kích thước cửa sổ biểu đồ
• Trình gỡ lỗi hiện có thể sử dụng khoảng thời gian cơ sở hoặc khoảng thời gian biểu đồ hiện tại (Công cụ->Tùy chọn, tab Trình gỡ lỗi, cài đặt "Khoảng thời gian thanh")
• Trình gỡ lỗi: triển khai Cửa sổ đầu ra (cho đầu ra printf() trong quá trình gỡ lỗi)
• Đã xóa trình chỉnh sửa AFL cũ khỏi mã, xóa hộp kiểm "sử dụng trình chỉnh sửa mới" khỏi tùy chọn. Trình chỉnh sửa mới hiện luôn được sử dụng.
• Giao diện người dùng: Các hình tam giác mở rộng trong chế độ xem cây và hộp kiểm trong danh sách Lớp khó nhấp trên màn hình HighDPI vì các biểu tượng quá nhỏ. Đã sửa. Bây giờ các biểu tượng lớn hơn (32x32) được sử dụng thay vì biểu tượng nhỏ (16x16) trên

các máy tính bảng nhỏ như Windows 8" có độ phân giải phù hợp (chẳng hạn như 1920x1200) có DPI rất cao (khoảng 200DPI).
• Cửa sổ theo dõi: bây giờ các biểu thức có chỉ số dưới 2D để truy cập các phần tử của ma trận matrix[x][y] được hỗ trợ

THAY ĐỔI CHO PHIÊN BẢN 6.04.0 (so với 6.03.0)

• Trình chỉnh sửa AFL & Cửa sổ theo dõi: nội dung của ma trận hiện được hiển thị trong chú giải công cụ giá trị và cửa sổ theo dõi
• Trạng thái trình gỡ lỗi (biến/biểu thức theo dõi và vị trí điểm dừng) hiện được lưu giữa các phiên trình chỉnh sửa AFL (theo công thức để mỗi công thức có "trạng thái gỡ lỗi" riêng). Dữ liệu được lưu trong tệp XML có phần mở rộng .dbg
• Trình gỡ lỗi: Chế độ MDI đã được triển khai. Trong 6.0.3, AFL Editor không hỗ trợ 'chế độ MDI' (thậm chí bị sập), giờ chức năng này đã trở lại. Cửa sổ Watch cũng khả dụng ở chế độ MDI và trình gỡ lỗi cũng hoạt động ở chế độ MDI.
• Tùy chọn: thêm trang mới với các thiết lập của Trình gỡ lỗi. Các thiết lập như sau:
+ Giới hạn BarCount thành - xác định số lượng thanh tối đa trong mảng (BarCount) được sử dụng trong quá trình gỡ lỗi
+ Tự động cuộn đến mục thay đổi đầu tiên - khi tùy chọn này BẬT, danh sách "Mảng" trong cửa sổ Watch sẽ tự động cuộn đến mục mảng đầu tiên đã thay đổi (do đó bạn không cần phải định vị các phần tử đã thay đổi theo cách thủ công)
+ Giữ trạng thái gỡ lỗi - khi tùy chọn này BẬT, trình soạn thảo AFL sẽ lưu trạng thái gỡ lỗi (điểm dừng và đồng hồ) trong tệp .dbg cùng với công thức khi đóng trình soạn thảo và khôi phục trạng thái khi công thức được mở lại.
• Tùy chọn: tab 'Biên tập viên' được di chuyển để các tab 'AFL', 'Biên tập viên' và 'Trình gỡ lỗi' nằm cạnh nhau, cũng như các tab
'Cảnh báo' và 'Tiền tệ' được di chuyển trước 'Khác'
• Cửa sổ theo dõi: một tab mới "Mảng" hiển thị đầu ra mảng giống như khám phá để kiểm tra chi tiết nội dung mảng (20 mảng đầu tiên từ cửa sổ theo dõi được báo cáo)
• Cửa sổ theo dõi: triển khai tô sáng thay đổi mục mảng và tự động cuộn để mục thay đổi đầu tiên luôn hiển thị trong tab 'Mảng'
• Cửa sổ theo dõi: triển khai trình đánh giá biểu thức, do đó bạn không chỉ có thể hiển thị các biến mà còn có thể hiển thị các biểu thức liên quan đến các biến như (high+low)/2 hoặc các phần tử mảng riêng lẻ như myvariable[i] trong đó i là bộ đếm vòng lặp động
• Cửa sổ theo dõi: triển khai tô sáng thay đổi giá trị: các giá trị đã thay đổi được hiển thị với nền màu vàng nhạt, ngoài ra các giá trị số (vô hướng) được hiển thị màu xanh lá cây khi chúng tăng hoặc màu đỏ nếu chúng giảm
• Cửa sổ theo dõi: nhấn phím DELETE trong khi chỉnh sửa tên biến khiến biến bị xóa khỏi chế độ theo dõi. Đã sửa.
• Cửa sổ theo dõi: giờ đây có thể kéo thả các biến từ cửa sổ trình soạn thảo AFL

THAY ĐỔI CHO PHIÊN BẢN 6.03.0 (so với 6.02.0)

• Trình soạn thảo AFL: Đã triển khai Trình gỡ lỗi Visual AFL hoàn toàn mới tích hợp LƯU Ý: đây là công việc đang tiến hành, nhiều tính năng hơn sẽ được thêm vào trong các bản beta sau
• Trình gỡ lỗi: Đã triển khai các điểm dừng (bao gồm cả các điểm dừng được đặt trước khi biên dịch cũng như thêm/xóa điểm dừng trong quá trình gỡ lỗi)

Để thêm/xóa điểm dừng, hãy sử dụng nút thanh công cụ hình tròn màu đỏ hoặc nhấn F9
Có thể thêm và xóa điểm dừng bất kỳ lúc nào (trong quá trình chỉnh sửa, khi trình gỡ lỗi ở trạng thái 'đang chạy' hoặc dừng tại điểm dừng)
Điểm dừng được triển khai theo từng câu lệnh (dưới dạng từng bước). Lưu ý rằng chỉ có thể có một điểm dừng trong bất kỳ dòng nào, vì vậy nếu dòng có nhiều hơn một câu lệnh

như thế này:
x=1;y=1;
thì điểm dừng sẽ kích hoạt trước câu lệnh đầu tiên trong dòng này.

Điểm dừng hiện đang hoạt động với:
a) các câu lệnh thông thường (kết thúc bằng dấu chấm phẩy). Đối với các câu lệnh nhiều dòng, hãy đặt điểm dừng ở dòng đầu tiên của câu lệnh
b) vòng lặp for
c) vòng lặp while
d) vòng lặp do-while (bạn cần đặt điểm dừng tại vị trí mệnh đề 'while', nó sẽ không dừng ở dòng 'do' vì về cơ bản nó không phải là lệnh, nếu bạn muốn dừng ở đầu
của do, chỉ cần đặt điểm dừng ở câu lệnh đầu tiên bên trong { block }
d) câu lệnh if
e) câu lệnh return
f) câu lệnh switch/case
g) câu lệnh break

Điểm dừng mà bạn đặt trên các dòng khác sẽ không kích hoạt. Trình soạn thảo AFL sẽ không cho phép đặt điểm dừng trên dòng trống hoặc dòng bắt đầu bằng // chú thích hoặc dấu ngoặc nhọn duy nhất
• Trình gỡ lỗi: Đã triển khai bước đơn Step Into, Step Over

Bước đơn được thực hiện theo từng câu lệnh. Vì vậy, nó hoạt động trên từng câu lệnh tại một thời điểm. Ví dụ, các dòng trống bị bỏ qua và các câu lệnh kéo dài nhiều hàng như bên dưới được coi là một bước.

```
x = "test" +
" second row" + " third row";
```

Nhưng nếu bạn đặt hai câu lệnh trên một dòng như thế này:
x = 1; y = 2;
Thì dòng này sẽ được coi là hai bước (mỗi biểu thức x = 1; và y = 2; riêng biệt). Phím tắt: Bước qua F10 Bước vào F11
• Trình gỡ lỗi: Đã triển khai công cụ chú giải kiểm tra giá trị biến (di chuột qua biến để xem giá trị của biến)
• Trình gỡ lỗi: Đã triển khai cửa sổ Watch với giá trị biến tự động cập nhật khi trình gỡ lỗi thực hiện từng bước hoặc đạt đến điểm dừng

Để thêm biến vào cửa sổ watch, chỉ cần nhấp đúp vào danh sách (ở cuối để thêm biến mới). Bạn cũng có thể thay đổi tên biến bằng cách nhấp đúp vào mục hiện có
• Đổi tên PersistVars.temp thành PersistVars.bin không thành công trên một số phiên bản hệ điều hành. Đã sửa.
• Cửa sổ trích dẫn RT: chế độ xem danh sách kéo-thả hiện hiển thị điểm đánh dấu mũi tên nhỏ cho điểm thả.
• Cửa sổ trích dẫn RT: khi di chuyển các mục bằng cách kéo-thả, hình ảnh các mục đã di chuyển bị nhấp nháy. Đã sửa
• Giao diện người dùng: Thu phóng biểu đồ thông qua Ctrl+bánh xe chuột hiện hoạt động để cố gắng lấy vị trí chuột hiện tại làm điểm "trung tâm" khi có thể. (Trước đây, nó luôn hoạt động nên đường viền bên phải là 'trung tâm').
• AFL: Sum(array,N) đưa ra các giá trị NULL cho các chỉ số bắt đầu từ 0 đến chỉ số N-1 thay vì N
• Khi dữ liệu trong ngày được sử dụng, dấu thời gian 00:00:00 không được hiển thị trong chú giải công cụ dữ liệu (hiển thị trường trống). Đã sửa.

THAY ĐỔI CHO PHIÊN BẢN 6.02.0 (so với 6.01.0)

• Trình chỉnh sửa AFL: Đã triển khai tính năng chú thích/bỏ chú thích Dòng (Chỉnh sửa->Chú thích Dòng, Ctrl+Q) để tự động thêm/xóa // chú thích dòng khỏi một hoặc nhiều dòng. Nếu nhiều dòng được chọn, nội dung của dòng đầu tiên được chọn sẽ quyết định khối dòng nào sẽ được chú thích hay không được chú thích
• AFL: Nếu bất kỳ ô ma trận nào có giá trị Null thì toán tử tích ma trận @ sẽ tạo ra Null tương ứng
hàng/cột của ma trận kết quả.
• AFL: nếu người dùng gọi (a) GetPerfomanceCounter(1) thì (b) GetPerformanceCounter(0) thì lệnh gọi tiếp theo đến GetPerformanceCounter(0) sẽ trả về thời gian tích lũy không phải từ lệnh gọi (a) mà từ khởi động hệ thống. Đã sửa.
• AFL: giờ đây có thể sử dụng định danh ma trận trong câu lệnh if-else. Điều kiện như vậy sẽ kiểm tra bất cứ khi nào phần tử đầu tiên của ma trận [0][0] KHÔNG PHẢI NULL. Điều này hữu ích để kiểm tra đầu ra của các hàm như MxInverse/MxSolve sẽ trả về NULL trong tất cả các ô nếu ma trận là số ít
Vậy

```
m = Matrix( 10, 10, 0 );

// làm gì đó với matrix
if( m )
{
// một số mã
}
```

tương đương với

```
if( NOT IsNull( m[ 0 ][ 0 ] ) )
{
// một số mã
}
```

• AFL: MxSolve/MxInverse hiện trả về một ma trận chứa Nulls nếu ma trận nguồn không thể đảo ngược và tạo ra cảnh báo cấp độ 2 thay vì lỗi.
• AFL: Trên Windows Vista và các biến tĩnh cao hơn, sử dụng khóa đọc-ghi mỏng (SRW) thay vì phần quan trọng. Điều này giúp tăng hiệu suất 5% trong các tình huống đa luồng.
• AFL: StaticVarAdd("name", value, keepAll = True, persistent = False) - một hoạt động bổ sung nguyên tử (đọc-thêm-ghi liên kết) cho các biến tĩnh

Đây là hoạt động bổ sung an toàn cho đa luồng đối với các biến tĩnh được nhiều luồng chia sẻ. Hàm này là nguyên tử đối với các lệnh gọi đến các hàm biến tĩnh khác.

Cờ KeepAll khi được đặt thành true sẽ mô phỏng hành vi của AddToComposite. Nó giữ lại tất cả các giá trị đã có, do đó nếu có lỗ hổng dữ liệu trong ký hiệu hiện tại,
các thanh có trong biến tĩnh nhưng không có trong ký hiệu hiện tại vẫn không bị ảnh hưởng. Khi KeepAll được đặt thành false thì chỉ các thanh có trong ký hiệu hiện tại mới được giữ lại. Bất kỳ thanh nào khác có trong biến tĩnh nhưng không có trong các ký hiệu hiện đang xử lý đều bị xóa. Đó là điều thường xảy ra với StaticVarSet().

Trên thực tế, khi KeepAll được đặt thành False, StaticVarAdd có thể được xem như mã giả sau:

```
EnterCriticalSection
x = Nz( StaticVarGet( "name" ) ); // đọc giá trị hiện có (và chuyển Null thành số không)
```

x += Nz(value); // thêm giá trị vào StaticVarSet("name", x); hiện có // lưu trữ giá trị cập nhật LeaveCriticalSection

Có thể sử dụng hàm này để tạo các hợp chất như sau:

```
if( status("stocknum") == 0 )
{
// xóa bất kỳ giá trị hợp chất nào trước đó StaticVarRemove("~Composite");
}

StaticVarAdd( "~Composite", MACD() > Signal() );

Buy = 0;
```

LƯU Ý:

1. StaticVarAdd tự động chuyển đổi tất cả Null thành số không (như AddToComposite thực hiện).
2. Nếu bạn muốn thay thế AddToComposite bằng StaticVarAdd, hãy nhớ rằng theo mặc định AddToComposite bỏ qua các ký hiệu trong nhóm 253. Điều này được thực hiện để các ký hiệu hợp chất
không được thêm vào chính chúng. Nếu bạn có các ký hiệu hợp chất trong cơ sở dữ liệu của mình và muốn bỏ qua các ký hiệu trong nhóm 253, bạn có thể sử dụng
if(GroupID() != 253) StaticVarAdd("~Composite", values);

3. Nhờ vào việc điều chỉnh mã mở rộng, StaticVarAdd thường cung cấp hiệu suất tốt hơn AddToComposite vốn đã cực nhanh. StaticVarAdd luồng đơn có thể nhanh gấp đôi ATC. Với 8 luồng chạy StaticVarAdd có thể nhanh gấp 4 lần (nó không mở rộng nhiều như người ngây thơ có thể nghĩ, vì phần quan trọng giới hạn hiệu suất do tranh chấp khóa). Để minh họa cho lượng điều chỉnh tinh chỉnh được áp dụng, có thể nói rằng phiên bản 'trực tiếp' đầu tiên của StaticVarAdd thực sự chậm hơn ATC 20 lần.
4. Hãy cẩn thận khi sử dụng "quickafl" vì StaticVarAdd sẽ không tăng 'thanh bắt buộc' (như ATC), vì vậy nếu bạn thực sự muốn thêm tất cả các thanh và afl nhanh được bật trong phân tích, tốt hơn là nên thêm SetBarsRequired(sbrAll, sbrAll)
• AFL: Study() trả về mảng NULL khi ngày bắt đầu của dòng lớn hơn ngày kết thúc. Đã sửa (bây giờ nó cũng hoạt động với các dòng được vẽ từ phải sang trái).

• Tài liệu: Ví dụ về công thức phù hợp đa thức cho thấy cách xử lý ma trận kỳ dị và tràn/không phải số a một cách khéo léo trong phép tính đa thức

```
order = Param( "n-th Order", 10, 1, 16, 1 );
length = 60;

lvb = BarCount - 1; fvb = lvb - length;

yy = Matrix( length + 1, 1, 0 );
xx = Matrix( length + 1, order + 1, 1 );

yy = MxSetBlock( yy, 0, length, 0, 0, Ref( C, fvb ) ); x = BarIndex() - length/2;
for( j = 1; j <= order; j++ )
{
xx = MxSetBlock( xx, 0, length, j, j, x ^ j );
}
```

```
xxt = MxTranspose( xx );
aa = MxSolve( xxt @ xx, xxt ) @ yy;
//aa = MxInverse( xxt @ xx ) @ xxt @ yy; // cách thay thế

if( aa ) // kiểm tra xem ma trận có phải là null không (do đó, nghiệm tồn tại)
{
rr = Null; // lưu trữ phép khớp trong rr
for( i = fvb; i <= lvb; i++ )
{
rr[i] = aa[0][0];

for( j = 1; j <= order; j++ )
{
rr[i] += aa[j][0] * x[ i - fvb ] ^ j;
}
}

if( IsNan( rr[ fvb ] ) )
{
// đa thức của chúng ta tạo ra kết quả vô hạn hoặc không phải là số do tràn/tràn
Title = "Polyfit không thành công. Thứ tự của đa thức quá cao";

}
else
{
SetChartOptions( 0, chartShowDates );
SetBarFillColor( IIf( C > O, ColorRGB( 0, 75, 0 ), IIf( C <= O, ColorRGB( 75, 0, 0 ),
colorLightGrey ) ) );
Plot( rr, "rr", colorWhite, styleLine | styleThick);
}
}
else
{
Title = "Ma trận là kỳ dị. Bậc của đa thức quá cao";
}

Plot( C, "", IIf( C > O, ColorRGB( 0, 255, 0 ), IIf( C <= O, ColorRGB( 255,
0, 0 ), colorLightGrey ) ), styleDots | styleNoLine );
```

• Các biến tĩnh liên tục hiện được lưu vào PersistVars.temp và sau khi ghi thành công, tệp sẽ được đổi tên thành PersistVars.bin. Điều này nhằm ngăn ngừa mất dữ liệu khi ghi các tập hợp biến liên tục rất lớn.
• Khi phát hiện nhiều hơn một lỗi trong một dòng của công thức thì thông báo lỗi đầu tiên sẽ được hiển thị thay vì thông báo lỗi cuối cùng trong biểu đồ/bình luận/phân tích.

THAY ĐỔI CHO PHIÊN BẢN 6.01.0 (so với 6.00.0)

• AFL: MxDet(mx, phương pháp = 0) - tính toán định thức của ma trận
phương pháp = 0 - tự động (sử dụng phương pháp chậm cho các ma trận lên đến và bao gồm 5x5, nhanh cho các ma trận lớn hơn) phương pháp = 1 - chậm (chậm, chính xác hơn) phương pháp = 2 - nhanh (phân tích LU, kém chính xác hơn)

phương pháp "chậm" sử dụng khai triển Laplace phương pháp "nhanh" sử dụng phân tích LU
phương pháp "chậm" cho các ma trận nhỏ (1x1, 2x2, 3x3, 4x4) thực sự nhanh hơn "nhanh", nhanh như nhau đối với ma trận 5x5 và
chậm hơn phương pháp "nhanh" đối với các ma trận lớn hơn 5x5

Vì lý do này, phương pháp "tự động" chỉ sử dụng phương pháp LU "nhanh" cho các ma trận lớn hơn 5x5

Phân tích LU nhanh nhưng có nhiều lỗi số hơn. Phương pháp "Chậm" chậm hơn nhưng lại tạo ra kết quả đáng tin cậy hơn nhiều.
Ví dụ, Octave/MatLab sử dụng phân tích LU sẽ nói rằng định thức của ma trận kỳ dị như thế này
{ {16, 2, 3, 13}, { 5, 11, 10, 8}, {9, 7, 6, 12}, {4, 14, 15, 1 } }
là -1.4495e-012 do lỗi làm tròn của phương pháp LU.

Nếu bạn muốn tính định thức bằng phương pháp nhanh (phân tích LU), hãy gọi MxDet với tham số nhanh được đặt thành 2.

CAVEAT: Phương pháp Laplace có độ phức tạp là O(N!) và vì lý do này, ngay cả khi bạn sử dụng phương pháp = 1, thì kích thước tối đa cho phương pháp này bị giới hạn ở 10x10. Các ma trận lớn hơn luôn được tính toán bằng phương pháp LU

• AFL: MxFromString() - tạo một ma trận mới từ chuỗi theo kiểu danh sách Mathematica/Wolfram: "{ { 1, 2, 3
}, { 4, 5, 6 } }" hoặc kiểu Matlab/Maple "[[1, 2, 3], [4, 5, 6]]" hoặc kiểu dấu phẩy-dấu chấm phẩy GNU Octave [
1, 2, 3; 4, 5, 6]
• AFL: MxGetBlock(matrix, startrow, endrow, startcol, endcol, asArray = False)

Lấy các mục từ ma trận con hình chữ nhật (khối) và trả về ma trận nhỏ hơn (khi asArray được đặt thành False)
hoặc mảng AFL "bình thường" (khi asArray được đặt thành True). Nếu mảng có số thanh khác nhau, các phần tử không sử dụng sẽ được điền bằng Null.

```
z = Matrix( 2, 20, 0 );
// hàng đầu tiên
z = MxSetBlock( z, 0, 0, 0, 19, Close );
// hàng thứ hai
z = MxSetBlock( z, 1, 1, 0, 19, RSI( 5 ) );
printf("Matrix z\n");

printf( MxToString( z ) );

x = MxGetBlock( z, 0, 1, 0, 19, True );

printf("Các mục hiện nằm trong mảng thông thường (chuỗi dữ liệu):\n" ); for( i = 0; i < 20; i++ )
printf( NumToStr( x[ i ] ) + "\n" );

z = MxGetBlock( z, 0, 1, 0, 1 ); // lấy ma trận con 2x2 trên printf("Ma trận con trên z\n");
printf( MxToString( z ) );
```
• AFL: MxInverse(mx) - tính toán nghịch đảo của ma trận (xem các bình luận về MxSolve để biết thêm thông tin)

• AFL: MxSetBlock(matrix, startrow, endrow, startcol, endcol, values = 0)
Đặt giá trị trong khối ô hình chữ nhật (các hàng trong phạm vi startrow..endrow và các cột trong phạm vi startcol..endcol bao gồm).
Điều này cho phép điền toàn bộ hoặc một phần các hàng, cột và tất cả các loại vùng hình chữ nhật khác trong ma trận bằng dữ liệu do người dùng chỉ định
Số hàng và số cột bắt đầu từ số không.
Nếu tham số values là số vô hướng, tất cả các ô trong khối được chỉ định sẽ được điền bằng giá trị đó.
Nếu tham số values là một mảng, các ô trong khối sẽ được điền từ trái sang phải và từ trên xuống dưới bằng các giá trị liên tiếp lấy từ mảng đó.
Nếu có nhiều ô trong khối hơn giá trị trong mảng, bộ đếm mục mảng sẽ chuyển về 0 và bắt đầu lấy giá trị từ đầu

Lưu ý: hàm tạo ma trận mới như một kết quả (do đó ma trận nguồn không bị ảnh hưởng trừ khi bạn thực hiện việc gán kết quả trở lại biến gốc)

Ví dụ 1:
```
// Tạo ma trận 6x6
// và điền vào bên trong 4x4 (trừ các cạnh có số tăng liên tiếp)

y = Matrix( 6, 6, 0 );

y = MxSetBlock( y, 1, 4, 1, 4, Cum(1)); printf("Matrix y\n");
printf( MxToString( y ) );
```

Ví dụ 2:
```
// Tạo ma trận 2 hàng x 20 cột và điền vào các hàng 0, 1 với 20 giá trị đầu tiên của mảng Close và RSI(5) tương ứng

z = Matrix( 2, 20, 0 );

// hàng đầu tiên
z = MxSetBlock( z, 0, 0, 0, 19, Close );
// hàng thứ hai
z = MxSetBlock( z, 1, 1, 0, 19, RSI( 5 ) );
printf("Matrix z\n"); printf( MxToString( z ) );
```
• AFL: MxSolve(A, B) - giải hệ phương trình tuyến tính A@X = B

A cần là ma trận vuông NxN
B phải có N hàng và ít nhất một cột (vectơ dọc).

Sau đó, gọi
X = MxSolve(A, B) sẽ cung cấp giải pháp giữ vectơ dọc của hệ phương trình A @ X = B

B cũng có thể là một ma trận, với mỗi cột của nó biểu diễn một vectơ B khác nhau. Theo cách này, một lệnh gọi duy nhất đến MxSolve có thể giải một số hệ có cùng ma trận A nhưng các vectơ bên phải khác nhau.
Nếu B là ma trận NxM thì MxSolve cũng sẽ tạo ra kết quả có các ô NxM với mỗi cột biểu diễn một giải pháp duy nhất.
Ví dụ 1:

```
A = MxFromString("[ 1, 1, 1, 1; 0, 2, 5, -1; 2, 5, -1, 1; 2, 2, 2, 1 ]");
B = MxFromString("[ 7; -5; 28; 13 ]" ); // vectơ dọc đơn B
```

```
printf( "Giải A * X = B\n" ); printf("Ma trận A\n");
printf( MxToString( A ) ); printf("\nMa trận B\n"); printf( MxToString( B ) );

X = MxSolve( A, B ); printf("\nGiải pháp X\n"); Ví dụ 2:
A = MxFromString("[ 1, 1, 1, 1; 0, 2, 5, -1; 2, 5, -1, 1; 2, 2, 2, 1 ]");
B = MxFromString("[ 7, 14 ; -5, -10; 28, 56; 13, 26 ]" ); // 2 vectơ dọc bên phải

printf( "Giải A * X = B\n" ); printf("Ma trận A\n");
printf( MxToString( A ) ); printf("\nMa trận B\n"); printf( MxToString( B ) );

X = MxSolve( A, B ); printf("\nGiải pháp X\n");
printf( MxToString( X ) ); // hai giải pháp
```

Ghi chú kỹ thuật (Cao) về độ chính xác số:

Mặc dù cả MxSolve và MxInverse đều sử dụng ma trận giải số học/đảo ngược độ chính xác kép phải tuân theo độ chính xác số của IEEE kép
và ví dụ kết quả bằng không có thể xuất hiện dưới dạng 1,4355e-16 (0,0000000000000001) do độ chính xác kép vẫn bị giới hạn về độ chính xác (16 chữ số).

Kết quả của
X = MxInverse(A) @ B;
mặc dù về mặt toán học giống như giải hệ phương trình, nhưng sẽ cho kết quả hơi khác vì nếu bạn thực hiện phép nghịch đảo, ma trận trả về sẽ được chuyển đổi trở lại
thành độ chính xác đơn và tích ma trận được thực hiện với độ chính xác đơn. Khi bạn sử dụng MxSolve, bạn

sẽ thực hiện tất cả các phép tính bằng độ chính xác 64 bit (kép) và
chỉ có kết quả cuối cùng được chuyển đổi trở lại thành độ chính xác đơn. Ví dụ, mã điều chỉnh đa thức hoạt động tốt hơn với MxSolve so với MxInverse

```
// Kiểm tra điều chỉnh đa thức bình phương nhỏ nhất

order = Param( "n-th Order", 15, 1, 25, 1 );
length = 60;

lvb = BarCount - 1; fvb = lvb - length;

yy = Matrix( length + 1, 1, 0 );
xx = Matrix( length + 1, order + 1, 1 );

yy = MxSetBlock( yy, 0, length, 0, 0, Ref( C, fvb ) ); x = BarIndex() - length/2;
for( j = 1; j <= order; j++ )
{
xx = MxSetBlock( xx, 0, length, j, j, x ^ j );
}

xxt = MxTranspose( xx );
aa = MxSolve( xxt @ xx, xxt ) @ yy;
//aa = MxInverse( xxt @ xx ) @ xxt @ yy; // cách khác rr = Null; // lưu trữ sự phù hợp trong rr
for( i = fvb; i <= lvb; i++ ) { rr[i] = aa[0][0];
```

```
for( j = 1; j <= order; j++ ) { rr[i] += aa[j][0] * x[ i - fvb ] ^ j;
} } SetChartOptions( 0, ChartShowDates );
SetBarFillColor( IIf( C > O, ColorRGB( 0, 75, 0 ), IIf( C <= O, ColorRGB( 75, 0, 0 ),
colorLightGrey ) ) );
Plot( rr, "rr", colorWhite, styleLine | styleThick);
Plot( C, "", IIf( C > O, ColorRGB( 0, 255, 0 ), IIf( C <= O, ColorRGB( 255,
0, 0 ), colorLightGrey ) ), styleDots | styleNoLine );
```

• AFL: MxSort(mx, dim = -1, ascening = True) - sắp xếp ma trận

Sắp xếp tất cả các mục trong ma trận
Khi dim == -1 (mặc định), nó sẽ sắp xếp:
a) một hàng nếu chỉ có một hàng (vector nằm ngang)
b) một cột nếu chỉ có một cột (vector nằm dọc)

c) mỗi cột riêng biệt nếu có nhiều hàng và cột hơn một (do đó chúng ta có ma trận 2D thực tế).

Khi dim == 0, hàm sẽ sắp xếp các mục trong mỗi hàng riêng biệt Khi dim == 1, hàm sẽ sắp xếp các mục trong mỗi cột riêng biệt

```
// ví dụ
m = MxFromString("[ 9, 5, 6; 8, 7, 3 ]"); printf( MxToString( m ) + "\n\n" );
printf("%g, %g\n\n", MxGetSize( m, 0 ), MxGetSize( m, 1 ) ); m2 = MxSort( m, 0 ) ;
printf( MxToString( m2 ) + "\n\n" ); m3 = MxSort( m, 1 ) ;
printf( MxToString( m3 ) + "\n\n" );
```

• AFL: MxSortRows(mx, ascending = True, col1 = 0, col2 = -1, col3 = -1)

Sắp xếp các hàng của ma trận theo thứ tự tăng dần/giảm dần của cột col1. Khi cột col1 có các giá trị bằng nhau, SortRows sẽ sắp xếp theo các cột col2 và col3 liên tiếp (nếu col2 và col3 được chỉ định và >= 0).
Số cột bắt đầu từ số không.

Gợi ý: nếu bạn muốn sắp xếp các cột thay vào đó, bạn có thể Chuyển vị/Sắp xếp các hàng/Chuyển vị ngược lại.

```
m = MxFromString("[ 9, 1, 6; 40, 30, 20; 8, 7, 3; 3, 5, 1 ]");
printf("Ma trận đầu vào\n");

printf( MxToString( m ) + "\n\n" );

printf("Hàng %g, Cột %g\n\n", MxGetSize( m, 0 ), MxGetSize( m, 1 ) ); printf("Sắp xếp từng hàng riêng biệt\n");
m2 = MxSort( m, 0 ) ;

printf( MxToString( m2 ) + "\n\n" );

printf("Sắp xếp từng cột riêng biệt\n"); m3 = MxSort( m, 1 ) ;

printf( MxToString( m3 )+ "\n\n");

printf("Sắp xếp các hàng theo nội dung của cột đầu tiên\n"); m4 = MxSortRows( m, True, 0 )
;

printf(MxToString( m4 )+ "\n\n");
```

printf("Sắp xếp các hàng theo nội dung của cột thứ hai\n"); m5 = MxSortRows(m, True, 1) ;

printf(MxToString(m5)+ "\n\n");
• AFL: MxToString - tạo chuỗi từ biến ma trận theo kiểu danh sách Wolfram như thế này (cho ma trận 3x3): { { x00, x01, x02 }, { x10, x11, x12 }, { x20, x21, x22 } }

NHỮNG THAY ĐỔI CHO CÁC PHIÊN BẢN TRƯỚC ĐƯỢC GHI TRONG TÀI LIỆU GHI CHÚ PHÁT HÀNH MÀ BẠN CÓ THỂ TÌM THẤY TRONG THƯ MỤC CÀI ĐẶT AMIBROKER.

Hướng dẫn

Chương này sẽ hướng dẫn bạn những phần quan trọng nhất của AmiBroker.
Nhiệm vụ cơ bản:

- Các thao tác cơ bản

Chủ đề giao diện người dùng:

- Hướng dẫn lập biểu đồ cho người mới bắt đầu
- Cách sử dụng giao diện biểu đồ kéo và thả
- Chủ đề biểu đồ
- Tùy chỉnh giao diện người dùng
- Làm việc với các trang biểu đồ và bố cục cửa sổ
- Làm việc với các lớp
- Sử dụng Web Research
- Sử dụng Account Manager
- Sử dụng Fundamental data
- Sử dụng cửa sổ New Analysis
- Sử dụng cửa sổ Batch

Cập nhật báo giá:

- Cách lấy báo giá từ nhiều sàn giao dịch khác nhau
- Thiết lập nguồn cấp dữ liệu RT eSignal (chỉ dành cho phiên bản RT)
- Thiết lập nguồn cấp dữ liệu RT myTrack (chỉ dành cho phiên bản RT)
- Thiết lập Quote Tracker làm nguồn dữ liệu RT
- Thiết lập nguồn cấp dữ liệu RT IQFeed (chỉ dành cho phiên bản RT)
- Cách sử dụng AmiBroker ở chế độ Thời gian thực (chỉ dành cho phiên bản RT)
- Sử dụng AmiBroker với nguồn dữ liệu bên ngoài khác (Quotes Plus, TC2000 / TCNet, Metastock, FastTrack)
- Tự động cập nhật báo giá EOD cho thị trường Hoa Kỳ và Canada từ Yahoo
- Sử dụng chế độ thủ công của trình tải xuống AmiQuote (Yahoo, MSN Money Central, Quote.com Livecharts)
- Sử dụng trình nhập Metastock

Quản lý cơ sở dữ liệu:

- Hiểu các khái niệm cơ sở dữ liệu
- Hiểu các danh mục
- Làm việc với danh sách theo dõi

Các chủ đề về Ngôn ngữ công thức AmiBroker:

- Hiểu cách ngôn ngữ AFL hoạt động
- Tạo các chỉ báo của riêng bạn
- Sử dụng kiểu biểu đồ và màu sắc trong các chỉ báo
- Cách tạo khám phá của riêng bạn
- Cách viết bình luận biểu đồ của riêng bạn
- Sử dụng các nghiên cứu trong các công thức AFL của bạn
- Kiểm tra ngược các ý tưởng giao dịch của bạn
- Kiểm tra ngược danh mục đầu tư
- Đọc báo cáo kiểm tra ngược
- Cách tối ưu hóa hệ thống giao dịch (nâng cao)
- Kiểm tra Walk-Forward (nâng cao)

- Kiểm tra ngược hợp đồng tương lai (nâng cao)
- Kim tự tháp/mở rộng và nhiều loại tiền tệ trong backtester danh mục đầu tư (nâng cao)
- Mô phỏng Monte Carlo của hệ thống giao dịch (nâng cao)
- Sử dụng cảnh báo dựa trên công thức (nâng cao)
- Sử dụng cửa sổ giải thích (nâng cao)
- Hỗ trợ nhiều khung thời gian (nâng cao)
- Sử dụng hiệu quả đa luồng (nâng cao)
- Chức năng xếp hạng (nâng cao)
- Cách sử dụng đoạn mã
- Sử dụng AFL Debugger
- Sử dụng các điều khiển GUI trên biểu đồ (nâng cao)

Thông tin thêm:

- Video hướng dẫn trực tuyến

Các thao tác cơ bản

Thêm ký hiệu mới

Để thêm ký hiệu mới vào cơ sở dữ liệu, bạn có thể sử dụng mục menu Symbol->New hoặc nút thanh công cụ Add symbol.

Sau khi chọn chức năng này, bạn sẽ được nhắc nhập ký hiệu ticker mới. Độ dài ký hiệu ticker tối đa là 48 ký tự. Để nhập đúng chức năng, bạn nên nhập ký hiệu bằng CHỮ IN HOA.

Xóa ký hiệu

Để xóa ký hiệu hiện có khỏi cơ sở dữ liệu, bạn có thể sử dụng mục menu Symbol->Remove hoặc nút thanh công cụ Remove symbol. Sau khi chọn chức năng này, bạn sẽ được yêu cầu xác nhận xóa ký hiệu.

Lưu ý rằng thao tác này không thể hoàn tác!!!

Có thể xóa nhiều ký hiệu cùng lúc bằng Assignment organizer.

Chia tách cổ phiếu

Để thực hiện chia tách cổ phiếu, hãy sử dụng mục menu Symbol->Split hoặc nút thanh công cụ Split.

AmiBroker cung cấp cách dễ dàng để xử lý chia tách cổ phiếu. Chương trình sẽ cố gắng đoán ngày chia tách và tỷ lệ bằng cách phân tích báo giá. Nếu chỉ có một báo giá duy nhất sau khi chia tách, điều này sẽ hiệu quả, nếu không, bạn sẽ được yêu cầu nhập ngày chia tách và tỷ lệ. Lưu ý rằng thao tác này không thể hoàn tác!!!

Từ phiên bản 2.0 trở lên, chức năng chia tách cung cấp nhiều chức năng hơn: bạn có thể sử dụng tỷ lệ kiểu cũ hoặc bạn có thể chỉ định chia tách bằng cách sử dụng biểu thức sau:

x->y

có nghĩa là x cổ phiếu trước khi chia tách sẽ trở thành y sau khi chia tách. Ví dụ 2->3 có nghĩa là 2 cổ phiếu sẽ trở thành 3 sau khi chia tách. Vì vậy, chia tách thông thường thành năm phần sẽ là 1->5.

Như bạn có thể đoán, bây giờ có thể thực hiện chia tách ngược, ví dụ 2->1, có nghĩa là 2 cổ phiếu được ghép lại thành 1 cổ phiếu.

Xóa báo giá

Để xóa báo giá, chỉ cần chọn báo giá bạn muốn xóa bằng cách nhấp vào biểu đồ (một đường thẳng đứng sẽ xuất hiện hiển thị ngày và báo giá đã chọn). Sau đó, chọn tùy chọn menu Chỉnh sửa->Xóa báo giá.

Để xóa báo giá của tất cả các cổ phiếu từ ngày nhất định, bạn nên sử dụng Phiên chỉnh sửa->Xóa. Bạn cũng có thể sử dụng Trình chỉnh sửa báo giá để xóa báo giá.
Thêm/xóa biểu tượng khỏi mục yêu thích

Để thêm biểu tượng vào mục yêu thích, bạn nên chọn hộp yêu thích trong cửa sổ Thông tin. Để xóa biểu tượng khỏi mục yêu thích, chỉ cần bỏ chọn hộp đó. Ngoài ra, bạn có thể nhấp vào cây bằng nút chuột phải

và chọn tùy chọn "Thêm vào mục yêu thích" và "Xóa khỏi mục yêu thích" từ menu ngữ cảnh.

Hợp nhất các trích dẫn của hai ký hiệu

Đôi khi mã chứng khoán cho ký hiệu bị thay đổi, sau đó bạn có thể nhận được hai mã chứng khoán trong cơ sở dữ liệu của mình
- một mã chứng khoán chứa các trích dẫn lịch sử và mã chứng khoán thứ hai chứa các trích dẫn mới nhất (sau khi đổi tên). Để đưa tất cả các trích dẫn vào một mã chứng khoán duy nhất, bạn nên sử dụng tính năng Symbol->Merge. Bạn chỉ cần chọn mã chứng khoán mới (sau khi đổi tên) và chọn Symbol->Merge. Sau đó, từ kết hợp, bạn nên chọn mã chứng khoán gốc ("hợp nhất với") và tùy chọn kiểm tra các trường sau:

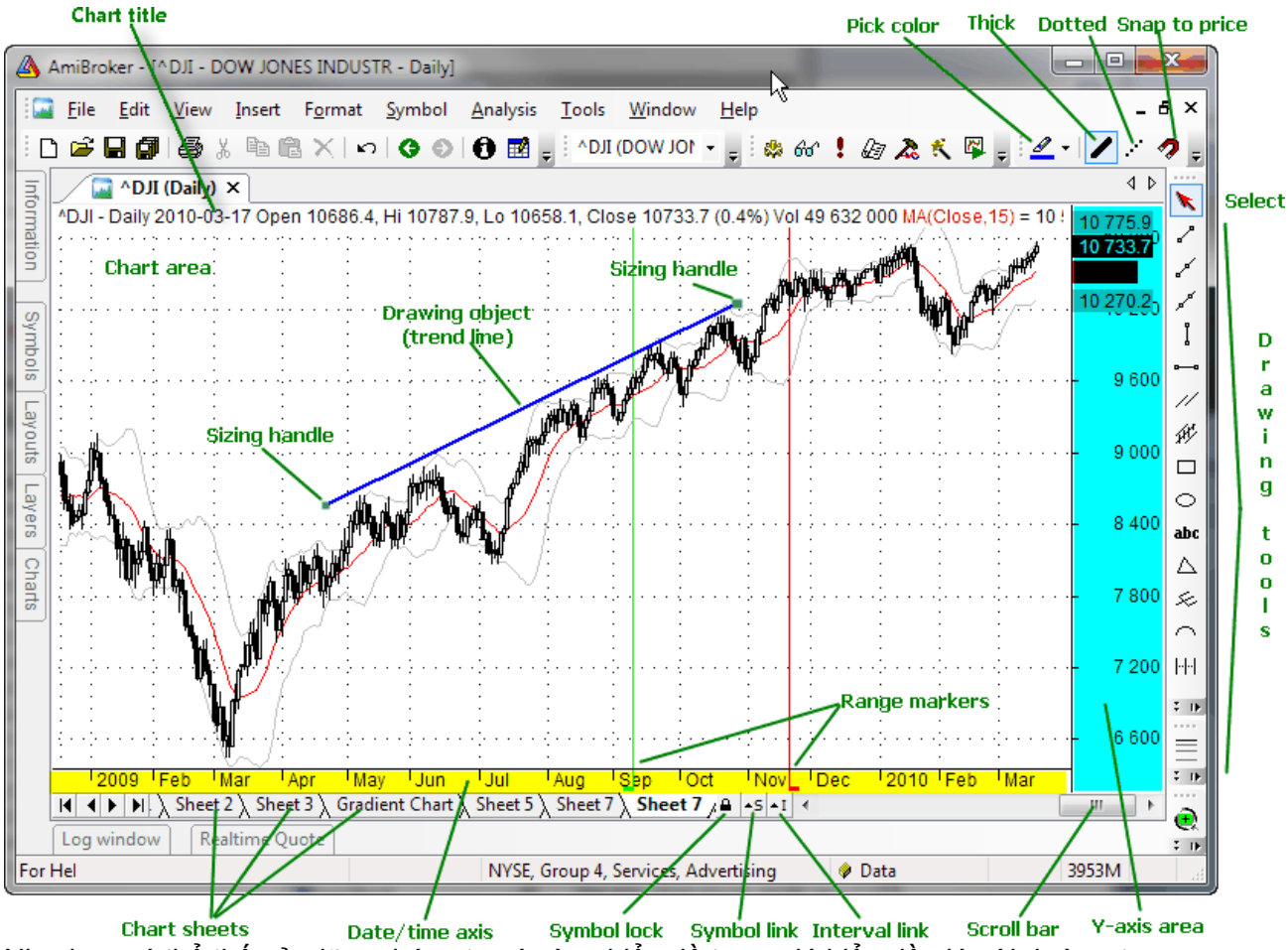
• ghi đè các trích dẫn trùng lặp - việc kiểm tra tùy chọn này sẽ ghi đè các trích dẫn đã có trong mã chứng khoán "mới" bằng các trích dẫn có trong mã chứng khoán "cũ" (thực ra không phải như vậy, nhưng có thể xảy ra).
• xóa "hợp nhất với" sau đó - kiểm tra tùy chọn này sẽ xóa mã chứng khoán "cũ" sau khi hợp nhất
• chỉ định tên bí danh - kiểm tra tùy chọn này sẽ sao chép mã chứng khoán "cũ" vào trường bí danh của mã chứng khoán "mới"

Hướng dẫn lập biểu đồ cho người mới bắt đầu

Giới thiệu

Công cụ lập biểu đồ AmiBroker cho phép thao tác theo hướng đối tượng đối với tất cả các bản vẽ. Bây giờ bạn có thể dễ dàng di chuyển, thay đổi kích thước, cắt, sao chép, dán và xóa tất cả các đối tượng vẽ. Chương này sẽ hướng dẫn bạn qua hầu hết các khía cạnh quan trọng khi sử dụng các công cụ lập biểu đồ.

Bây giờ chúng ta hãy xem giao diện người dùng:



Như bạn có thể thấy ở giữa, chúng ta có vùng biểu đồ trong đó biểu đồ giá với đường trung bình động và dải Bollinger được vẽ (bạn có thể kiểm soát giao diện của các biểu đồ tích hợp từ cửa sổ Công cụ->Tùy chọn).

Ở dưới cùng của biểu đồ, bạn có thể thấy trục ngày (được đánh dấu màu đỏ) và bên dưới là thanh cuộn và điều khiển tab biểu đồ. Thanh cuộn có thể được sử dụng để hiển thị các báo giá trước đó, trong khi tab biểu đồ cho phép xem các trang/biểu đồ khác nhau (nhấp vào đây để tìm hiểu thêm về biểu đồ).

Ở bên phải, bạn có thể thấy vùng trục Y (được đánh dấu màu xanh lam) hiển thị nhãn giá trị và thang đo Y. Nhãn giá trị là các trường màu hiển thị chính xác "giá trị cuối cùng" của các biểu đồ. "Giá trị cuối cùng" là giá trị của chỉ báo (hoặc giá) cho thanh hiện đang hiển thị cuối cùng (ngoài cùng bên phải). Khu vực trục Y cũng được sử dụng để di chuyển/thay đổi kích thước biểu đồ theo chiều dọc.

Bên cạnh bên phải là thanh công cụ đối tượng bản vẽ cho phép bạn chọn từ các loại bản vẽ có sẵn (lưu ý rằng chỉ hiển thị hầu hết các công cụ phổ biến nhất ở đây, bộ công cụ đầy đủ có sẵn từ menu Chèn). Một công cụ đặc biệt có tên là

"Chọn" (mũi tên màu đỏ) được sử dụng để chọn/di chuyển/thay đổi kích thước các đối tượng đã vẽ và để chọn trích dẫn từ biểu đồ.

Ở phần trên, bạn có thể thấy thanh công cụ định dạng cho phép bạn nhanh chóng sửa đổi màu sắc, kiểu (dày/chấm) và chế độ (bắt vào giá) của đối tượng bản vẽ hiện đang được chọn.

Trong hình, bạn cũng có thể thấy đường xu hướng được vẽ với các tay cầm thay đổi kích thước được đánh dấu. Các tay cầm này được sử dụng để kéo/thay đổi kích thước đối tượng như sẽ được giải thích bên dưới.

Các thao tác cơ bản

Cuộn

Để cuộn biểu đồ về phía trước/phía sau, chỉ cần kéo ngón tay cái của thanh cuộn hoặc sử dụng các mũi tên < và > ở bên trái và bên phải của thanh cuộn. Lưu ý rằng sử dụng các mũi tên thanh cuộn < > cho phép bạn di chuyển biểu đồ theo một thanh. Để cuộn biểu đồ, bạn cũng có thể sử dụng chuột được trang bị bánh xe. Chỉ cần lăn bánh xe lên và xuống để cuộn tới lui.

Phóng to

Để phóng to biểu đồ (tăng hoặc giảm số điểm dữ liệu (thanh) được hiển thị), bạn có thể sử dụng
Trình đơn View->Zoom, thanh công cụ zoom hoặc bánh xe chuột.
Bạn cũng có thể phóng to bằng cách kéo cạnh trái hoặc phải của thanh cuộn. Có các tùy chọn sau: zoom-in - giảm số điểm dữ liệu được hiển thị, zoom-out - tăng số điểm dữ liệu được hiển thị, zoom-all - hiển thị tất cả các thanh có sẵn, zoom-normal đặt lại số thanh được hiển thị thành giá trị được xác định trong Tools->Preferences->Charting. Có thể truy cập các tùy chọn phóng to và thu nhỏ trực tiếp từ thanh công cụ View. (xem hình bên dưới). Để phóng to bằng bánh xe chuột, chỉ cần nhấn và giữ phím CTRL và lăn bánh xe. Bạn cũng có thể phóng to đến bất kỳ phạm vi từ-đến nào được chọn trên biểu đồ (xem 'Đánh dấu phạm vi' sau trong hướng dẫn này)

Thu nhỏ, mở rộng và di chuyển thang trục Y

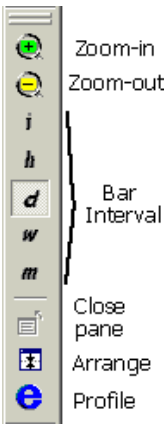
Để di chuyển thang trục Y, hãy di chuột đến vùng trục Y (được đánh dấu bằng màu xanh lam trong hình trên) và bạn sẽ thấy con trỏ thay đổi thành mũi tên lên/xuống. Bây giờ, bạn nhấp và kéo trục Y lên/xuống và thả nút khi trục ở đúng vị trí.

Để thu nhỏ/mở rộng thang trục Y: nhấn phím SHIFT và nhấp vào vùng trục Y, bây giờ hãy thu nhỏ/mở rộng thang trục Y bằng cách di chuyển chuột lên và xuống. Thả nút để hoàn tất.

Để đặt lại thang và vị trí trục Y, chỉ cần nhấp đúp vào vùng trục Y.

Thay đổi khoảng thời gian thanh (chu kỳ)

Bạn có thể dễ dàng chuyển đổi giữa các khoảng thời gian hàng ngày/hàng tuần/hàng tháng và trong ngày bằng cách chọn từ menu Xem và nhấn nút thanh công cụ (xem bên dưới).



Thanh công cụ sử dụng ký hiệu sau cho các khoảng thời gian - i - trong ngày, h - hàng giờ, d - hàng ngày, w - hàng tuần, m - hàng tháng. i biểu thị khoảng thời gian trong ngày "cơ sở" như được định nghĩa trong Tệp->Cài đặt cơ sở dữ liệu. Các khoảng thời gian trong ngày còn lại có sẵn từ menu Xem->Trong ngày.
Cài đặt khoảng thời gian chỉ ảnh hưởng đến cửa sổ đang hoạt động, do đó mỗi cửa sổ có thể có khoảng thời gian khác nhau. Xin lưu ý rằng các khoảng thời gian trong ngày sẽ bị vô hiệu hóa nếu cơ sở dữ liệu của bạn ở chế độ cuối ngày.
Chế độ trong ngày chỉ khả dụng cho các cơ sở dữ liệu có "Khoảng thời gian cơ sở" trong
Tệp->Cài đặt cơ sở dữ liệu được đặt thành bất kỳ giá trị nào nhỏ hơn cuối ngày. Ví dụ, nếu bạn đặt "Khoảng thời gian cơ sở" trong Tệp->Cài đặt cơ sở dữ liệu thành 5 phút, tất cả các chu kỳ biểu đồ từ 5 phút trở lên sẽ được bật.

Các khoảng thời gian sau được tích hợp sẵn:

- hằng ngày
- hằng tuần
- hằng tháng
- hằng giờ (trong ngày)
- 15 phút (trong ngày)
- 5 phút (trong ngày)
- 1 phút (trong ngày)
- 15 giây (chỉ RT trong ngày)
- 5 giây (chỉ RT trong ngày)
- tích tắc (chỉ RT trong ngày)

Ngoài ra, bạn có thể xác định 5 khoảng thời gian thanh n phút tùy chỉnh và 5 khoảng thời gian n tích tắc tùy chỉnh trong
Công cụ->Tùy chọn->Trong ngày. Các khoảng thời gian tùy chỉnh chỉ khả dụng trong menu Xem->Trong ngày.

Chọn báo giá

Bạn có thể dễ dàng xem báo giá và giá trị của các chỉ báo trước đó bằng cách sử dụng chế độ "chọn". Để chọn báo giá trước đó, trước tiên hãy chuyển sang chế độ "Chọn" (mũi tên màu đỏ trên thanh công cụ), sau đó nhấp vào vùng biểu đồ (nhưng không phải trên đối tượng bản vẽ). Một đường thẳng đứng sẽ hiển thị đánh dấu báo giá bên dưới con trỏ. Tiêu đề biểu đồ sẽ hiển thị báo giá thanh này. Các ngăn chỉ báo sẽ hiển thị giá trị chỉ báo cho thanh đã cho. Sau khi báo giá được chọn, bạn có thể di chuyển đến báo giá trước đó/tiếp theo bằng cách sử dụng phím mũi tên trái và phải (con trỏ) <- và -> trên bàn phím.

Để tắt lựa chọn báo giá, hãy nhấp lại vào dòng hoặc nhấp vào vùng trục ngày (được đánh dấu bằng màu đỏ trong hình trên) hoặc nhấp vào vùng lề phải (trích dẫn trống). Khi lựa chọn tắt, tiêu đề biểu đồ sẽ hiển thị các giá trị cho thanh cuối cùng hiển thị.

Phạm vi đánh dấu

Để hiển thị đánh dấu phạm vi, chỉ cần nhấp đúp vào biểu đồ ở đầu phạm vi và nhấp đúp lại vào cuối phạm vi. Bạn cũng có thể sử dụng phím F12 kết hợp với chế độ "chọn" (được mô

tả ở trên). Chỉ cần chọn trích dẫn và nhấn F12 để bắt đầu và SHIFT+F12 để kết thúc phạm vi. Bạn có thể tắt dấu phạm vi bằng cách nhấn phím CTRL+F12 hoặc nhấp đúp vào cùng một vị trí hai lần.

Có thể sử dụng dấu phạm vi để chọn phạm vi phóng to (Xem->Thu phóng->Phạm vi) và thực hiện các phép tính trên các giá trị đã chọn thông qua các hàm BeginValue và EndValue AFL.

Thêm/đóng ngăn biểu đồ

Mỗi cửa sổ có thể bao gồm một số ngăn hiển thị nhiều biểu đồ/chỉ báo khác nhau. Để hiển thị một chỉ báo mới trong một ngăn biểu đồ riêng biệt, chỉ cần tìm chỉ báo trong danh sách Biểu đồ (sử dụng menu Cửa sổ -> Biểu đồ) và nhấp đúp vào tên chỉ báo. Để biết thêm thông tin, hãy xem hướng dẫn về biểu đồ Kéo và thả.

Để đóng bất kỳ ngăn biểu đồ nào: nhấp vào ngăn, sau đó sử dụng View->Pane->Close từ menu chính hoặc nhấp vào ngăn bằng nút chuột phải và chọn Close từ menu ngữ cảnh.

Liên kết và khóa biểu đồ

Nhiều cửa sổ biểu đồ (mở bằng File->New->Default Chart hoặc File->New->Blank chart) có thể được liên kết theo khoảng, liên kết theo ký hiệu bằng các nút "S" và "I" nhỏ thích hợp xuất hiện ở bên trái thanh cuộn. Khi bạn nhấp vào nút, menu hiển thị màu sẽ bật lên, chọn một màu từ 2 biểu đồ trở lên và các ký hiệu được liên kết bằng cùng màu sẽ được liên kết bằng ký hiệu và/hoặc khoảng. Liên kết có nghĩa là việc thay đổi ký hiệu và/hoặc khoảng trong một trong các cửa sổ được liên kết sẽ tự động thay đổi ký hiệu và/hoặc khoảng trong tất cả các cửa sổ được liên kết bằng cùng màu liên kết.

Bạn cũng có thể ngăn không cho ký hiệu bị thay đổi đối với biểu đồ đã cho. Có thể thực hiện bằng cách bật nút khóa nhỏ ("Khóa ký hiệu") ở bên phải thanh cuộn. Khi biểu đồ bị khóa ký hiệu, biểu đồ sẽ không cho phép thay đổi ký hiệu đã chọn trừ khi khóa được nhả (bằng cách nhấn lại nút khóa).



Sử dụng công cụ vẽ

AmiBroker có bộ công cụ vẽ mở rộng:

Các công cụ sau đây khả dụng:

- Đường xu hướng
- Tia (mới trong 4.20)
- Đường mở rộng (mới trong 4.20)
- Đường thẳng đứng
- Đường ngang
- Đường song song (mới trong 4.20)
- Kênh hồi quy: Raff, độ lệch chuẩn, lỗi chuẩn (tất cả đều mới trong 4.20)
- Nghiên cứu Fibonacci Retracement (nâng cao trong 4.20)
- Nghiên cứu Fibonacci Timezone
- Fibonacci Fan
- Fibonacci arc
- Gann Square (mới trong 4.20)
- Gann Fan (mới trong 4.20)
- Công cụ Ellipse
- Công cụ Arc

• Rectangle
• Công cụ hộp văn bản

Có sẵn trong menu Insert và thanh công cụ Draw. Mỗi đối tượng vẽ có thể được di chuyển, thay đổi kích thước, sao chép, xóa và sửa đổi sau khi vẽ.

Để vẽ một đối tượng trên biểu đồ, hãy bật nút công cụ thích hợp (xem hình bên dưới) và bắt đầu vẽ trên biểu đồ bằng cách trỏ chuột và nhấn nút chuột trái vào nơi bạn muốn bắt đầu vẽ. Sau đó, di chuyển chuột. Đường theo dõi nghiên cứu sẽ xuất hiện. Nhả nút chuột trái khi bạn muốn hoàn tất bản vẽ. Bạn cũng có thể hủy bản vẽ nghiên cứu bằng cách nhấn phím ESC (thoát).

Nếu bạn di chuột qua đối tượng, bạn sẽ thấy hình dạng con trỏ thay đổi theo khoảng cách gần đối tượng. Điều này có nghĩa là

Nếu con trỏ ở gần một trong hai đầu của đối tượng, nó sẽ thay đổi hình dạng thành con trỏ định cỡ:



Nếu con trỏ ở gần các phần còn lại của đối tượng, nó sẽ thay đổi hình dạng thành con trỏ di chuyển:



Sau khi vẽ đối tượng, bạn có thể chọn, di chuyển, thay đổi kích thước, xóa, sao chép đối tượng.



Để chọn đối tượng, chỉ cần di chuyển chuột qua đối tượng để "con trỏ di chuyển" xuất hiện và nhấp một lần - đối tượng sẽ được đánh dấu để các tay cầm định cỡ (xem hình đầu tiên) sẽ xuất hiện.

Để bỏ chọn, hãy nhấp vào khoảng trống biểu đồ.

Để thay đổi kích thước đối tượng, hãy nhấp vào tay cầm thay đổi kích thước và kéo đến vị trí mong muốn như trong hình.

Để di chuyển đối tượng, hãy nhấp vào bất kỳ phần nào khác của đối tượng và di chuyển đến vị trí mong muốn.

Để xóa đối tượng - trước tiên hãy chọn đối tượng đó và nhấn phím DEL (DELETE) trên bàn phím hoặc sử dụng menu Edit->Delete hoặc sử dụng nút thanh công cụ Delete.

Để sao chép đối tượng vào bảng tạm - trước tiên hãy chọn đối tượng đó và nhấn Ctrl+C hoặc sử dụng menu Edit->Copy hoặc sử dụng nút thanh công cụ Copy.
Để cắt đối tượng - hãy chọn đối tượng trước và nhấn Ctrl+X hoặc sử dụng menu Edit->Cut hoặc sử dụng nút thanh công cụ Cut.

Để dán đối tượng từ bảng tạm, hãy nhấn Ctrl+V hoặc sử dụng Edit->Paste hoặc sử dụng nút thanh công cụ Paste. Đối tượng được dán sẽ được vẽ ở cùng vị trí chính xác như đối tượng đã sao chép và sẽ được chọn tự động để bạn có thể di chuyển đến vị trí mới.

Để áp dụng màu hoặc kiểu cho đối tượng, hãy chọn đối tượng đó và sử dụng menu Format hoặc các nút thanh công cụ Format để thay đổi màu, độ dày, chấm và kiểu bám theo giá. Lưu ý rằng bạn cũng có thể chọn màu và kiểu của đối tượng trước khi vẽ đối tượng mới: chỉ cần bỏ chọn đối tượng trước đó (nếu có), thay đổi lựa chọn màu/kiểu và vẽ đối tượng mới.

Để sửa đổi thuộc tính của đối tượng - hãy nhấp đúp vào đối tượng hoặc sử dụng menu Edit->Properties hoặc phím Alt+ENTER Để xóa tất cả các đối tượng, hãy sử dụng menu Edit->Delete All

Thông tin thêm

Để tìm hiểu thêm về các công cụ vẽ, vui lòng đọc chương Tài liệu tham khảo về công cụ vẽ.

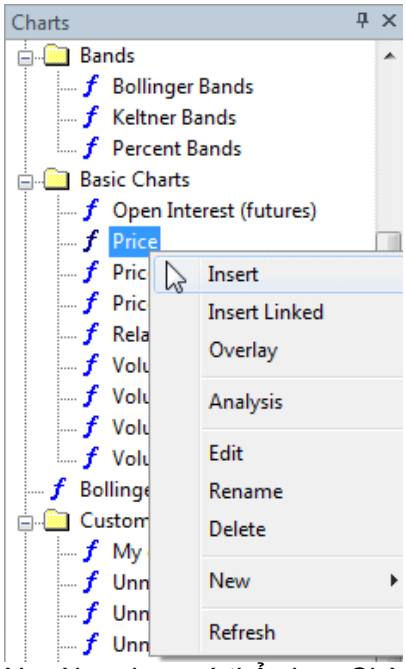
Cách sử dụng giao diện biểu đồ kéo thả

Giới thiệu
AmiBroker cho phép bạn dễ dàng tạo và sửa đổi các chỉ báo của mình chỉ bằng vài thao tác di chuyển chuột. Từ giờ trở đi, bạn có thể xây dựng các chỉ báo phức tạp mà không cần bất kỳ kiến thức lập trình nào. Các chỉ báo khả dụng (sẵn sàng sử dụng) được liệt kê trong tab Biểu đồ của cửa sổ Không gian làm việc.

Có một video hướng dẫn tại: http://www.amibroker.net/video/dragdrop1.html cho thấy cách sử dụng cơ bản chức năng kéo thả mới.

Cách chèn một chỉ báo mới.

Để hiển thị một chỉ báo mới trong một ngăn biểu đồ riêng biệt, chỉ cần tìm chỉ báo trong danh sách Biểu đồ (sử dụng menu Cửa sổ -> Biểu đồ) và nhấp đúp vào tên chỉ báo.



Ngoài ra, bạn có thể chọn Chèn từ menu ngữ cảnh. Kết quả là ngăn chỉ báo mới sẽ được tạo và hộp thoại Tham số sẽ được hiển thị. Tại đây, bạn có thể thay đổi các thuộc tính của chỉ báo (như màu sắc hoặc chu kỳ). Để chấp nhận các cài đặt, hãy nhấn nút OK. (bạn sẽ tìm thấy mô tả chi tiết về cửa sổ tham số bên dưới).

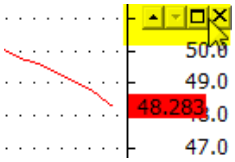
Ví dụ:
Để chèn khung RSI - tìm chỉ báo RSI trong danh sách, nhấp đúp vào tên, chọn số chu kỳ và màu, sau đó nhấn OK.

Cách chồng một chỉ báo lên một chỉ báo khác.

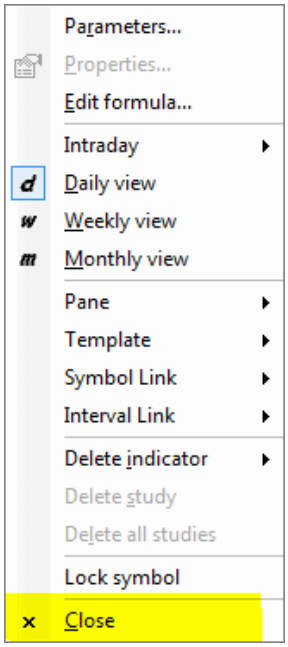
Để chồng một chỉ báo lên một chỉ báo khác, nhấn nút chuột TRÁI vào tên chỉ báo, kéo (giữ nút chuột) chỉ báo đã chọn vào khung đích và nhả nút.
Ví dụ:
Để chèn một RSI khác (dựa trên số chu kỳ khác nhau) vào cùng một khung - kéo RSI vào khung RSI đã tạo trước đó, thay đổi số chu kỳ trong cửa sổ Tham số và nhấn OK
Ngoài ra, bạn có thể chọn tùy chọn Chồng từ menu ngữ cảnh.

Cách xóa chỉ báo.

Để xóa chỉ báo, nhấn nút Đóng từ menu ở phía trên bên phải của khung chỉ báo (menu sẽ hiển thị nếu bạn đặt con trỏ chuột ở gần đó). Menu này cũng cho phép bạn di chuyển khung chỉ báo lên/xuống hoặc phóng to khung.

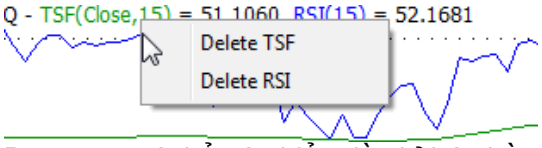


Bạn cũng có thể sử dụng lệnh Đóng từ menu ngữ cảnh hiển thị khi bạn nhấp vào khung biểu đồ bằng nút chuột phải.



Cách xóa biểu đồ chỉ báo khỏi ngăn.

Để xóa một trong các chỉ báo được hiển thị trong ngăn chỉ báo - nhấp chuột PHẢI vào tiêu đề biểu đồ (gần đầu ngăn biểu đồ) và chọn chỉ báo mà bạn muốn xóa.



Bạn cũng có thể xóa biểu đồ chỉ báo bằng tùy chọn Xóa chỉ báo từ menu ngữ cảnh biểu đồ.

Cách thay đổi tham số/màu sắc/kiểu của chỉ báo.

Cửa sổ Tham số cho phép bạn thay đổi tham số, màu sắc và kiểu của chỉ báo. Cửa sổ Tham số được hiển thị khi bạn chèn một chỉ báo mới. Bạn cũng có thể nhấp chuột PHẢI vào ngăn biểu đồ và chọn Tham số từ menu ngữ cảnh. Cửa sổ Tham số hiển thị tất cả các tham số được xác định trong mã AFL của một số chỉ báo nhất định (cũng là các tham số do người dùng xác định) nên nội dung của nó phụ thuộc vào chỉ báo đã chọn. Tuy nhiên - đối với hầu hết các chỉ báo, bạn sẽ thấy:

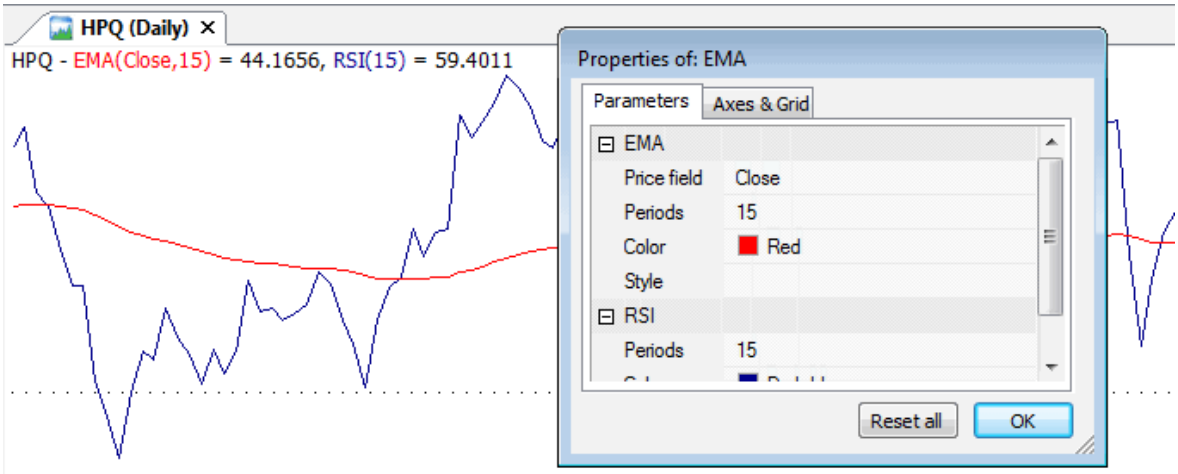
• Trường giá - dữ liệu được sử dụng để tính toán chỉ báo. Nếu 'Trường giá' chứa 'Đóng', điều đó có nghĩa là chỉ báo được tính toán từ giá Đóng. Trường giá không khả dụng cho tất cả các chỉ báo vì không phải tất cả các chỉ báo đều cho phép bạn chọn đầu vào (ví dụ: ADLine).
• Periods - xác định số chu kỳ được sử dụng để tính toán chỉ báo
• Color - cho phép bạn thay đổi màu của chỉ báo
• Style - cho phép bạn xác định kiểu của biểu đồ (các kiểu được mô tả chi tiết hơn trong phần hướng dẫn Sử dụng kiểu và màu biểu đồ.

Cách phủ các chỉ báo với các thang đo khác nhau.

Để có hai (hoặc nhiều) chỉ báo trong một ngăn sử dụng thang đo khác nhau, hãy kéo chỉ báo thứ hai vào chỉ báo thứ nhất, trong cửa sổ Parameters, hãy nhấp vào trường Style và kiểm tra cài đặt StyleOwnScale.
Ví dụ:
Kéo OBV (On Balance Volume) vào ngăn RSI. Sau đó, xác định kiểu là styleOwnScale. Kết quả là - cả hai chỉ báo đều hiển thị và hiển thị đúng.

Cách tạo chỉ báo dựa trên chỉ báo khác.

AmiBroker cũng cho phép bạn dễ dàng tạo các chỉ báo dựa trên giá trị của chỉ báo khác. Tất cả những gì bạn cần làm là nhấn nút chuột TRÁI vào tên chỉ báo, kéo (giữ nút chuột) chỉ báo đã chọn vào ngăn đích và nhả nút. Kết quả là - chỉ báo sẽ được đặt trong khung biểu đồ hiện có. Trong hộp thoại tham số, tham số trường Giá cho biết giá trị cơ sở nào được sử dụng để tính toán chỉ báo.
Ví dụ:
Để tính Trung bình động đơn giản của chỉ báo RSI đã tạo trước đó, hãy kéo chỉ báo MA vào khung RSI. Nội dung của tham số "Trường giá" cho biết Trung bình động được tính toán từ các giá trị RSI(15). (Xem hình ảnh bên dưới).



LƯU Ý: Phần bên dưới chứa thông tin kỹ thuật chỉ dành cho người dùng nâng cao. Người mới bắt đầu có thể bỏ qua phần này.

Sử dụng các hàm Param(), ParamColor(), ParamToggle(), ParamStyle()

Các hàm này, khi được sử dụng trong công thức, cho phép bạn thay đổi cài đặt của chỉ báo trực tiếp từ cửa sổ Parameters.

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

Thêm một tham số mới do người dùng định nghĩa, có thể truy cập thông qua hộp thoại Tham số.

- "name" - định nghĩa tên tham số sẽ được hiển thị trong hộp thoại tham số
- defvalue - định nghĩa giá trị mặc định của tham số
- min, max - định nghĩa giá trị tối thiểu và tối đa của tham số
- step - định nghĩa mức tăng tối thiểu của tham số thông qua thanh trượt trong hộp thoại Tham số
- sincr - định nghĩa mức tăng của giá trị mặc định khi có nhiều hơn một phần cùng loại được chèn (thả) vào biểu đồ. Ví dụ: nếu bạn chèn chỉ báo Trung bình động mặc định vào cùng một ngăn hai lần, thì trung bình động đầu tiên sẽ dựa trên 15 chu kỳ, trung bình động còn lại dựa trên 25 chu kỳ (defvalue=15 + sincr=10)

ParamColor("name", defaultcolor)
Thêm một tham số màu mới do người dùng định nghĩa, có thể truy cập thông qua Tham số dialog.

- "name" - định nghĩa tên tham số sẽ được hiển thị trong hộp thoại tham số
- defaultcolor - định nghĩa giá trị màu mặc định của tham số

Hàm ParamColor cho phép bạn sử dụng colorCycle làm giá trị mặc định. Khi bạn sử dụng tham số colorCycle, màu mặc định sẽ tuần hoàn qua đỏ, xanh lam, xanh lục, ngọc lam, vàng, tím, xanh lục sáng, vàng đậm, khi bạn chèn các chỉ báo của mình vào cùng một ngăn.

ParamStyle("name", defaultval = styleLine, mask = maskDefault) - cho phép chọn các kiểu được áp dụng cho biểu đồ từ cửa sổ Parameters. Ngoài các kiểu có sẵn trong các phiên bản AmiBroker trước đây, còn có hai hằng số kiểu mới:

- styleHidden - sự kết hợp của styleNoDraw | styleNoRescale
- styleDashed - đường đứt nét

Danh sách các kiểu có sẵn được hiển thị trong cửa sổ Parameters phụ thuộc vào tham số mask.

- maskDefault - hiển thị các kiểu dày, nét đứt, ẩn, tỷ lệ riêng (đây là mặt nạ mặc định cho ParamStyle)
- maskAll - hiển thị tất cả các cờ kiểu
- maskPrice - hiển thị dày, ẩn, tỷ lệ riêng, nến, thanh
- maskHistogram - hiển thị biểu đồ, dày, ẩn, tỷ lệ riêng, diện tích

ParamField("name", field = 3) - cho phép chọn trường Giá cho chỉ báo (trường được sử dụng để tính giá trị của chỉ báo). Hàm trả về mảng được xác định bởi tham số trường. Giá trị mặc định = 3 trả về Mảng đóng. Các giá trị có thể có của tham số trường là:

- -1 - ParamField trả về các giá trị của chỉ báo được chèn làm giá trị đầu tiên vào ngăn hoặc Close nếu không có chỉ báo nào
- 0 - trả về mảng Open
- 1 - trả về mảng High
- 2 - trả về mảng Low
- 3 - trả về mảng Close (mặc định)
- 4 - trả về mảng Average = (H+L+C)/3
- 5 - trả về mảng Volume
- 6 - trả về mảng Open Interest
- 7,8,9,.... - trả về các giá trị của chỉ báo được chèn vào ngăn.

ParamToggle("name","values",defaultval=0) - hàm cho phép sử dụng các tham số boolean (Có/Không).

- "name" - tên của tham số
- "values" - giá trị tham số (phân tách bằng ký tự |, ví dụ: "No|Yes" - chuỗi đầu tiên biểu thị giá trị false và chuỗi thứ hai biểu thị giá trị true)
- defaultval - giá trị mặc định của tham số

_

Ví dụ:

Chỉ báo bên dưới cho phép bạn kiểm tra cách các tham số hoạt động trong mã tùy chỉnh. Bạn có thể thay đổi cài đặt từ hộp thoại Tham số.

```
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,colorRed) );
}
```

Giải thích về các hàm đặc biệt: SECTION_BEGIN, _SECTION_END, _SECTION_NAME, _DEFAULT_NAME, _PARAM_VALUES (chỉ dành cho người dùng nâng cao)

Đây là các hàm mới được sử dụng theo cơ chế kéo & thả. Cặp quan trọng nhất là _SECTION_BEGIN("name") và _SECTION_END().

Khi bạn thả công thức vào khung biểu đồ, AmiBroker sẽ thêm công thức bạn đã kéo vào cuối công thức biểu đồ hiện có và bao bọc mã đã chèn bằng các dấu _SECTION_BEGIN("name") và _SECTION_END():

Vì vậy, nếu công thức gốc trông như sau:

```
P = ParamField("Price field",-1);
Periods = Param("Periods", 15, 2, 200, 1, 10 );

Plot( MA( P, Periods ), _DEFAULT_NAME(), ParamColor( "Color", colorCycle ),
ParamStyle("Style") );
```

AmiBroker sẽ chuyển đổi thành:

```
_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();
```

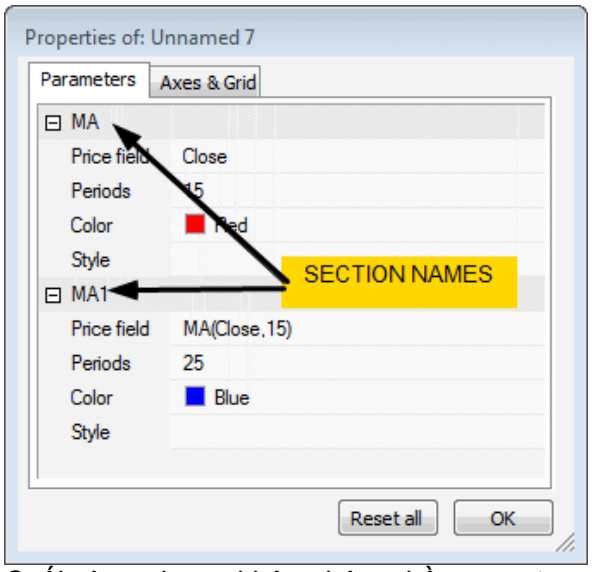
Các dấu _SECTION_BEGIN/_SECTION_END cho phép AmiBroker xác định các phần mã và sửa đổi chúng sau (ví dụ xóa các phần riêng lẻ). Ngoài ra, các phần đó cung cấp cách đảm bảo rằng các tham số có cùng tên trong nhiều phần mã không ảnh hưởng lẫn nhau. Ví dụ, nếu bạn xóa hai đường trung bình động, mã kết quả sẽ trông như sau:

```
_SECTION_BEGIN("MA");
P = ParamField("Trường giá",-1);
Periods = Param("Các chu kỳ", 15, 2, 200, 1, 10);
Plot( MA( P, Periods ), _DEFAULT_NAME(), ParamColor( "Màu", colorCycle ),
ParamStyle("Kiểu") );
_SECTION_END();

_SECTION_BEGIN("MA1");
P = ParamField("Trường giá",-1);
Periods = Param("Các chu kỳ", 15, 2, 200, 1, 10 );

Plot( MA( P, Periods ), _DEFAULT_NAME(), ParamColor( "Color", colorCycle ),
ParamStyle("Style") );
_SECTION_END();
```

Lưu ý rằng tên tham số code và is giống hệt nhau ở cả hai phần. Nếu không có section, các tham số có cùng tên sẽ gây nhiễu. Nhưng nhờ có section được đặt tên duy nhất nên không có xung đột. Điều này là do AmiBroker xác định tham số bằng tên section VÀ tên tham số, vì vậy nếu tên section là duy nhất thì các tham số có thể được xác định duy nhất. Khi thả chỉ báo AmiBroker tự động kiểm tra các tên section đã tồn tại và tự động đánh số các section có tên tương tự để tránh xung đột. Tên section cũng xuất hiện trong hộp thoại Parameter:



Cuối cùng nhưng không kém phần quan trọng: bạn KHÔNG NÊN xóa các dấu _SECTION_BEGIN / _SECTION_END khỏi công thức. Nếu bạn xóa, AmiBroker sẽ không thể nhận dạng các section bên trong công thức đã cho nữa và các tham số có cùng tên sẽ gây nhiễu lẫn nhau.

_SECTION_NAME là một hàm chỉ cung cấp tên của hàm (được đưa ra trong lệnh gọi _SECTION_BEGIN trước đó).

_DEFAULT_NAME là một hàm trả về tên mặc định của plot. Tên mặc định bao gồm tên phần và danh sách các giá trị được phân tách bằng dấu phẩy của các tham số số được xác định trong phần đã cho. Ví dụ trong mã này:

```
_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 sẽ đánh giá thành chuỗi "MA1(Close,15)".

_PARAM_VALUES hoạt động giống như _DEFAULT_NAME ngoại trừ không có tên phần nào được bao gồm (do đó chỉ có danh sách các giá trị tham số được trả về). Vì vậy, trong ví dụ trên _PARAM_VALUES sẽ đánh giá thành chuỗi "(Close, 15)".

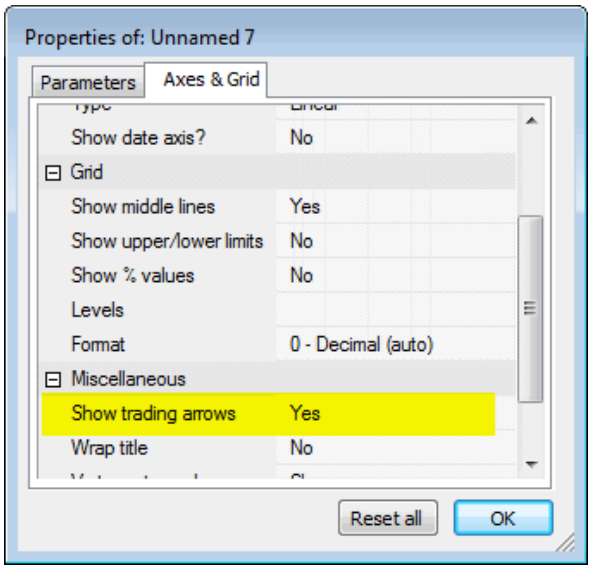
Câu hỏi thường gặp về chức năng kéo và thả

Hỏi. Sự khác biệt giữa tùy chọn Chèn và Chèn được liên kết trong menu biểu đồ là gì?

A. Lệnh Chèn tạo bản sao của tệp công thức gốc và đặt bản sao đó vào thư mục kéo-thả ẩn để công thức gốc sẽ không bị ảnh hưởng bởi việc chỉnh sửa tiếp theo hoặc chồng các chỉ báo khác lên đó. Nhấp đúp vào tên công thức trong cây biểu đồ tương đương với việc chọn lệnh Chèn từ menu. Mặt khác, lệnh Chèn được liên kết không tạo bất kỳ bản sao nào của công thức. Thay vào đó, nó tạo khung biểu đồ mới liên kết trực tiếp đến công thức gốc. Theo cách này, việc chỉnh sửa tiếp theo và/hoặc chồng các chỉ báo khác sẽ sửa đổi bản gốc

Hỏi. Tôi không thể thấy mũi tên mua/bán từ hệ thống giao dịch của mình

A. Mũi tên giao dịch có thể được hiển thị trên bất kỳ khung biểu đồ nào (không chỉ một biểu đồ giá tích hợp). Tuy nhiên, theo mặc định, màn hình mũi tên bị TẮT. Để BẬT, bạn phải mở hộp thoại Tham số, chuyển sang "Trục và lưới" và chuyển tùy chọn "Hiển thị mũi tên giao dịch" thành "Có".



H. Đọc tôi nói rằng: "Cửa sổ công thức Phân tích tự động hiện cũng là mục tiêu kéo & thả (bạn có thể kéo công thức và tệp AFL vào đó)". Điều đó có nghĩa là gì?

Đ. Điều đó có nghĩa là bạn có thể kéo công thức từ cây Biểu đồ hoặc tệp .AFL từ Windows Explorer và thả vào cửa sổ công thức Phân tích tự động (AA) và nó sẽ tải công thức vào cửa sổ AA. Đây là một giải pháp thay thế cho việc tải công thức thông qua nút "Tải" trong cửa sổ AA.

H**. Tôi có thể thả một phím tắt vào cửa sổ công thức không?**

Đ: Không, bạn không thể. Bạn chỉ có thể kéo & thả các tệp có phần mở rộng .AFL (các phím tắt trong Windows có phần mở rộng .lnk).

H. Tôi có thể thêm công thức của riêng mình vào cây Biểu đồ không?

Đ. Có, bạn có thể. Chỉ cần lưu công thức .AFL của bạn vào thư mục con Formulas của thư mục AmiBroker và nó sẽ xuất hiện trong cây "Charts" (Có thể cần View->Refresh All để đọc lại thư mục nếu bạn đang sử dụng trình soạn thảo bên ngoài)

H. Tôi đã thêm tệp mới vào thư mục Formulas, nhưng nó không hiển thị trong cây Charts trừ khi tôi khởi động lại AmiBroker? Có cách nào để làm mới Chart tree không?

A. Bạn có thể làm mới Chart tree bằng cách chọn menu View->Refresh All.

H. Nếu tôi sửa đổi công thức đi kèm với AmiBroker, công thức đó có bị ghi đè khi nâng cấp tiếp theo không?

A. Có, công thức sẽ bị ghi đè. Nếu bạn muốn thực hiện bất kỳ sửa đổi nào đối với các công thức được cung cấp cùng với AmiBroker, vui lòng lưu các phiên bản đã sửa đổi của bạn dưới tên mới hoặc (tốt hơn) trong thư mục con tùy chỉnh của riêng bạn.

H. Tôi có thể thấy nút Reset All trong hộp thoại Parameters nhưng nó đặt tất cả các tham số về giá trị mặc định. Có cách nào để đặt lại SINGLE parameter không?

A. Không, hiện tại chưa có tùy chọn nào như vậy, nhưng nó sẽ được thêm vào trong các bản beta sắp tới.

H. Tôi đã kéo RSI vào khung biểu đồ giá và thấy một đường thẳng màu đỏ ở cuối khung. Có vấn đề gì vậy?

A. Khi bạn thả hai chỉ báo/biểu đồ có giá trị khác nhau đáng kể, bạn phải sử dụng kiểu OwnScale cho một trong số chúng. Bạn có thể bật kiểu OwnScale bằng hộp thoại Tham số. Điều này đảm bảo rằng các thang đo được sử dụng cho từng chỉ báo là độc lập và bạn có thể nhìn thấy chúng một cách chính xác. Nếu không, chúng sử dụng một thang đo chung phù hợp với cả hai phạm vi giá trị, dẫn đến các biểu đồ bị làm phẳng.

H. Màu xám nhạt của các hàm đặc biệt AFL mới_SECTION_BEGIN, v.v. khiến chúng trở nên vô hình trong màu nền xanh xám IB của tôi. Làm thế nào để tôi có thể thay đổi màu của các hàm đặc biệt?

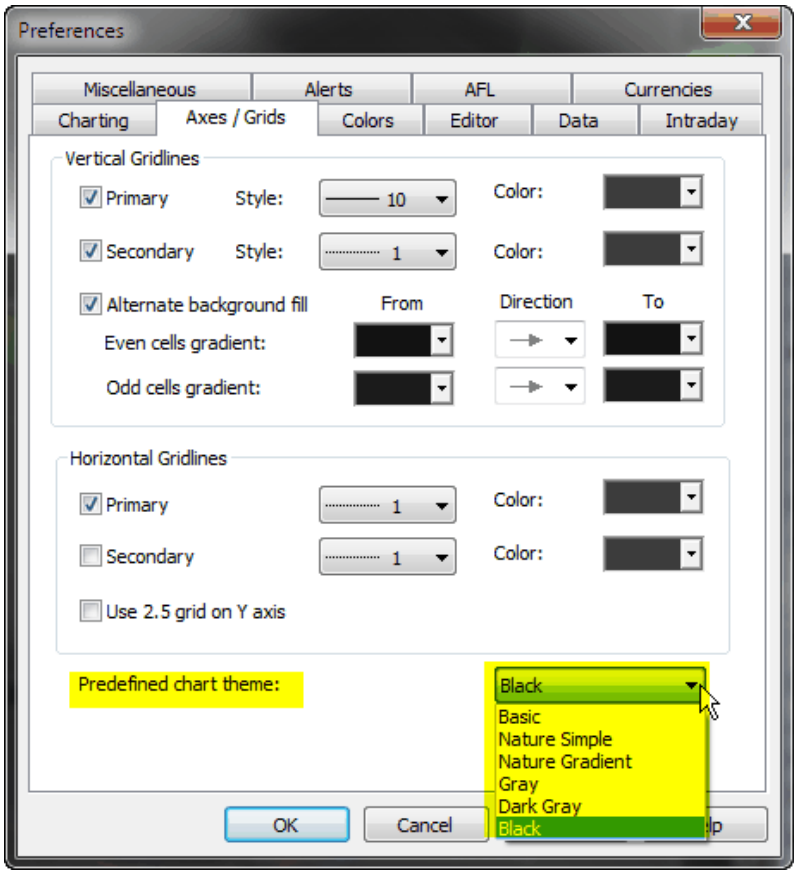
A. Hiện tại, bạn không thể. Nhưng sẽ có một thiết lập để tô màu các hàm đặc biệt trong phiên bản tiếp theo.

H. Khi tôi thả chỉ báo, hộp thoại Tham số không hiển thị tất cả các tham số. Điều này có đúng không?

A. Có, nó hoạt động theo cách đó. Ý tưởng đằng sau nó rất đơn giản. Khi bạn thả chỉ báo mới, AmiBroker sẽ hiển thị hộp thoại với các tham số CHỈ dành cho chỉ báo hiện đang thả. Điều này nhằm đảm bảo rằng các tham số chỉ báo mới được chèn sẽ hiển thị rõ ràng (ở trên cùng) và người dùng mới không bị choáng ngợp bởi hàng chục tham số khác tham chiếu đến các chỉ báo đã thả trước đó. Mặt khác, khi bạn chọn mục "Tham số" từ menu ngữ cảnh thì TẤT CẢ các tham số sẽ hiển thị - cho phép bạn sửa đổi tất cả chúng bất kỳ lúc nào sau đó.

Chủ đề biểu đồ

AmiBroker 5.52 giới thiệu 6 chủ đề biểu đồ được xác định trước có thể chuyển đổi trong Tools->Preferences, tab "Axes & Grid":



1. Chủ đề cơ bản



2. Chủ đề Nature Simple



3. Chủ đề Nature Gradient



4. Chủ đề Gray


AmiBroker - [GE - GEN ELECTRIC CO - Daily]
File Edit View Insert Format Symbol Analysis Tools Window Help
GE
GE (Daily)
GE - Daily 2012-01-03 Open 18.23, Hi 18.5, Lo 18.23, Close 18.36 (2.5%) MA(Close,15) = 17.3520
18.36
17.352
Oct
Nov
Dec
2012
GE - MACD(12,26) = 0.6018, Signal(12,26,9) = 0.4913
0.60178
0.49132
Sheet 1 Sheet 2 Sheet 3 Sheet 4 Sheet 5 Sheet 6 Sheet
Realtime Quote
For Hel
NYSE, Group 1, Conglomerates, Conglome
Data
3954M
Alert Output
Symbols
Layouts
Layers
Charts

5. Chủ đề Dark Gray



6. Chủ đề Black



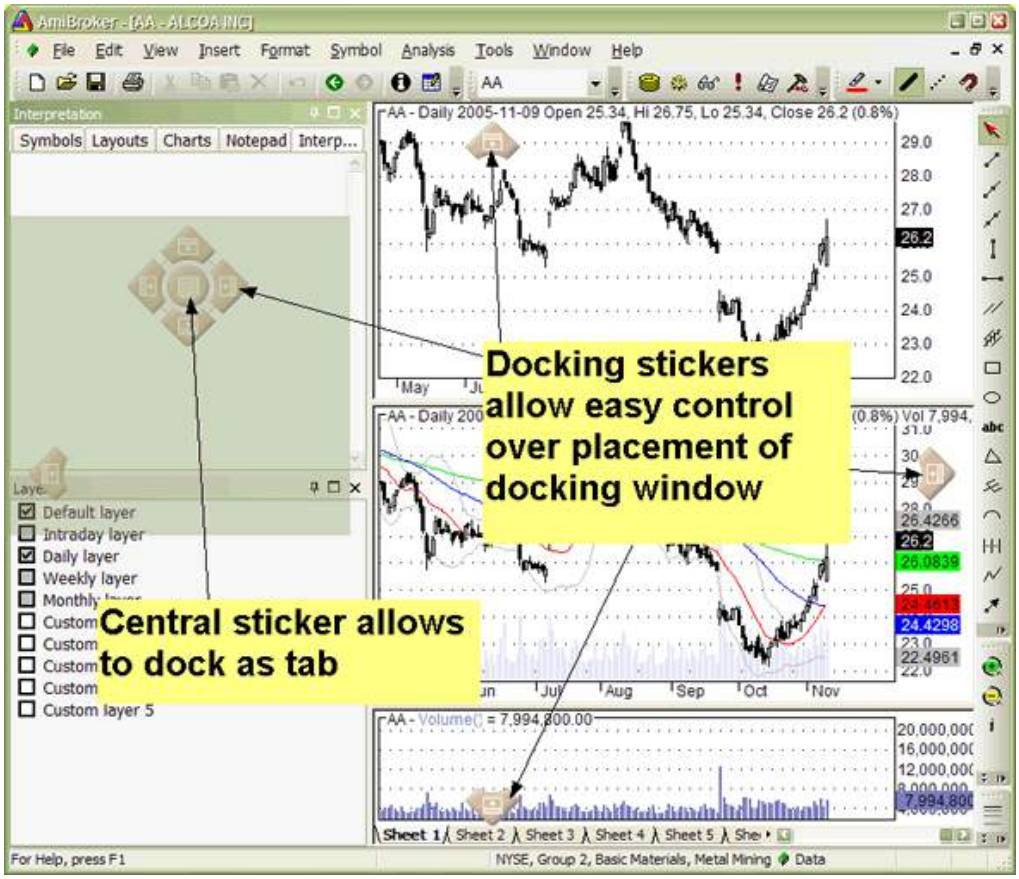
Tùy chỉnh giao diện người dùng

Giao diện người dùng tùy chỉnh mới được giới thiệu có một số tính năng tuyệt vời cho phép kiểm soát hoàn toàn giao diện người dùng AmiBroker.

Tab gắn kết/xé lồng nhau nâng cao



Để gắn một khung vào bất kỳ bên nào của ứng dụng hoặc dưới dạng tab, chỉ cần nhấp vào thanh tiêu đề cửa sổ gắn kết và kéo nó. Nếu bạn thực hiện thao tác này, các nhãn dán gắn kết sẽ hiển thị để giúp bạn dễ dàng chọn vị trí đích như hiển thị bên dưới



Bạn cũng có thể nhấp vào tab khung gắn kết và kéo nó (xé) và gắn thành một cửa sổ riêng biệt. Theo cách này, bạn có thể sắp xếp tất cả các cửa sổ gắn kết thành các cửa sổ riêng biệt hoặc thành các tab hoặc kết hợp cả hai cách tiếp cận này. Bạn cũng có thể làm cho cửa sổ/tab nổi nếu bạn kéo nó trong khi giữ phím CTRL.

Tự động ẩn khung trượt

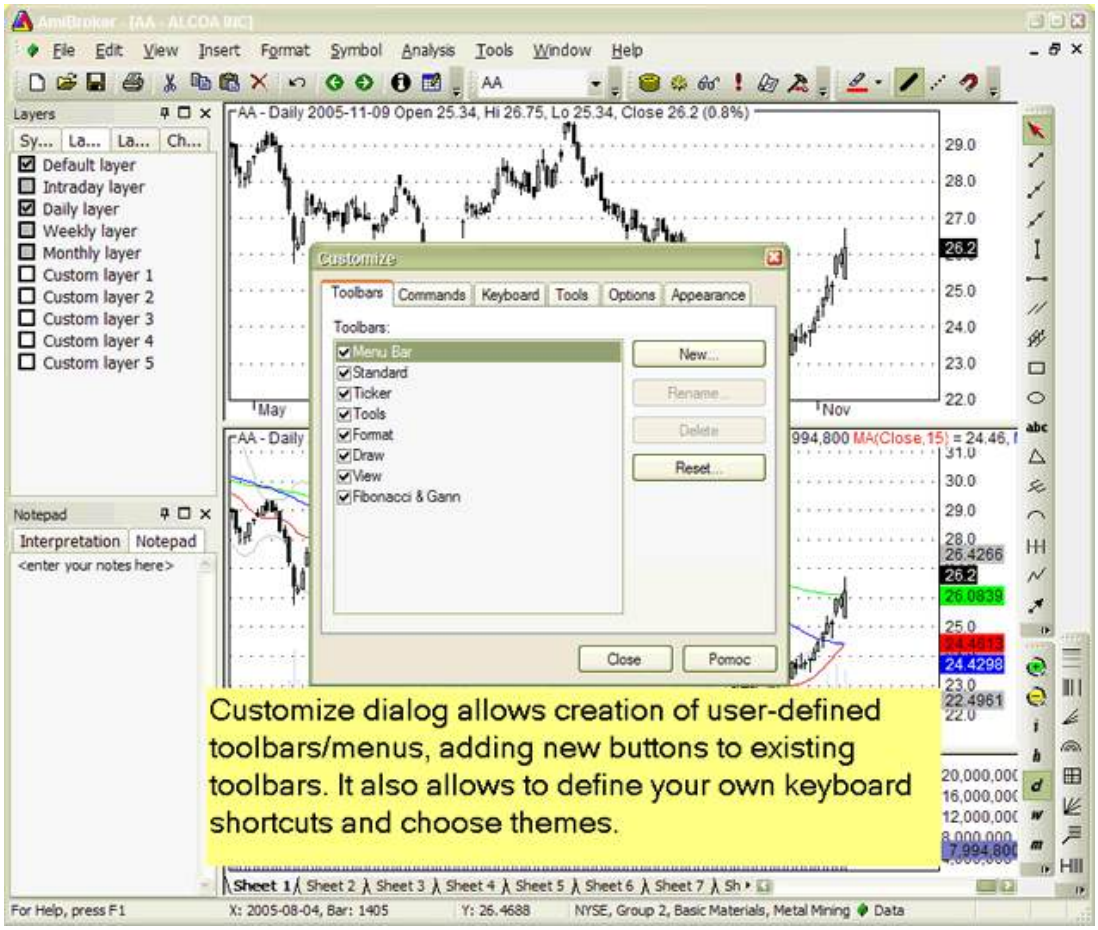
Một tính năng rất hữu ích khác cho phép tiết kiệm không gian quý giá trên màn hình của bạn là tự động ẩn khung. Để kiểm soát (bật/tắt) tính năng này, có một nút ghim ở góc trên bên phải của mỗi cửa sổ gắn kết. Nếu bạn bỏ ghim, khung sẽ tự động ẩn khi không còn được chú ý nữa.


Pinup (auto-hide) button
Auto-slide pane shown


Pane hidden automatically

Thanh công cụ, menu và phím tắt tùy chỉnh nâng cao

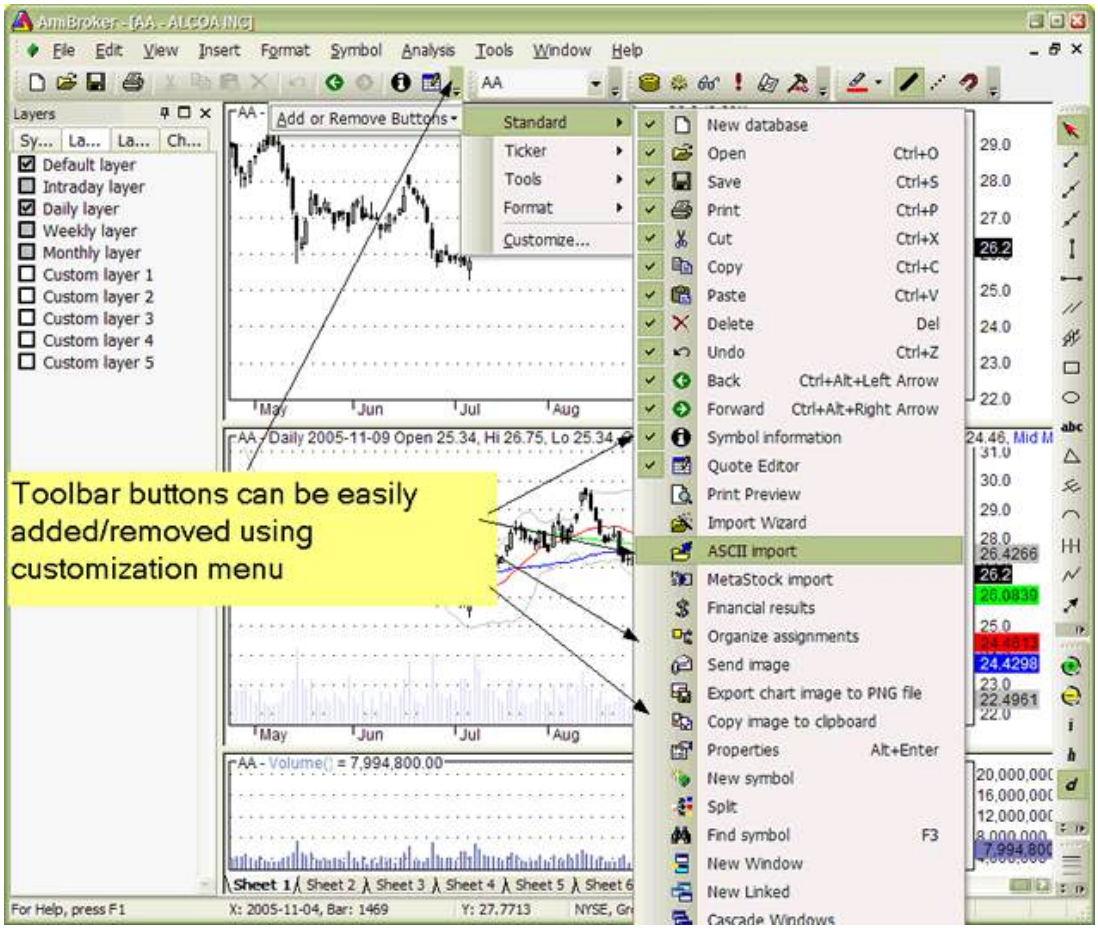
Giao diện người dùng mới cho phép người dùng kiểm soát hoàn toàn giao diện, bố cục và vị trí của tất cả các thanh công cụ, nút và menu. Giao diện này cho phép bạn thêm các nút của riêng mình, xóa/sắp xếp lại các nút hiện có. Ngoài ra, bạn cũng có thể định nghĩa hoặc định nghĩa lại các phím tắt mới/hiện có. Tất cả các tính năng tùy chỉnh này đều có sẵn từ menu Tools->Customize hoặc từ menu Customize chevron.



Menu Chevron có sẵn từ nút mũi tên nhỏ được đặt ở cuối dải thanh công cụ. Nó cho phép truy cập các thành phần tự động ẩn của thanh công cụ cũng như các tính năng tùy chỉnh.



Menu phụ Add or Remove buttons cho phép nhanh chóng hiển thị/ẩn các nút thanh công cụ theo sở thích của bạn. Ở chế độ tùy chỉnh (khi bạn vào bằng Tools->Customize, bạn cũng có thể di chuyển các nút xung quanh để thay đổi thứ tự xuất hiện của chúng và bạn cũng có thể thay đổi kích thước các trường chỉnh sửa và trường kết hợp (chẳng hạn như trường chọn ticker) bằng cách chọn chúng trước và thay đổi kích thước đường viền sẽ hiển thị sau khi thực hiện lựa chọn.



Bạn thậm chí có thể thêm và thiết kế các nút của riêng mình bằng trình chỉnh sửa hình ảnh tích hợp:



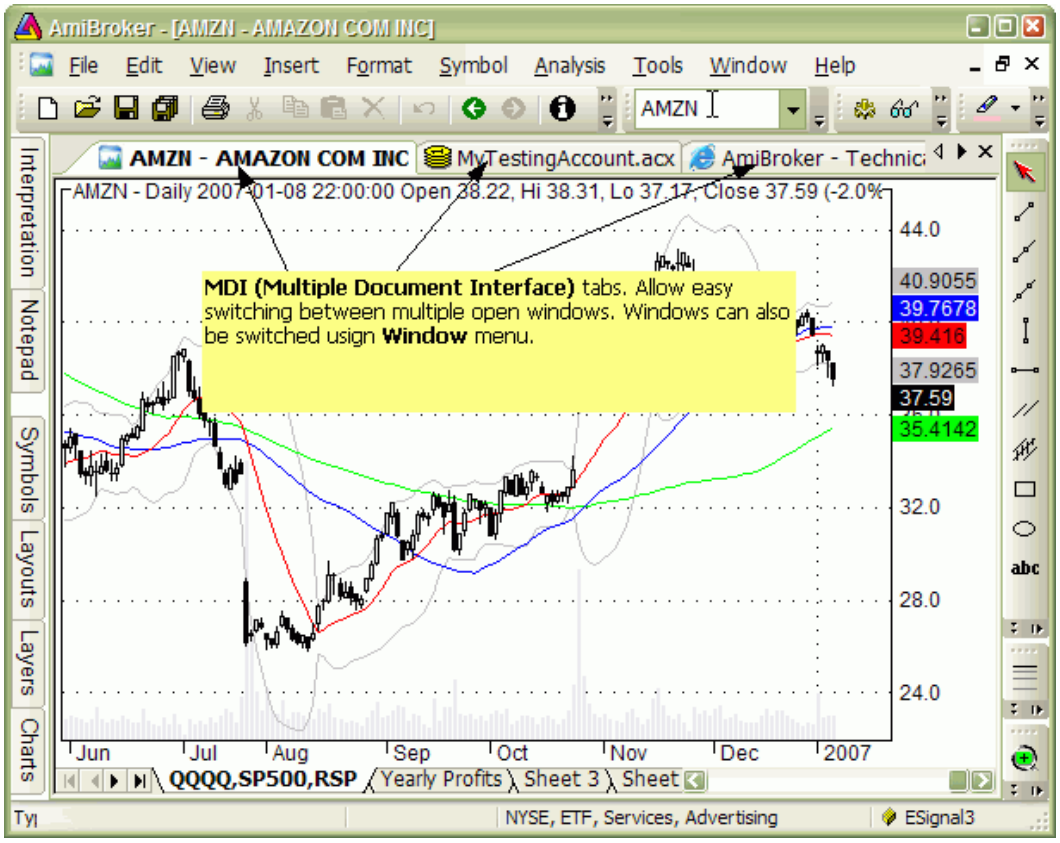


Giao diện theo chủ đề

AmiBroker cũng cho phép bạn chọn "giao diện" hoặc "chủ đề" giao diện người dùng ưa thích của mình để phù hợp với sở thích cá nhân.



Tab MDI (giao diện nhiều tài liệu)



AmiBroker là ứng dụng giao diện nhiều tài liệu (MDI). Nói tóm lại, ứng dụng này cho phép bạn mở và làm việc với nhiều cửa sổ cùng một lúc. Để tìm hiểu thêm về MDI là gì, bạn có thể xem bài viết này: http://en.wikipedia.org/wiki/Multiple_document_interface

Bây giờ, các tab MDI (hiển thị trong hình trên) chỉ là một cách bổ sung để chuyển đổi nhiều cửa sổ đang mở (ngoài menu Cửa sổ, nơi cũng có danh sách các cửa sổ tài liệu đang mở).

Điều quan trọng là phải hiểu rằng các tab MDI không phải là "có thể định nghĩa bởi người dùng" theo nghĩa là bạn không thể tự do định nghĩa tên của chúng, không giống như các trang biểu đồ (có thể định nghĩa được). Tên của chúng được tự động lấy từ

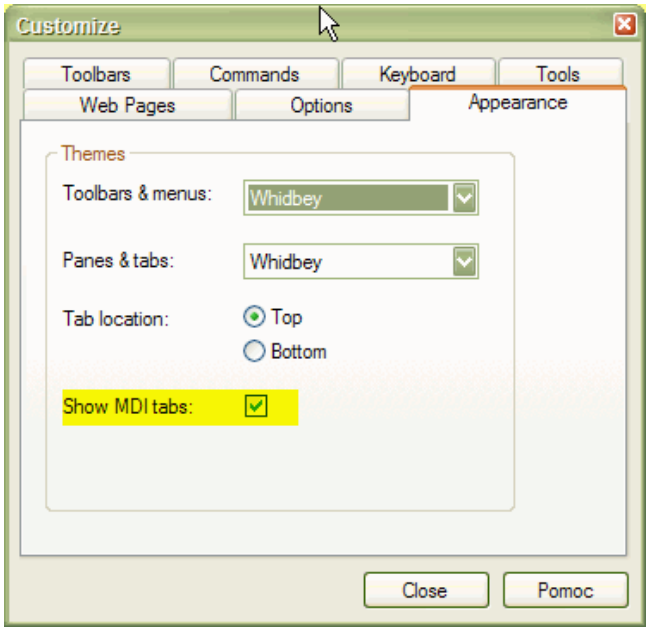
tên tài liệu/cửa sổ. Đối với cửa sổ biểu đồ, tên luôn ở định dạng: Biểu tượng - Tên đầy đủ, cửa sổ trình duyệt web sử dụng tiêu đề trang HTML (theo định nghĩa của tài liệu HTML), cửa sổ quản lý tài khoản sử dụng tên tệp tài khoản thực tế (mà bạn có thể chọn khi lưu chúng).
Các tab MDI về cơ bản là trình chuyển đổi cửa sổ tài liệu (giống như THANH TÁC VỤ của Windows ở phía dưới) và chúng được AmiBroker tự động quản lý bất cứ khi nào bạn mở cửa sổ mới hoặc đóng cửa sổ.

Và nó hoạt động chính xác bằng cùng một ý tưởng như thanh tác vụ của Windows. Chúng ta hãy xem xét kỹ hơn phép so sánh này: Khi bạn sử dụng Thanh tác vụ của Windows:
- bạn mở ứng dụng - một nút mới trên thanh tác vụ sẽ xuất hiện
- và bạn có thể chuyển đổi giữa các ứng dụng đang mở bằng các nút trên thanh tác vụ.
- bạn không thể đổi tên nút vì nó đại diện cho tên ứng dụng.
- và bạn cần cẩn thận khi mở quá nhiều ứng dụng vì tất cả các ứng dụng đang mở đều tiêu tốn tài nguyên hệ thống

Bây giờ sử dụng tab AmiBroker MDI:

- bạn mở tài liệu (cửa sổ) -> một nút (tab) mới xuất hiện
- bạn có thể chuyển đổi giữa các cửa sổ đang mở bằng các nút (tab)
- bạn không thể đổi tên nút vì nó đại diện cho tên tài liệu/cửa sổ
- và bạn cần cẩn thận khi mở quá nhiều tài liệu/cửa sổ vì tất cả các tài liệu đang mở đều tiêu tốn tài nguyên hệ thống

Bạn có thể tắt tab MDI bằng cách bỏ chọn hộp "Hiển thị tab MDI" trong trang Công cụ->Tùy chỉnh, Giao diện, như được hiển thị bên dưới:



Ghi chú lịch sử: Trong các phiên bản trước 4.90, để chuyển đổi tài liệu, bạn cần sử dụng menu Cửa sổ. Bây giờ ngoài ra bạn có thể sử dụng tab. Nhưng đây chỉ là tính năng tiện lợi, thông tin chi tiết hơn tại: http://en.wikipedia.org/wiki/Tabbed_Document_Interface (Lưu ý rằng các liên kết wikipedia mô tả TDI / MDI đã lỗi thời và AmiBroker thực sự kết hợp các ưu điểm của CẢ hai phương pháp TDI và MDI (ví dụ bạn có thể xếp ô cửa sổ trong TDI của AB)

Để biết thêm thông tin, hãy xem bài thuyết trình tại hội nghị Houston:
http://www.amibroker.com/docs/Houston1.pdf (định dạng PDF),
http://www.amibroker.com/docs/Houston1.html (định dạng Flash).

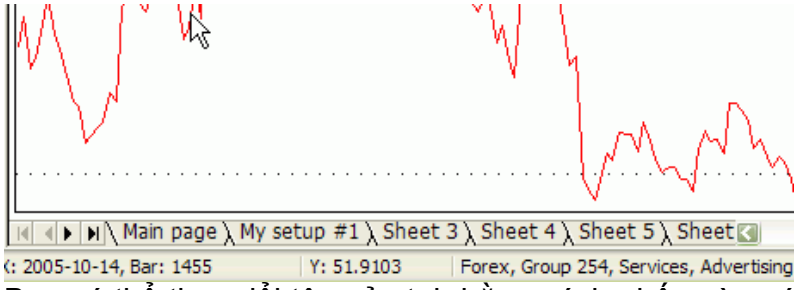
Làm việc với Bảng biểu đồ và Bố cục cửa sổ

AmiBroker quản lý nhiều bảng biểu đồ và **bố cục đa cửa sổ** với khả năng tải / lưu nhanh chúng. Tính năng này giúp bạn chuyển đổi nhanh chóng giữa các bộ chỉ số khác nhau, tiết kiệm thời gian đáng kể.

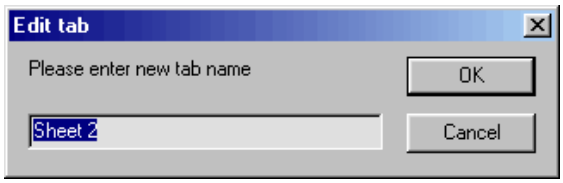
Bảng Biểu đồ và Mẫu

Một bảng biểu đồ là một bộ các bảng biểu đồ (với các chỉ số) được hiển thị trong một khung duy nhất.

Bạn có thể chuyển đổi giữa các bảng khác nhau bằng cách nhấp vào các thẻ nằm ở phía dưới cửa sổ AmiBroker như hình dưới đây:

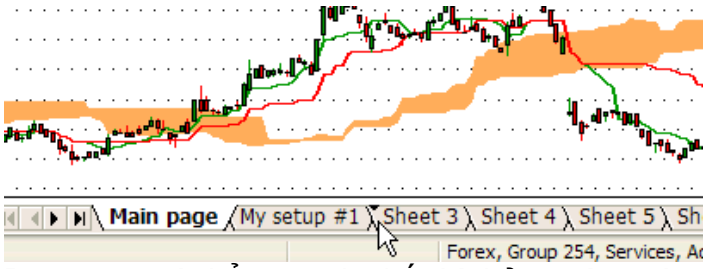


Bạn có thể thay đổi tên của tab bằng cách nhấp vào nó bằng nút chuột PHẢI, vì vậy cửa sổ sau xuất hiện:

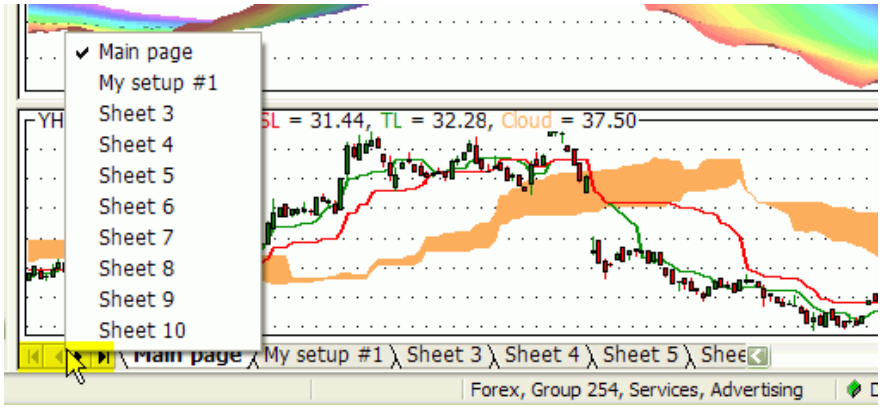


Bạn có thể thay đổi tất cả bốn tên thẻ (một cách lần lượt) để chúng được mô tả rõ hơn (và chúng liên quan đến nội dung của bảng).

Bạn có thể cuộn các thẻ sử dụng nút hình mũi tên và bạn có thể sắp xếp chúng lại bằng cách kéo (nhấp vào thẻ, giữ nút chuột trái và kéo đến vị trí mong muốn - một mũi tên sẽ chỉ vị trí đích).



Bạn cũng có thể truy cập bất kỳ bảng nào một cách nhanh chóng bằng cách nhấp chuột PHẢI vào mũi tên để mở menu pop-up liệt kê tất cả các tab và cho phép lựa chọn tức thì (không cần cuộn)



Bước tiếp theo là thiết lập bảng của bạn theo sở thích cá nhân. Chỉ cần thêm/bỏ các khung biểu đồ vào/từ mỗi bảng. Như vậy, bạn có thể có tới 60 bộ chỉ số khác nhau mà bạn có thể gọi lại rất nhanh bằng cách chuyển sang tab phù hợp. Số lượng thực sự của bảng được định nghĩa trong **Tools->Preferences->Charting "Number of chart sheets"**.

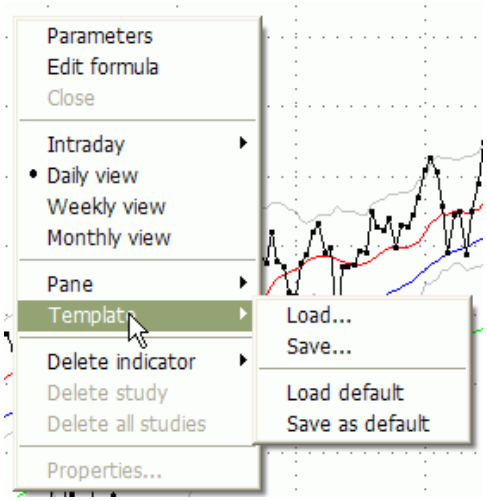
Tất cả các bảng biểu đồ được gọi chung là "template" và bạn có thể làm cho cài đặt này trở nên vĩnh viễn chỉ bằng cách nhấp chuột phải lên biểu đồ và chọn mục menu sau **(Template->Save, Template->Save as default):**

Mẫu mặc định được sử dụng nếu bạn tạo cửa sổ mới **(Window->New)**

Bạn cũng có thể tải mẫu đã lưu bằng cách chọn **Template->Load** từ menu nút chuột phải của biểu đồ.

Thêm vào đó, một định dạng mới đã được thêm vào mẫu định dạng cũ với phần .chart mở rộng, giữ không chỉ kích thước cửa sổ và tham chiếu công thức (đường dẫn) mà còn cả công thức tự thân, vì vậy tất cả những gì bạn cần làm là lưu biểu đồ của mình vào một tệp (Chart Template, Complete *.chart) và sao chép tệp đó vào máy tính khác và biểu đồ sẽ được tạo lại với tất cả công thức liên kết với nó.

Để lưu biểu đồ dưới định dạng mới, hãy thực hiện như sau:



1. Nhấp chuột PHẢI vào biểu đồ và chọn **Template->Save…**

2. Trong hộp thoại tệp, chọn **"Files of type"** và sau đó chọn **"Chart Template, Complete (*.chart)"**

3. Đặt tên cho tệp và nhấn **Save**.

Để tải lại biểu đồ đã lưu trước đó, hãy thực hiện các bước sau:

1. Nhấp CHUỘT PHẢI lên biểu đồ và chọn **Template->Load...**

2. Trong hộp thoại tệp, chọn tệp *.chart đã lưu trước đó và nhấn **"Open"**

Lưu ý: Quy trình mà AmiBroker thực hiện như sau: Khi bạn lưu biểu đồ dưới định dạng mới, nó sẽ lưu tệp XML với:

a) Tên của tất cả các trang, bảng, kích thước của chúng, vị trí và các cài đặt khác
b) Đường dẫn đến tất cả các công thức được sử dụng bởi tất cả các bảng
c) Nội dung của chính các công thức

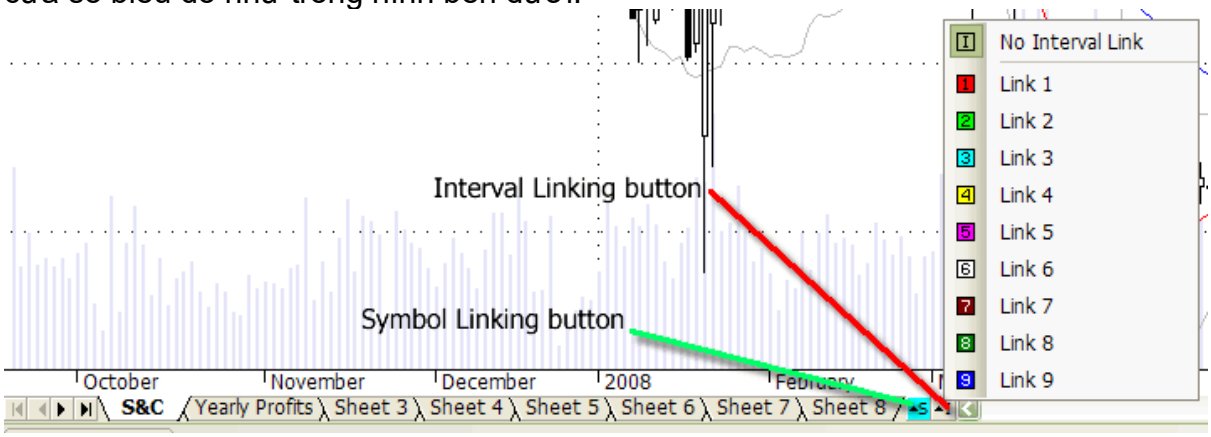
Khi bạn tải biểu đồ dưới định dạng mới AmiBroker:
a) Cài đặt các trang/bảng theo thông tin được lưu trữ trong tệp
b) Đối với mỗi công thức được lưu trong tệp, nó sẽ kiểm tra xem công thức đó có tồn tại trên máy tính chủ hay không:
- Nếu không tồn tại - nó sẽ tạo ra một công thức mới
- Nếu tồn tại và nội dung giống với công thức được lưu trong tệp .chart nó sẽ không làm gì
- Nếu tồn tại nhưng nội dung khác thì nó sẽ tạo ra một tệp công thức MỚI với hậu tố _imported.afl (vì vậy tệp cũ không bị thay đổi) và sẽ tham chiếu bảng đến công thức _imported.afl thay thế.

LƯU Ý QUAN TRỌNG: nếu bạn sử dụng bất kỳ tệp #include nào, AmiBroker sẽ lưu nội dung của các tệp include bên trong tệp biểu đồ và sẽ cố gắng phục hồi chúng trên máy tính chủ. Xin lưu ý rằng trong trường hợp của include, nó sẽ kiểm tra xem nó có tồn tại và nếu nó có khác không. Nếu cả hai điều kiện đều được đáp ứng (tệp khác đã tồn tại) nó sẽ hỏi xem có thay thế hay không. Nếu bạn chọn thay thế - nó sẽ thay thế và tạo bản sao lưu của tệp hiện có với phần mở rộng .bak. Nếu bạn đang sử dụng bất kỳ tệp nào trong "tệp include chuẩn" và include chúng bằng dấu ngoặc <>, AmiBroker sẽ khôi phục tệp trong thư mục include chuẩn của máy tính chủ cũng như vậy (ngay cả khi đường dẫn thư mục include chuẩn khác nhau trên máy nguồn).

Định dạng .chart mới dự kiến sẽ được sử dụng trong việc chuyển giao các biểu đồ giữa các máy tính khác nhau. Để lưu trữ các bố cục/mẫu trên máy tính cục bộ, bạn nên sử dụng các định dạng cũ vì chúng tiêu tốn ít dung lượng hơn (chỉ lưu trữ các tham chiếu, không phải các công thức). Tuy nhiên, bạn có thể sử dụng định dạng mới cho mục đích lưu trữ vì nó giữ cả công thức và tất cả các tham chiếu trong một tệp, rất thuận tiện cho việc sao lưu.

Liên kết Biểu tượng và Khoảng thời gian

Bây giờ, bạn có thể liên kết các cửa sổ biểu đồ theo biểu tượng và/hoặc khoảng thời gian. Để liên kết các cửa sổ biểu đồ, sử dụng các nút liên kết nằm ở dưới cùng của cửa sổ biểu đồ như trong hình bên dưới:

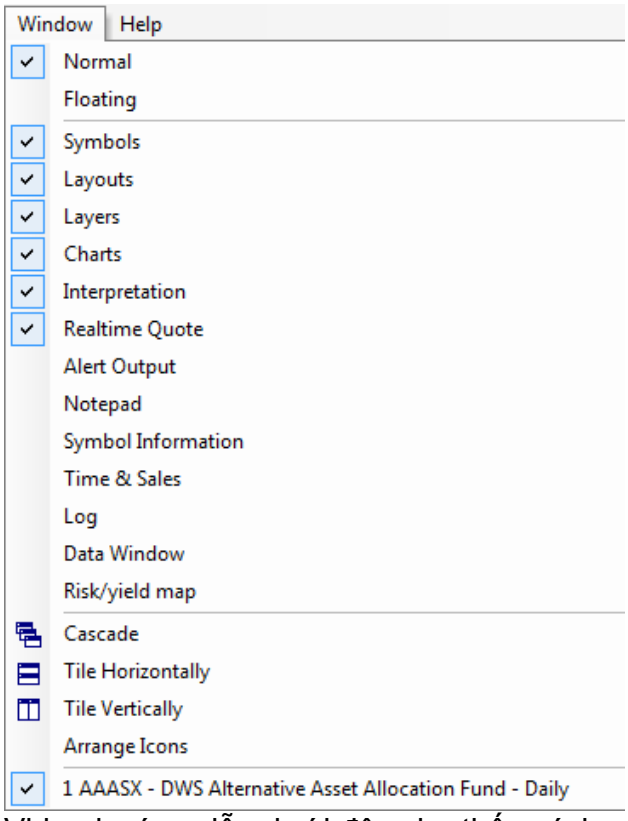


Các nút "S" và "I" màu xám có nghĩa là không có liên kết. Mọi màu sắc khác (đỏ, xanh lá, đỏ tím, vàng, hồng, trắng, nâu, xanh đậm, xanh lam) có nghĩa là biểu đồ đó thuộc về nhóm liên kết được mã màu tương ứng. Tất cả các cửa sổ có màu liên kết giống nhau sẽ chuyển đổi biểu tượng và/hoặc khoảng thời gian đồng thời.

Cửa sổ Nổi

Nếu bạn đang sử dụng nhiều màn hình, bạn có thể thấy sự hữu ích của việc hiển thị các biểu đồ AmiBroker trên nhiều cửa sổ. Để đơn giản hóa, AmiBroker 5.10 giới thiệu các cửa sổ biểu đồ "nổi". Thông thường, tất cả các cửa sổ biểu đồ nằm trong cửa sổ ứng dụng chính của AmiBroker. Nếu bạn làm cho cửa sổ biểu đồ nổi, bạn đang tách cửa sổ biểu đồ ra khỏi khung AmiBroker chính, cho phép bạn di chuyển nó ra ngoài, ví dụ như sang màn hình khác.

Bạn có thể chuyển đổi giữa trạng thái bình thường và nổi bằng cách sử dụng menu Cửa sổ như được hiển thị bên dưới:



Video hướng dẫn dưới đây cho thấy cách sử dụng cửa sổ nổi và liên kết biểu tượng trong thực tế: http://www.amibroker.com/video/FloatAndLink.html

Các Bố Cục Cửa Sổ

Một bố cục cửa sổ là một bộ hoàn chỉnh gồm nhiều cửa sổ mở, mỗi cửa sổ có biểu tượng, khoảng thời gian hiển thị, kích thước, bộ biểu đồ khác nhau.

Hình dưới đây cho thấy bố cục 4 cửa sổ, mỗi cửa sổ có một bộ khung chỉ báo khác nhau. Bên trái bạn có thể thấy bảng "Layouts" trong cửa sổ Workspace hiển thị danh sách các bố cục được lưu trữ địa phương và toàn cầu.



Sử dụng AmiBroker 4.20, bạn giờ đây có thể sở hữu số lượng không giới hạn các mẫu cửa sổ tùy chỉnh, nhiều cửa sổ mà bạn có thể chuyển đổi giữa chúng chỉ bằng cách nhấp đúp vào tên bố cục trong tab **"Layouts"** của cửa sổ Workspace.

Bạn có thể **mở, lưu, xóa** bố cục bằng cách nhấp chuột phải vào cây **Layout** và chọn chức năng thích hợp. Tùy chọn **"Save As"** sẽ lưu bố cục hiện tại với tên mới.

Các bố cục địa phương là theo từng cơ sở dữ liệu trong khi **các bố cục Toàn cầu** có thể nhìn thấy từ tất cả cơ sở dữ liệu.

Thông tin được lưu trong các bố cục bao gồm: kích thước cửa sổ và vị trí, trạng thái cửa sổ tối đa hóa/ tối thiểu hóa, các khung biểu đồ có sẵn trên mỗi trang (độc lập cho mỗi cửa sổ), khoảng thời gian của thanh đã chọn, biểu tượng đã chọn, tờ biểu đồ đã chọn.

Bố cục được sử dụng gần đây nhất có thể được lưu tự động khi thoát và chuyển đổi cơ sở dữ liệu (xem: **Tools->Preferences->Miscellaneous** "Save on exit: Layouts").

Lưu ý: từ phiên bản 4.90, các cửa sổ đa có thể được chuyển đổi không chỉ bằng cách sử dụng menu Window theo kiểu cũ mà còn có thể sử dụng các thẻ MDI mới. Thông

tin thêm về các thẻ MDI có thể được tìm thấy trong chương "Tùy chỉnh giao diện ng dùng".

Sử Dụng Lớp (Layers)

Lớp là gì?

Lớp giống như những mảnh nhựa trong suốt. Bạn có thể vẽ lên chúng. Lớp có thể được tạo hiển thị hoặc ẩn đi. Điều này cho phép hiển thị/ẩn những bản vẽ trên từng lớp nhất định mà không ảnh hưởng đến bản vẽ trên các lớp khác.

Làm thế nào để làm việc với lớp

Đầu tiên, hãy chắc chắn rằng cửa sổ Workspace nhìn thấy đươc (Window->Layers)

Sau đó chuyển đến thẻ "Layers". Ở đây bạn có thể thấy danh sách các lớp được định nghĩa trước.

Các hộp kiểm ở phía bên trái của mỗi lớp điều khiển khả năng hiển thị của lớp. Nếu hộp kiểm được đánh dấu thì lớp đó sẽ hiển thị, nếu không - lớp sẽ không hiển thị. Ban đầu, năm lớp đầu tiên sẽ bị "khoá" vào các khoảng thời gian.
Những lớp được tích hợp này bao gồm:

- Lớp mặc định - luôn hiển thị
- Lớp intraday - chỉ hiển thị khi xem biểu đồ ngày trong ngày
- Lớp hàng ngày - chỉ hiển thị khi xem biểu đồ hàng ngày
- Lớp hàng tuần - chỉ hiển thị khi xem biểu đồ hàng tuần
- Lớp hàng tháng - chỉ hiển thị khi xem biểu đồ hàng tháng

A locked layer changes its visibility automatically when interval changes and you can not change its visibility by clicking on the left-side checkbox. The remaining layers are not locked and they can be shown/hidden freely by marking the checkbox. To draw a study in a given layer simply a) SELECT the layer first (click on name to highlight it) b) DRAW the study as usual As long you select the other layer all drawings will be placed on selected layer. After drawing a study you can assign it to any other layer via object properties box.

Một lớp khóa sẽ tự động thay đổi độ hiển thị khi thay đổi khoảng thời gian, và bạn không thể thay đổi độ hiển thị của nó bằng cách nhấp vào ô kiểm bên trái.

Các lớp còn lại không bị khóa và có thể được hiển thị/ẩn tự do bằng cách đánh dấu vào ô kiểm.

Để vẽ một nghiên cứu trên một lớp nhất định, bạn thực hiện các bước sau:

a. Chọn Lớp (Nhấp vào tên lớp để làm nổi bật nó)
b. Tiến hành vẽ như thường lệ.

Từ khi bạn chọn lớp khác thì tất cả các bản vẽ sẽ được đặt trên lớp đã chọn. Sau khi vẽ một nghiên cứu, bạn có thể gán nó cho bất kỳ lớp nào khác thông qua hộp thuộc tính đối tượng.

Menu ngữ cảnh

Nếu bạn nhấp chuột phải vào tên lớp, bạn sẽ thấy menu ngữ cảnh chứa các lựa chọn sau:

- Thêm lớp
- Loại bỏ lớp
- Hiển thị tất cả các lớp
- Ẩn tất cả các lớp
- Chuyển đổi
- Mở khoá lớp được tích hợp
- Khoá lớp được tích hợp
- Thuộc tính

Thêm / Xóa lớp là ko cần giải thích gì thêm. Xin lưu ý rằng bạn không thể xóa 5 lớp đầu tiên (lớp tích hợp).

Hiển thị tất cả / Ẩn tất cả - hiển thị và ẩn tất cả các lớp KHÔNG BỊ KHÓA.
Chuyển đổi - chuyển đổi độ hiển thị của tất cả các lớp KHÔNG BỊ KHÓA.

Mở khóa / Khóa các lớp tích hợp - cho phép bạn mở khóa hoặc khóa 5 lớp đầu tiên (lớp tích hợp). Khi một lớp được mở khóa, độ hiển thị của nó sẽ không thay đổi tự động khi khoảng thời gian thay đổi, và bạn có thể hiển thị/ẩn nó một cách thủ công.

Thuộc tính (Properties) - công cụ này khởi chạy hộp thuộc tính cho phép bạn đổi tên lớp và quyết định xem lớp đã cho có nên hoặc không nên bị khóa vào khoảng thời gian hiển thị.

Nếu bạn đánh dấu vào ô "Khóa tầm nhìn vào khoảng thời gian" (Lock visibility to interval), lớp sẽ tự động hiện/ẩn tùy thuộc vào khoảng thời gian hiện đang được hiển thị. Bạn có thể định rõ tầm nhìn cho **mỗi** lớp bằng cách sử dụng combo "Khoảng thời gian" (Interval) và các nút "Tự động hiển thị/ẩn" (Show/hide automatically). Lưu ý rằng có một cài đặt tầm nhìn riêng biệt cho MỖI khoảng thời gian. Hộp thuộc tính của lớp LUÔN LUÔN hiển thị khoảng thời gian "hàng tháng" (monthly interval) khi khởi động, nhưng đây chỉ là điều kiện khởi động, bạn chỉ cần chuyển sang khoảng thời gian cụ thể và sửa đổi tầm nhìn. Để cài đặt hoàn toàn lớp bị khóa, bạn phải đặt tầm nhìn cho **mỗi lớp được liệt kê** trong combo-box "Khoảng thời gian" (Interval). Chỉ cần chọn khoản thời gian và quyết định nếu lớp nên được hiển thị hoặc ẩn đi trong khoản thời gian này, sau đó chọn khoản thời gian kế tiếp và lại quyết định hiển thị hay ẩn, chọn tiếp và tiếp tục như vậy... cho đến khi bạn định rõ tầm nhìn cho tất cả các khoảng thời gian.

Sử dụng cửa sổ Nghiên cứu Web

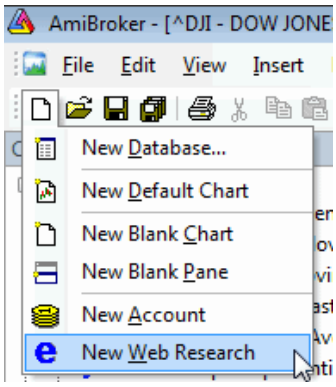
Cửa sổ Nghiên cứu Web cho phép bạn xem tin tức trực tuyến, nghiên cứu, hồ sơ, thống kê và tất cả các loại thông tin liên quan đến ký hiệu đang được chọn hiện tại có sẵn trên Internet (World Wide Web). Việc sử dụng Nghiên cứu Web thay vì trình duyệt web thông thường có lợi thế về tốc độ, vì bạn không cần phải gõ địa chỉ phức tạp/dài (URL) mỗi lần cần thông tin.

Cửa sổ Nghiên cứu Web được giới thiệu trong phiên bản 4.90, thay thế và nâng cao cửa sổ Hồ sơ trước đó. Giờ đây, nó cho phép số lượng trang nghiên cứu web (hồ sơ) do người dùng định nghĩa không giới hạn, duyệt bất kỳ trang web nào (chỉ cần gõ URL), duyệt theo tab, mở nhiều trang cùng một lúc và đồng bộ hóa tự động có chọn lọc.

Nghiên cứu Web sử dụng công cụ Internet Explorer, vì vậy bạn có thể yên tâm rằng các trang được hiển thị với chất lượng tương tự như từ trình duyệt độc lập.

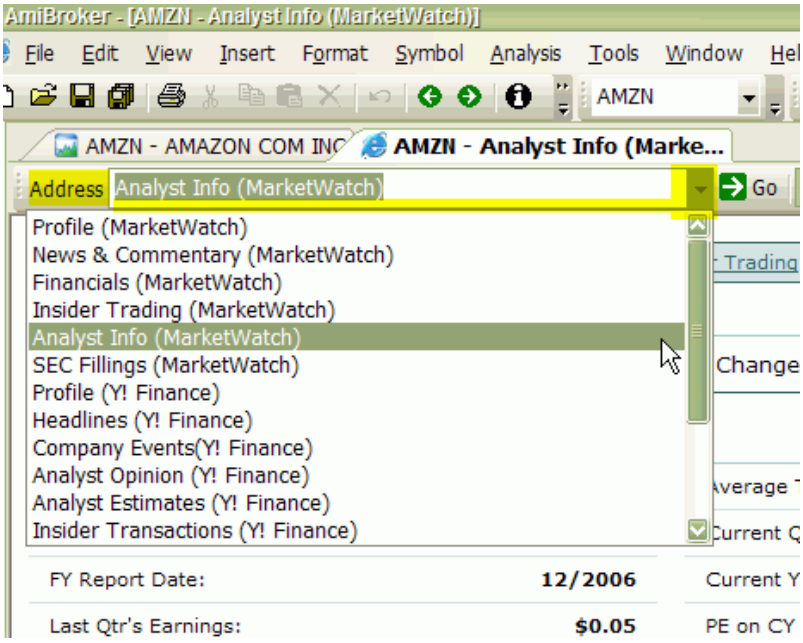
MỞ CỬA SỔ NGHIÊN CỨU WEB MỚI

Sử dụng menu **File->New->Web Research** để tạo cửa sổ nghiên cứu web mới.

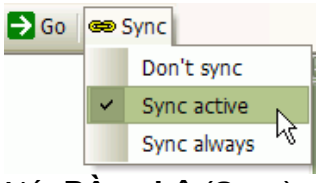


CHỌN TRANG NGHIÊN CỨU WEB ĐƯỢC ĐỊNH NGHĨA TRƯỚC

Để hiển thị bất kỳ trang nghiên cứu web nào đã được định nghĩa trước, chỉ cần nhấp vào mũi tên thả xuống trong hộp kết hợp Địa chỉ và chọn một mục từ danh sách. Khi bạn làm điều này, trang web liên quan đến ký hiệu đang được chọn sẽ tự động được hiển thị.



Bây giờ bạn có thể chỉ định xem trang hiển thị có nên thay đổi tự động hay không nếu bạn chọn ký hiệu khác.

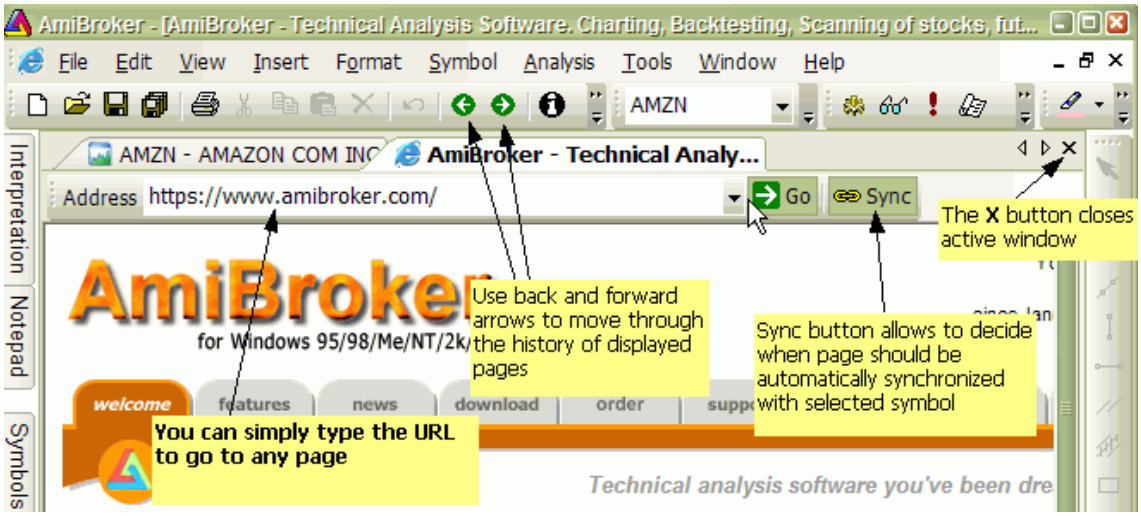


Nút **Đồng bộ** (Sync) cho phép bạn quyết định khi nào trang nên được tự động đồng bộ hóa với ký hiệu đang được chọn hiện tại.

- **Không đồng bộ** - có nghĩa là trang sẽ không được đồng bộ hóa với ký hiệu đang được chọn.
- **Đồng bộ khi hoạt động** - có nghĩa là trang chỉ nên được đồng bộ hóa KHI nó đang hoạt động hoặc trở thành hoạt động (bằng cách người dùng nhấp vào tab đó) - đây là cài đặt được khuyến nghị cho các hồ sơ nghiên cứu web vì nó tiết kiệm băng thông và tài nguyên (các trang không hoạt động sẽ không được đồng bộ hóa và không tiêu tốn băng thông).
- **Luôn đồng bộ** - có nghĩa là trang sẽ luôn được đồng bộ hóa với ký hiệu đang được chọn, bất kể nó có đang hoạt động hay không.

DẪN ĐƯỜNG

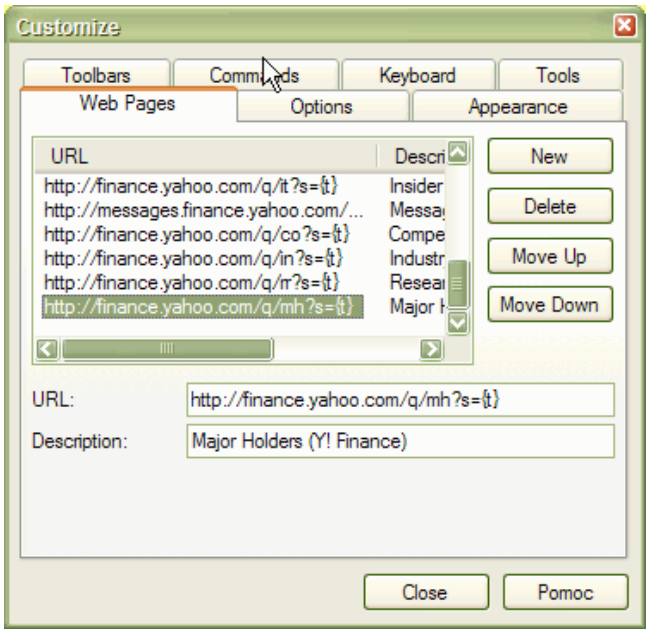
Cửa sổ Nghiên cứu Web hoạt động theo cách rất giống với trình duyệt độc lập. Để hiển thị bất kỳ trang web nào, chỉ cần gõ địa chỉ URL vào trường "Địa chỉ" và nhấn phím ENTER (RETURN). Để điều hướng trở lại và tới trong lịch sử, sử dụng các nút <- và ->.



Để đóng trang đang hiển thị, sử dụng nút đóng cửa sổ **X** thông thường như hình minh họa ở trên.

ĐỊNH NGHĨA CÁC ĐỊA ĐIỂM NGHIÊN CỨU WEB CỦA RIÊNG BẠN

Ngoài các trang nghiên cứu web đã được định nghĩa trước, bạn có thể định nghĩa bất kỳ số lượng địa điểm nào của riêng bạn. Để làm điều này, sử dụng menu **Tools->Customize**, thẻ **Web Pages.**



Để thêm địa điểm mới, nhấn nút **New**, sau đó gõ mẫu URL vào trường URL và mô tả trang web vào trường Description.

Mẫu URL là địa chỉ web có các phần phụ thuộc vào ký hiệu được chọn. Mẫu URL này sẽ được AmiBroker phân tích để tạo ra URL thực tế cho trang web. Ví dụ, để xem trang hồ sơ của Yahoo, bạn có thể sử dụng mẫu URL sau:
http://biz.yahoo.com/p/**{t0}/{t}**.html.

Các ký hiệu được gói trong ngoặc **{}** định nghĩa các trường được đánh giá tại thời gian thực thi. Ký hiệu **{t0}** được đánh giá thành ký tự đầu tiên của tên mã chứng khoán và **{t}** được đánh giá thành tên mã chứng khoán đầy đủ. Vì vậy, nếu AAPL được chọn, AmiBroker sẽ tạo URL sau từ mẫu trên:
http://biz.yahoo.com/p/a/aapl.html

Sau đó AmiBroker sử dụng trình duyệt web tích hợp (cửa sổ Web Research) để hiển thị nội dung của trang.

Hệ mã hóa trường đặc biệt

Như minh họa trong ví dụ trên, URL mẫu có thể chứa các trường đặc biệt, được thay thế thời gian chạy bằng giá trị tương ứng với biểu tượng đang được chọn. Định dạng của trường đặc biệt là {x} nơi x mô tả loại trường. Hiện tại, có ba loại trường được phép: ký hiệu chứng khoán gốc {t}, ký hiệu chứng khoán viết thường **{s}**, ký hiệu chứng khoán viết HOA **{S}**, biệt danh **{a}**, web id **{i}**. Bạn có thể chỉ định những trường này ở bất cứ đâu trong URL và AmiBroker sẽ thay thế chúng bằng các giá trị thích hợp được nhập trong cửa sổ Thông tin. Bạn cũng có thể tham chiếu đến một ký tự đơn của mã chứng khoán, biệt danh hoặc id web. Điều này hữu ích khi một trang web nhất định sử dụng các ký tự đầu tiên của, ví dụ, mã chứng khoán để nhóm các tệp html (trang Yahoo Finance làm điều đó), vì vậy bạn có các tệp cho các mã chứng khoán bắt đầu bằng 'a' được lưu trữ trong thư mục con 'a'. Để tham chiếu đến một ký tự đơn của trường, sử dụng kiểu định dạng thứ hai {xn} nơi x là loại trường được mô tả ở trên và n là chỉ số dựa trên không của ký tự. Vì vậy, **{a0}** sẽ đánh giá thành ký tự đầu tiên của chuỗi biệt danh. Để lấy hai ký tự đầu tiên của một mã chứng khoán, bạn chỉ cần viết **{t0}{t1}**. Lưu ý về trường Web ID: trường mới này trong cửa sổ Thông tin được thêm vào để xử lý các tình huống khi các trang web không sử dụng tên mã chứng khoán để lưu trữ các tệp hồ sơ. Tôi đã tìm thấy một số trang sử dụng hệ thống đánh số riêng, vì vậy họ gán một số duy nhất cho mỗi ký hiệu. AmiBroker cho phép bạn sử dụng mã không chuẩn này để xem hồ sơ. Tất cả những gì bạn cần làm là nhập đúng ID vào trường Web ID và sử dụng mẫu URL phù hợp với từ khóa **{i}**.

Trang được lưu trữ cục bộ

Bạn có thể muốn lưu trữ tất cả các trang trên ổ đĩa cứng của mình. Điều này có lợi thế là các hồ sơ có thể truy cập ngay lập tức nhưng chúng có thể chiếm một lượng lưu trữ đáng kể và bạn sẽ cần cập nhật chúng từ thời gian này sang thời gian khác. Để truy cập các tệp được lưu trữ cục bộ, hãy sử dụng URL mẫu sau (ví dụ, C: chỉ đến ổ đĩa): file://C:\the_folder_with_profile_files{t}.html. Bạn không bị giới hạn chỉ sử dụng các tệp HTML, mà có thể sử dụng tệp TXT đơn giản thay thế. Sau đó, tạo (hoặc tải về) các tệp .html (hoặc .txt) cho mỗi ký hiệu trong danh mục đầu tư. Những tệp này nên tuân theo quy tắc đặt tên sau: <ticker>.html. Ví dụ, đối với APPLE (ký hiệu AAPL), hồ sơ nên có tên là AAPL.html (hoặc AAPL.txt).

Hồ sơ dựa trên web

Nếu bạn muốn hiển thị hồ sơ từ các trang web từ xa, bạn sẽ cần tìm hiểu cách truy cập chúng (URL đến trang web) và cách dữ liệu cho các ký hiệu khác nhau được truy

cập. Tôi sẽ mô tả vấn đề này dựa trên ví dụ của trang Sharenet (www.sharenet.co.za) cung cấp dữ liệu cho các công ty niêm yết trên Sàn giao dịch Chứng khoán Johannesburg. Sharenet cung cấp thông tin công ty có thể truy cập tại địa chỉ (URL) sau:
http://www.sharenet.co.za/free/free_company_na.phtml?code=**JSECODE**&scheme=default

Vấn đề là cơ sở dữ liệu do Sharenet cung cấp sử dụng tên ký hiệu dài và **JSECODE** là mã ký hiệu ngắn. Ví dụ, đối với công ty "Accord Technologies", ký hiệu trong cơ sở dữ liệu Sharenet là ACCORD nhưng mã là ACR. Để giải quyết vấn đề này, chúng ta sẽ cần sử dụng trường **Web ID** trong cửa sổ Thông tin của ký hiệu. Nếu bạn có cơ sở dữ liệu Sharenet, chỉ cần chọn ACCORD từ danh sách ký hiệu, mở cửa sổ *Symbol->Information* và nhập ACR vào ô chỉnh sửa **Web ID**, sau đó nhấp OK. Tiếp theo, nhập mẫu URL sau vào ô chỉnh sửa **URL**:

http://www.sharenet.co.za/free/free_company_na.phtml?code={i}&scheme=default

Để chắc chắn 100%, vui lòng chọn văn bản trên bằng chuột. Sau đó, sao chép nó vào clipboard (Chọn -> Sao chép, hoặc CTRL-C). Chuyển sang AmiBroker và nhấp vào hộp chỉnh sửa URL cấu hình. Xóa tất cả mọi thứ khỏi đó và nhấn CTRL-V (điều này sẽ dán văn bản). Nhập "Sharenet" vào trường **Description**.

Xin lưu ý rằng chúng tôi đã sử dụng trường đặc biệt **{i}** trong mẫu sẽ được thay thế bởi AmiBroker với văn bản nhập vào trường Web ID của cửa sổ thông tin biểu tượng. Bây giờ vui lòng chọn *File->New->Web Research* và chọn Sharenet từ hộp combo Địa chỉ. Bạn nên thấy hồ sơ ACCORD.

Bạn cũng có thể xóa bất kỳ mục nào bằng cách chọn nó từ danh sách và nhấn nút **Delete**. Bạn có thể thay đổi thứ tự các trang xuất hiện trong danh sách địa chỉ nghiên cứu trang web bằng các nút **Move Up** và **Move Down** (chọn mục trước rồi sau đó sử dụng nút).

Dữ liệu cấu hình được lưu trữ trong tệp webpages.cfg đơn giản chứa bất kỳ số lượng mẫu URL ở dạng:
URLTemplate|Description

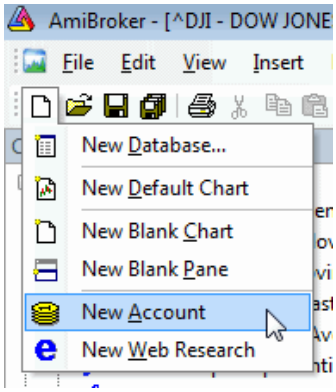
(mỗi mục trên một dòng riêng biệt)

Sử dụng Account Manager

Account Manager là công cụ giúp bạn quản lý các giao dịch và hiệu suất của mình. Bạn có thể nhập các giao dịch đã thực hiện, nạp/rút tiền, kiểm tra thống kê và hiệu suất làm việc trong quá khứ. Tất cả giao dịch đều được ghi lại để bạn không bao giờ quên những gì đã xảy ra trong quá khứ. Account Manager cho phép bạn theo dõi số lượng tài khoản không giới hạn.

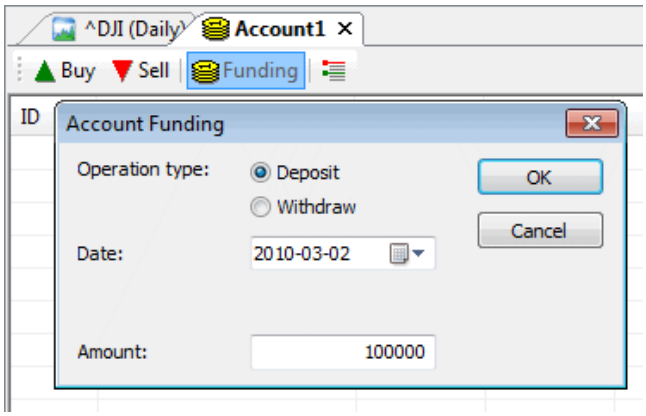
Account Manager mới thay thế và cải tiến chức năng được cung cấp bởi portfolio manager trong các phiên bản trước 4.90.

Tạo Tài Khoản Mới
Sử dụng menu File -> New -> Account để tạo tài khoản mới



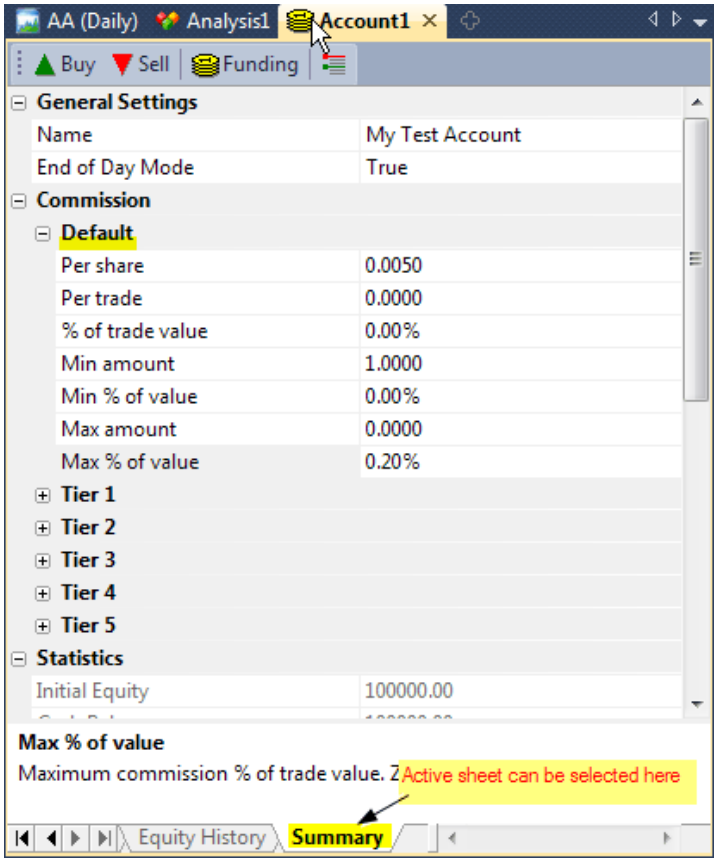
Nạp tiền vào Tài Khoản

Trước khi bạn thực hiện bất kỳ giao dịch nào, bạn cần nạp tiền vào tài khoản. Để làm điều này, nhấn nút "FUNDING" trên thanh công cụ của account manager, sau đó chọn "Deposit" là loại hoạt động, nhập "DATE" khi bạn đã nạp tiền vào tài khoản và nhập số tiền.
Lưu ý rằng ngày nạp tiền phải ĐƯỢC THỰC HIỆN TRƯỚC bất kỳ giao dịch nào, vì account manager sẽ không cho phép bạn giao dịch trước ngày nạp tiền. Khoản tiền gốc sẽ được hiển thị dưới dạng "initial equity" trong tab tóm tắt.



CÀI ĐẶT

Là ý tưởng tốt nếu bạn đi tới "Summary tab" và thiết lập hoa hồng và chế độ giao dịch. Nếu tài khoản này được sử dụng cho giao dịch cuối ngày, bạn nên đặt "EOD Mode" thành YES, ngược lại (nếu bạn giao dịch intraday) bạn nên đặt "EOD Mode" là NO. Tùy thuộc vào cài đặt này, hộp thoại Mua/Bán sẽ cho phép bạn nhập ngày và giờ của giao dịch hoặc chỉ nhập ngày.



Bảng hoa hồng cho phép bạn nhập cả hoa hồng theo cổ phiếu (theo hợp đồng) và hoa hồng được tính theo phần trăm giá trị giao dịch. Hoặc một sự kết hợp của cả hai. Bạn cũng có thể thiết lập mức tối thiểu và tối đa được biểu thị bằng số tiền đô la và/hoặc phần trăm giá trị giao dịch. Ví dụ, nếu nhà môi giới của bạn sử dụng hoa hồng 0,01 đô la (một cent) cho mỗi cổ phiếu, bạn sẽ sử dụng PerShare = 0,01 và %OfTradeValue = 0. Nếu nhà môi giới của bạn sử dụng 0,2% giá trị giao dịch, bạn sẽ sử dụng PerShare = 0 và %OfTradeValue = 0,2;

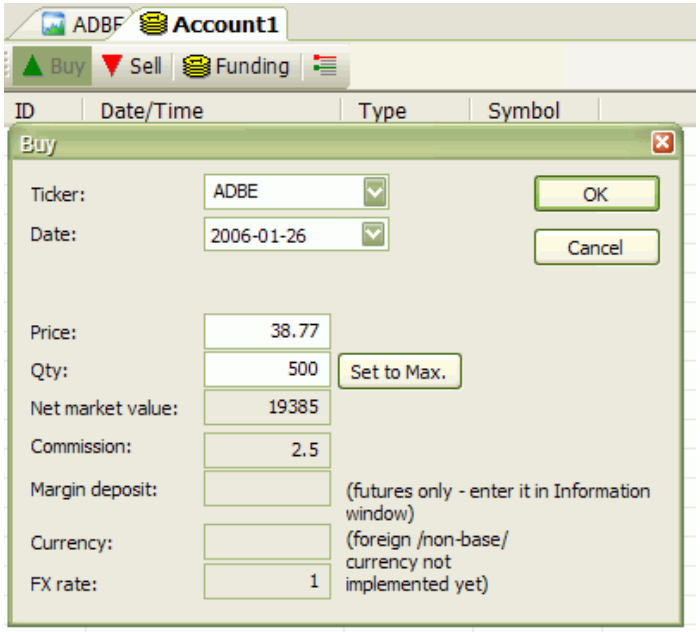
Ví dụ thực tế: Hoa hồng mặc định của Interactive Brokers cho cổ phiếu Mỹ là 0,005 đô la mỗi cổ phiếu, nhưng không thấp hơn 1 đô la và không cao hơn 0,2% giá trị giao dịch. Các cài đặt phù hợp cho lịch trình này được hiển thị trong hình chụp màn hình trên.

Bảng hoa hồng hoạt động như sau: trước tiên, tổng hoa hồng theo cổ phiếu và phần trăm giá trị giao dịch sẽ được tính toán. Sau đó, kết quả sẽ được kiểm tra so với các giới hạn tối thiểu và tối đa; nếu giá trị tính toán vượt quá giới hạn, hoa hồng sẽ được đặt bằng giá trị của giới hạn đó, ngược lại, giá trị tính toán sẽ được sử dụng mà không thay đổi.

Trang tóm tắt cũng chứa một số thống kê cơ bản.

NHẬP GIAO DỊCH

Khi bạn đã nạp tiền vào tài khoản, bạn có thể nhập giao dịch. Để mua (nhập vị thế dài hoặc đóng vị thế ngắn), hãy nhấp vào nút "BUY".



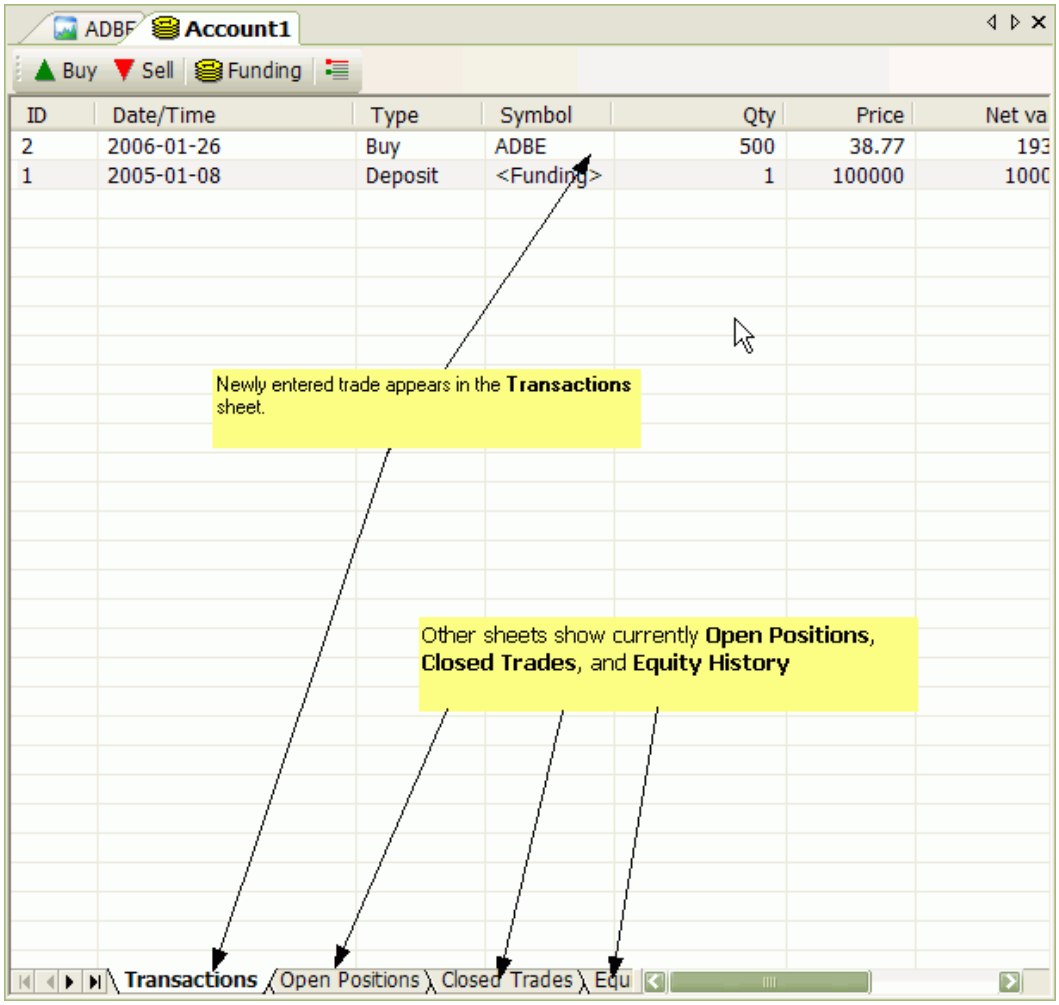
Sau đó, trong hộp thoại Mua, bạn cần chọn ký hiệu và ngày/giờ giao dịch. Khi các thông tin này được nhập, AmiBroker sẽ hiển thị giá của ký hiệu đó tại ngày/giờ đã chọn (hoặc ngày/giờ trước đó nếu không tìm thấy kết quả chính xác). Nó cũng sẽ tính toán số lượng tối đa có thể dựa trên giá và số tiền có sẵn.

Bạn có thể thay đổi giá và số lượng một cách thủ công.

Tất cả các giá trị khác (giá trị thị trường ròng, hoa hồng, tiền đặt cọc thị trường, tiền tệ, tỷ giá hối đoái) đều được tính toán hoặc truy xuất tự động từ trang Symbol->Information. Khi các giá trị đã chính xác, hãy nhấp OK để xác nhận giao dịch. Nếu bạn mắc lỗi, bạn có thể nhấn UNDO (Edit->Undo) để hoàn tác giao dịch gần nhất.

Quy trình tương tự áp dụng cho việc bán (nhập vị thế ngắn hoặc đóng vị thế dài) với ngoại lệ là bạn nên nhấn nút "SELL" thay vì.

Tất cả các giao dịch bạn đã thực hiện được liệt kê trong bảng "Transactions". Tất cả các vị thế mở được liệt kê trong bảng "Open Positions". Nếu bạn nhập giao dịch cho một ký hiệu đã có vị thế mở, AmiBroker sẽ điều chỉnh "vị thế mở" tương ứng (thực hiện việc mở rộng hoặc thu hẹp). Khi vị thế mở được đóng, nó sẽ bị xóa khỏi danh sách "vị thế mở" và chuyển đến bảng "Closed trades".



Sau mỗi giao dịch, bảng "Equity history" sẽ được cập nhật với giá trị vốn hiện tại của tài khoản và bảng "Summary" cũng sẽ được cập nhật với các thống kê cơ bản về giao dịch mở/long/short (còn nhiều thống kê khác sẽ được cập nhật sau).

QUAN TRỌNG

Bạn cần nhớ rằng bạn phải nhập tất cả các giao dịch theo thứ tự thời gian (cũ nhất trước, mới nhất sau), vì trình quản lý tài khoản sẽ không cho phép bạn thêm giao dịch không theo thứ tự. Nếu bạn mắc lỗi, có một cấp độ hoàn tác mà bạn có thể sử dụng để quay lại trạng thái trước giao dịch cuối cùng. Nếu bạn mắc nhiều lỗi hơn, lựa chọn duy nhất là đóng tài khoản mà không lưu và mở lại tệp gốc.

LƯU DỮ LIỆU TÀI KHOẢN CỦA BẠN

Để lưu các chỉnh sửa đã thực hiện cho tài khoản, sử dụng File->Save (hoặc File->Save As để lưu dưới tên mới). Lưu ý rằng **các tệp tài khoản HIỆN TẠI KHÔNG được mã hóa**, và khá dễ dàng để đọc tệp đối với bất kỳ ai có quyền truy cập. Vì vậy, hãy đảm bảo không để tệp của bạn trên máy tính công cộng. Việc bảo vệ bằng mật khẩu/mã hóa đã được lên kế hoạch nhưng CHƯA ĐƯỢC THỰC HIỆN.

MỞ TÀI KHOẢN ĐÃ TẠO TRƯỚC ĐÓ

Để mở tệp tài khoản, vào File->Open, trong hộp thoại File, chọn "Account (*.acx)" từ hộp kết hợp "Files of type", và chọn tệp tài khoản mà bạn muốn tải.

ĐA TÀI KHOẢN

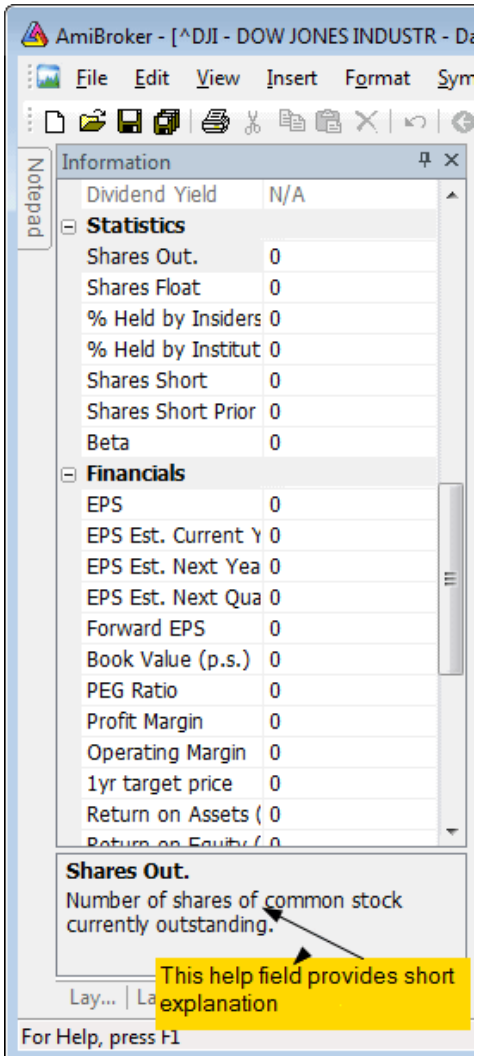
Bạn có thể tạo/mở nhiều tài khoản cùng lúc (chỉ cần sử dụng File->New->Account, File->Open nhiều lần).

Sử dụng Dữ liệu Cơ bản

AmiBroker 4.90 thêm khả năng sử dụng 32 mục dữ liệu cơ bản. Dữ liệu cơ bản có thể được tải tự động cho tất cả các cổ phiếu của Mỹ miễn phí bằng cách sử dụng AmiQuote. **Cửa sổ Thông tin** mới cho phép bạn xem các mục này, trong khi hàm AFL mới: **GetFnData** cho phép truy cập cơ sở vững chắc theo cách lập trình.

Cửa sổ Thông tin

Để hiển thị dữ liệu cơ bản trong cửa sổ Thông tin, vui lòng sử dụng menu **Symbol->Information.** Điều này sẽ mở cửa sổ Thông tin với nhiều trường dữ liệu cơ bản như hình dưới đây (nếu bạn tạo cơ sở dữ liệu mới, có lẽ ban đầu nó sẽ không có dữ liệu này và bạn cần tải chúng về)



Tải Miễn phí Dữ liệu Cơ bản từ Yahoo

Phiên bản mới của AmiQuote giờ đây cho phép tải dữ liệu cơ bản miễn phí từ trang web Yahoo Finance. Điều này được thực hiện bằng cách sử dụng 2 trang Yahoo khác nhau:

1. **Yahoo Fundamental - Nguồn dữ liệu cơ bản** (dữ liệu cơ bản cơ bản miễn phí, 200 ký hiệu trong một yêu cầu). Dữ liệu được lấy từ URL sau: http://finance.yahoo.com/q?s={Ticker} (Liên kết tải dữ liệu)

   Trang này cung cấp các dữ liệu sau:
   EPS (ttm)
   EPS Ước lượng cho Năm Hiện Tại
   EPS Ước lượng cho Năm Tới
   EPS Ước lượng cho Quý Tới
   PEG Ratio
   Book Value
   EBITDA
   Sales Revenue
   Ngày Trả Cổ Tức
   Ngày Cổ Tức Đặc Biệt
   Cổ Tức Trên Mỗi Cổ Phiếu
   Mục Tiêu Giá Cổ Phiếu 1 năm
   Cổ Phiếu Lưu Hành
   Cổ Phiếu Tổng Cộng

Giải thích về các giá trị: http://help.yahoo.com/help/us/fin/quote/quote-03.html

2. **Yahoo Fundamental - Nguồn dữ liệu bổ sung** (dữ liệu cơ bản mở rộng, 1 biểu tượng trong một yêu cầu, thêm dữ liệu - chỉ có sẵn trong phiên bản đã đăng ký).
   Dữ liệu được truy xuất từ URL sau: http://finance.yahoo.com/q/ks?s={Ticker} (Trang Thống Kê Chủ Chốt)

Trang này cung cấp các dữ liệu sau:

   Forward P/E
   PEG Ratio
   Tỉ lệ Lợi Nhuận
   Tỉ lệ Hoạt Động
   Lợi Tức về Tài Sản
   Lợi Tức về Vốn Chủ Sở Hữu
   Revenue (ttm)
   Sự Tăng Trưởng Doanh Thu Hàng Quý
   Lợi Nhuận Gộp
   EBITDA (Diluted)
   EPS
   Sự Tăng Trưởng Lợi Tức Hàng Quý
   Giá Trị Sổ Sách Trên Mỗi Cổ Phiếu
   Cash Flow Hoạt Động
   Dòng Tiền Tự Do Đòn Bẩy

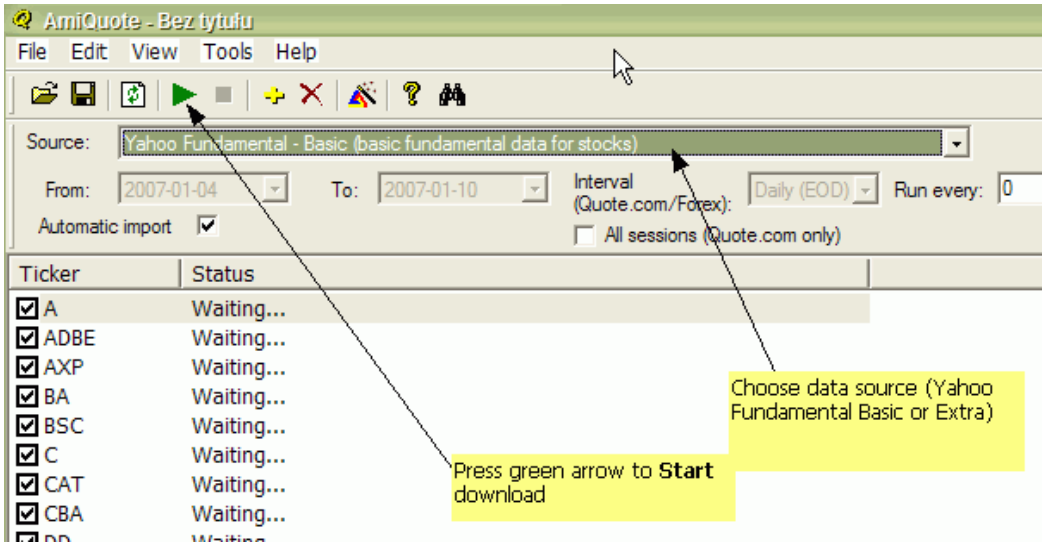
Beta
Cổ Phiếu Tổng Cộng
Lượng Cổ Phiếu Nổi (Float)
Tỉ Lệ % Cổ Phiếu Nắm Giữ Bởi Insider
Tỉ Lệ % Cổ Phiếu Nắm Giữ Bởi Tổ Chức
Cổ Phiếu Bị Short (trước tháng)
Cổ Phiếu Bị Short
Tỷ Lệ Cổ Tức Hàng Năm Trái Phái
Tỷ Lệ Cổ Tức Hàng Năm Theo Dõi
Ngày Cổ Tức
Ngày Trả Cổ Tức Đặc Biệt
Yếu Tố Chia Cổ Phiếu Cuối Cùng
Ngày phân chia cuối cùng

Giải thích về giá trị: http://help.yahoo.com/help/us/fin/research/research-12.html

GHI CHÚ QUAN TRỌNG: Phiên bản chưa đăng ký của AmiQuote cho phép bạn tải xuống dữ liệu cơ bản-ex cho 20 tickers đầu tiên trong danh sách. Để tải xuống dữ liệu cho nhiều biểu tượng, bạn cần đăng ký AmiQuote.

Việc tải dữ liệu rất dễ dàng và trực tiếp:

1. Chạy AmiQuote
2. Trong AmiQuote, chọn **Công cụ->Lấy tickers từ AmiBroker**
3. Chọn **Yahoo Fundamental - Basic** hoặc **Yahoo Fundamental - Extra** từ danh sách chọn nguồn
4. Đảm bảo rằng **hộp nhập tự động** đã được đánh dấu
5. Nhấn mũi tên màu xanh để **bắt đầu tải xuống**



Một khi quá trình tải xuống hoàn tất, bạn sẽ thấy dữ liệu cơ bản được cập nhật trong cửa sổ Thông tin trong AmiBroker.

Truy cập dữ liệu cơ bản từ Công thức (Afl) Level

Để truy cập dữ liệu cơ bản từ AFL level, bạn có thể sử dụng chức năng **GetFnData** mới. Nó có cú pháp đơn giản:
GetFnData("field") nơi "field" là bất kì trường dữ liệu cơ bản nào được hỗ trợ. Vui lòng xem danh sách chi tiết trong **tham chiếu chức năng GetFnData.**

Chức năng trả về số (vô hướng) biểu thị giá trị hiện tại của mục dữ liệu cơ bản. Không có lịch sử của các giá trị (không có mảng nào được trả lại), vì vậy nó hữu ích cho việc quét, khám phá (đối với tình hình hiện tại), bình luận / giải thích về thị trường, nhưng không phù hợp cho việc backtesting. Công thức khai thác ví dụ như sau:

```
AddColumn( Close / GetFnData( "EPS" ) , "Tỷ lệ P/E hiện tại" );
AddColumn( Close / GetFnData( "EPSEstNextYear" ) , "Tỷ lệ P/E dự kiến năm sau" );
Filter = Status("lastbarinrange");
```

Nhập Dữ Liệu Cơ Bản Từ Các Nguồn Khác

AmiBroker cũng cho phép nhập dữ liệu cơ bản sử dụng **trình nhập ASCII** linh hoạt của nó và/hoặc **giao diện OLE,** vì tất cả các trường mới đều được trình bày dưới dạng thuộc tính của đối tượng **Stock**.

Lệnh $FORMAT của trình nhập ASCII giờ đây hỗ trợ các trường thêm vào cho dữ liệu cơ bản sau đây:
DIV_PAY_DATE
EX_DIV_DATE
LAST_SPLIT_DATE
LAST_SPLIT_RATIO
EPS EPS_EST_CUR_YEAR
EPS_EST_NEXT_YEAR
EPS_EST_NEXT_QTR
FORWARD_EPS
PEG_RATIO
BOOK_VALUE (đòi hỏi SHARES_OUT cũng cần được xác định)
BOOK_VALUE_PER_SHARE
EBITDA
PRICE_TO_SALES (đòi hỏi CLOSE cũng cần được xác định)
PRICE_TO_EARNINGS (đòi hỏi CLOSE cũng cần được xác định)
PRICE_TO_BV (đòi hỏi CLOSE cũng cần được xác định)
FORWARD_PE (đòi hỏi CLOSE cũng cần được xác định)
REVENUE
SHARES_SHORT
DIVIDEND
ONE_YEAR_TARGET
MARKET_CAP (đòi hỏi CLOSE cũng cần được xác định - nó được sử dụng để tính shares outstanding)
SHARES_FLOAT
SHARES_OUT
PROFIT_MARGIN
OPERATING_MARGIN

RETURN_ON_ASSETS
RETURN_ON_EQUITY
QTRLY_REVENUE_GROWTH
GROSS_PROFIT
QTRLY_EARNINGS_GROWTH
INSIDER_HOLD_PERCENT
INSTIT_HOLD_PERCENT
SHARES_SHORT_PREV
FORWARD_DIV
OPERATING_CASH_FLOW
FREE_CASH_FLOW
BETA

Lưu ý là nếu bạn chỉ muốn nhập dữ liệu cơ bản với trình nhập ASCII (không kèm theo báo giá) bạn cần sử dụng lệnh **$NOQUOTES 1**. Xem những tệp tin Formats\aqfe.format và Formats\aqfn.format để ví dụ về cách sử dụng - những là những tệp tin thực sự được sử dụng bởi AmiQuote để thực hiện nhập tự động dữ liệu cơ bản tải về từ Yahoo.

Các tên thuộc tính bổ sung của đối tượng 'Stock' giống như những gì được sử dụng bởi hàm '**GetFnData**'. Chi tiết về chúng được liệt kê trong tham chiếu của đối tượng 'OLE' (**OLE objects reference**).

Sử dụng Cửa sổ Phân tích Mới

Giới thiệu

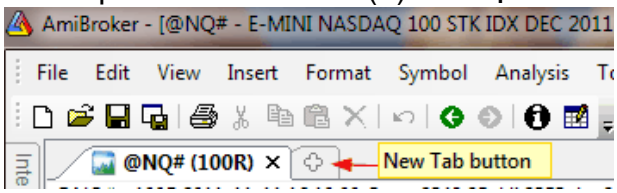
Cửa sổ Phân tích Mới được giới thiệu trong phiên bản 5.50 (lần đầu tiên thực sự trong 5.41.0 BETA) mang lại những cải tiến sau đây so với Phân tích Tự động cũ:

- **Hoạt động đa luồng = tốc độ** - cửa sổ Phân tích Mới sử dụng tất cả các CPU/ lõi có sẵn để thực thi công thức trong nhiều luồng song song, cung cấp tăng tốc đáng kể. Ví dụ, trên Intel i7 4 lõi có thể chạy lên đến 8 luồng, nó có thể chạy nhanh hơn lên đến 8 lần so với cửa sổ Phân tích cũ. Tốc độ tăng chính xác phụ thuộc vào độ phức tạp của công thức (càng phức tạp, tốc độ tăng càng có thể), số lượng dữ liệu được xử lý (truy cập RAM có thể không nhanh như CPU do giới hạn tăng tốc độ có thể).
- **Hoạt động không chặn** - bạn giờ đây có thể xem, cuộn và sắp xếp kết quả phân tích trong khi chúng vẫn đang được tạo ra, cũng như luồng giao diện người dùng không được sử dụng để xử lý phần lớn, biểu đồ và các phần của chương trình điều khiển giao diện đồ họa khác thì đáp ứng hơn nhiều so với phân tích tự động cũ
- **Nhiều trường hợp** - bạn có thể chạy nhiều hơn một thể hiện của Phân tích Mới cùng một lúc, vì vậy bạn có thể chạy nhiều quét/kiểm tra lại/khám phá/tối ưu hóa song song mà không phải chờ đợi một cái hoàn thành
- **Giao diện người dùng mượt mà** - Cửa sổ Phân tích Mới có thể hoạt động như một tài liệu có nhiều tab, có thể được nổi lên, các nút có thể được sắp xếp lại để tăng cường quy trình làm việc. Có nhiều không gian hơn cho danh sách kết quả, thông tin bổ sung về việc thực hiện được cung cấp trên tab "Thông tin" mới. Kết quả tiến bộ cũng hiển thị trong cửa sổ Phân tích Mới để giảm hỗn độn.

Giao diện người dùng

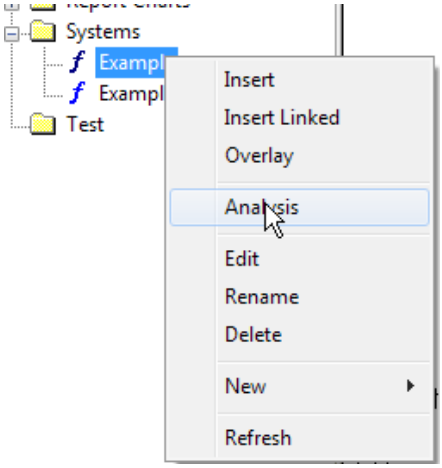
Bạn có thể mở cửa sổ Phân tích Mới theo nhiều cách:

1. Nhấp vào nút Tab Mới (+) và chọn **Phân tích Mới**



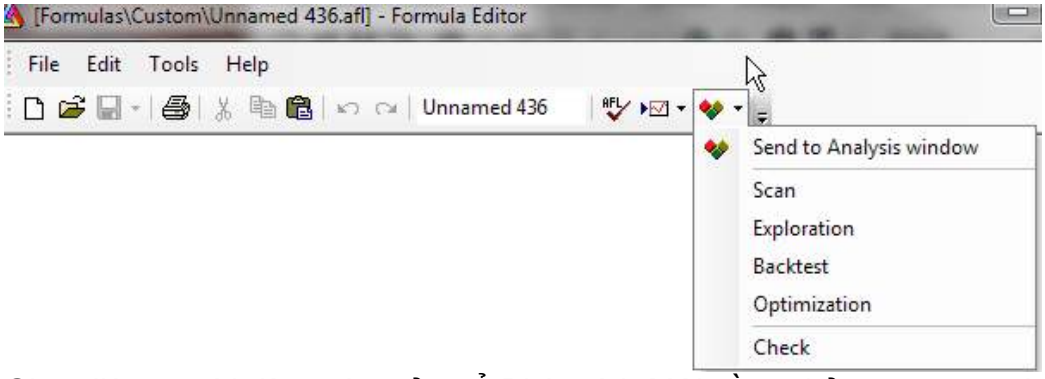
hoặc

2. **File -> Mới -> Menu Phân tích Mới**
hoặc
3. **Phân tích**-> Menu **Phân tích Mới**
4. Nhấp chuột phải vào công thức trong cửa sổ **Biểu đồ** và chọn **Phân tích**

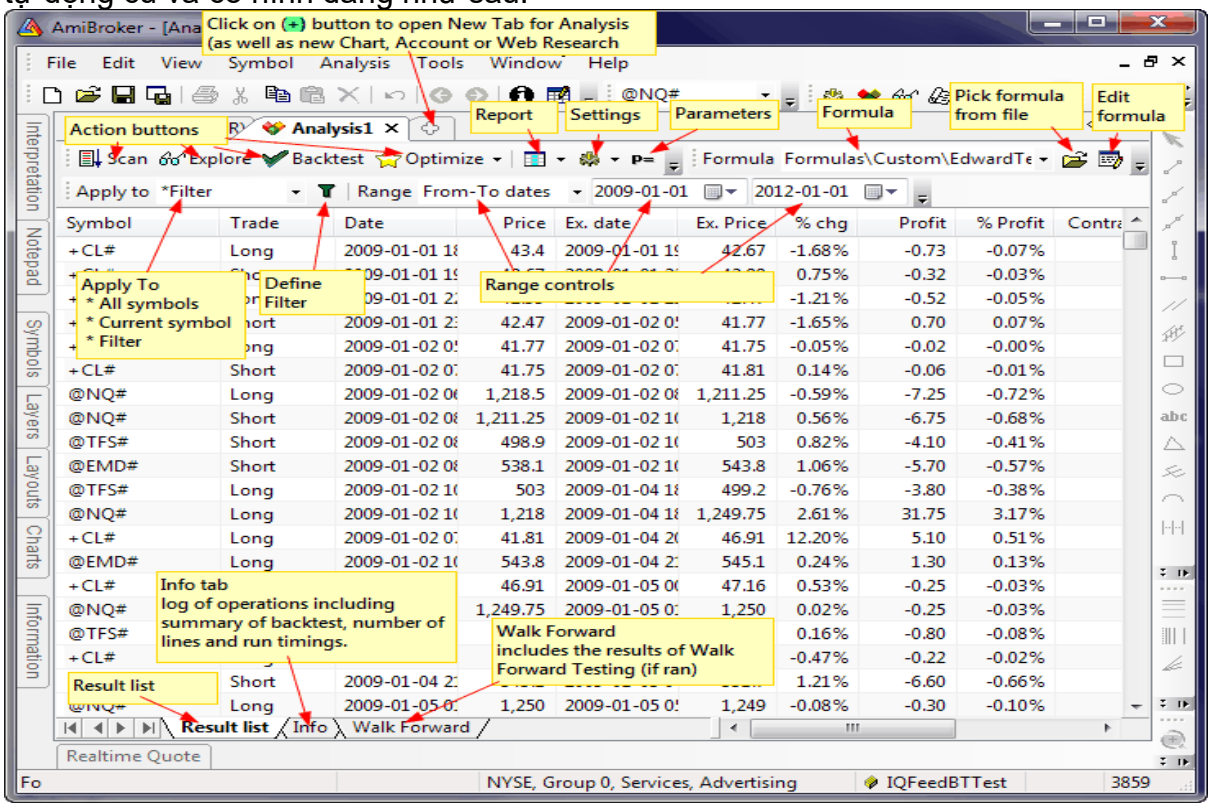


hoặc

5. from the Formula Editor, pressing Send to Analysis button



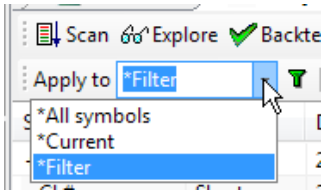
Giao diện người dùng cho cửa sổ Phân tích Mới về cơ bản tương tự như phân tích tự động cũ và có hình dáng như sau:



Các thao tác cơ bản

Chọn biểu tượng để áp dụng phân tích

Nhấp vào mũi tên xổ xuống trong combo **Áp dụng** để chọn chế độ hoạt động: **Tất cả biểu tượng / Biểu tượng** hiện tại / **Bộ lọc.**

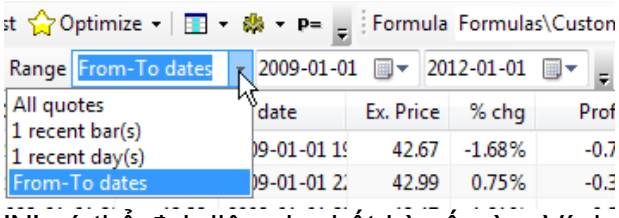


Định nghĩa Bộ lọc

Nếu **Áp dụng** được đặt thành **Bộ lọc**, cửa sổ Phân tích sẽ chạy trên các biểu tượng phù hợp với tiêu chí lọc có thể định nghĩa trong cửa sổ Cài đặt Bộ lọc. Để mở cửa sổ Cài đặt Bộ lọc, nhấn nút **Bộ lọc**.

Định nghĩa khoảng thời gian/ngày giờ

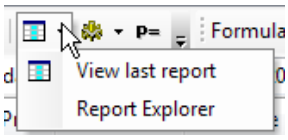
Nhấp vào mũi tên xổ xuống trong combo **Khoảng** để chọn chế độ chọn khoảng: **Tất cả biểu tượng / N thanh gần đây / N ngày gần đây / Từ-ngày đến-ngày.**



'N' có thể đại diện cho bất kỳ số nào. Ví dụ, để định nghĩa khoảng thời gian 15 ngày gần nhất, hãy chọn **1 ngày gần đây trước** và sau đó nhập **15** và nhấn ENTER. Bạn sẽ thấy văn bản tự động cập nhật thành **15 ngày gần đây**. Nhớ rằng bạn không cần nhập toàn bộ, chỉ cần nhập số là đủ.

Xem báo cáo / Chạy Trình khám phá Báo cáo

Để xem báo cáo từ lần kiểm tra lại trước, nhấp vào nút **Báo cáo**. Để chạy Trình khám phá **Báo cáo**, sử dụng menu xổ xuống của nút Báo cáo như được hiển thị dưới đây.

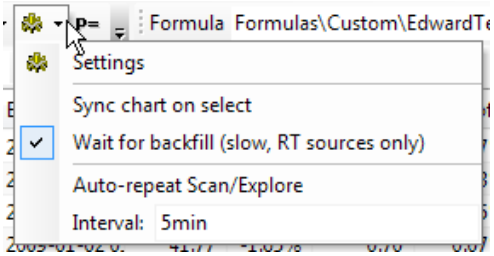


Thay đổi cài đặt / tùy chọn

Để thay đổi cài đặt kiểm tra lại, nhấp vào nút **Cài đặt**. Để bật/tắt các tùy chọn bổ sung như:

- Đồng bộ hóa biểu đồ khi chọn
- Chờ đợi để điền lại
- Tự động lặp lại Quét/Khám phá
- Khoảng thời gian tự động lặp lại

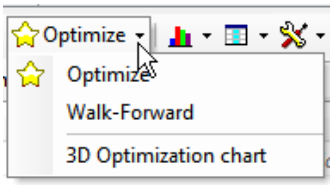
Nhấp vào mũi tên xổ xuống trên nút **Cài đặt** để hiển thị menu như dưới đây.



Khoảng thời gian tự động lặp lại có thể được nhập trong trường **Khoảng**. Lưu ý rằng các số đơn giản (như 5) đại diện cho phút. Để có giây, bạn cần nhập **5sec** hoặc **5s** và nhấn **ENTER**.

Chạy Kiểm tra Đi bộ tiến lên

Nhấp vào mũi tên trên nút Tối ưu hóa để hiển thị menu như dưới đây và chọn **Đi bộ Tiến lên.**



Kết quả của kiểm tra Đi bộ tiến lên sẽ được hiển thị trong tab **Đi bộ Tiến lên** (xem dưới cùng của cửa sổ Phân tích).

Hiển thị biểu đồ tối ưu hóa 3D

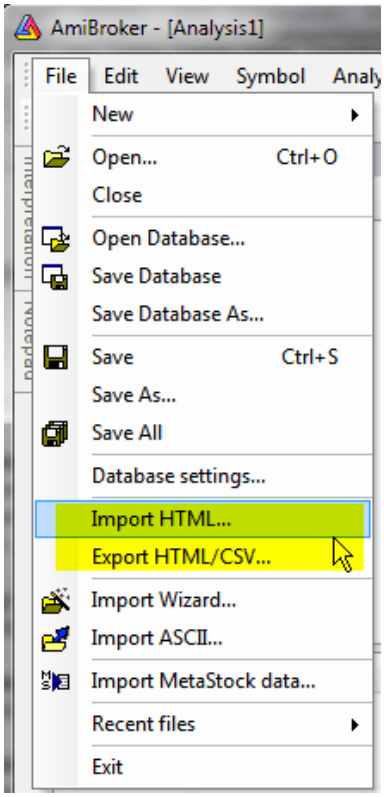
Để hiển thị biểu đồ tối ưu hóa 3D, trước tiên hãy chạy **Tối ưu hóa** với chính xác hai tham số tối ưu hóa và sau đó nhấp vào mũi tên trên nút Tối ưu hóa để hiển thị menu như trên và chọn **Biểu đồ Tối ưu hóa 3D.**

Hiển thị biểu đồ Vốn chủ sở hữu

Biểu đồ Vốn chủ sở hữu (danh mục và cá nhân) có thể được thêm vào các cửa sổ biểu đồ bằng cách sử dụng các tùy chọn Vốn chủ sở hữu **Danh mục / Vốn chủ sở hữu Cá nhân** như đã hiển thị ở trên.

Xuất và nhập danh sách kết quả

Để xuất dữ liệu ra tệp CSV hoặc tệp HTML, sử dụng menu **Tệp -> Xuất HTML/CSV** (từ cửa sổ chính). Để nhập tệp HTML đã xuất trước đó, sử dụng **Tệp -> Nhập HTML**... như hiển thị trong hình dưới đây. Lưu ý rằng các mục menu này chỉ xuất hiện nếu bạn có cửa sổ Phân tích Mới đang hoạt động.



Chạy một chuỗi hành động

Ngoài khả năng xử lý hàng loạt toàn diện, AmiBroker 6.40 mang lại khả năng chạy chuỗi hành động chỉ trong cửa sổ Phân tích.

Tính năng mới này không nhằm thay thế hàng loạt mà để tăng cường lựa chọn có sẵn và cho phép "hacking nhanh" để bạn có thể chạy chuỗi trực tiếp trong cửa sổ Phân tích mà không cần sử dụng hàng loạt toàn diện.

Có một câu lệnh mới là **#pragma sequence**(danh sách các hành động) sẽ định nghĩa chuỗi hành động cần thực hiện và một nút trên thanh công cụ "Chạy chuỗi" để cho phép bạn chạy chuỗi hành động qua một cú nhấp chuột.

Điều này cho phép bạn tạo composites và sau đó chạy khám phá sử dụng các composites đó tất cả qua một cú nhấp chuột vào nút "Chạy chuỗi".

Mã sử dụng **#pragma sequence** nhìn như thế này:

```
// the statement below defines sequence of actions
#pragma sequence(scan,explore)

if( Status("action") == actionScan )
{
AddToComposite( .. );
StaticVarAdd(...);
 _exit(); // quick exit in scan mode
}

Filter = …
AddColumn( ... ); // exploration code that can use composites / static vars created in scan step
```

Sử dụng Giới thiệu về cửa sổ Batch

Giới thiệu

Cửa sổ Batch được giới thiệu trong phiên bản 6.20 (lần đầu tiên thực sự trong 6.17.0 BETA) cho phép người dùng tự động hóa các tác vụ lặp đi lặp lại bằng cách cung cấp xử lý hàng loạt.

Ngay bây giờ, người dùng có thể dễ dàng định nghĩa và tự động chạy các chuỗi các hoạt động Phân tích như quét, thăm dò, thử nghiệm và tối ưu hóa và xuất file. Trước đây, việc tự động hóa như vậy chỉ khả dụng cho những người lập trình thông qua tự động hóa OLE. Bây giờ nó có sẵn cho mọi người thông qua giao diện dễ sử dụng.

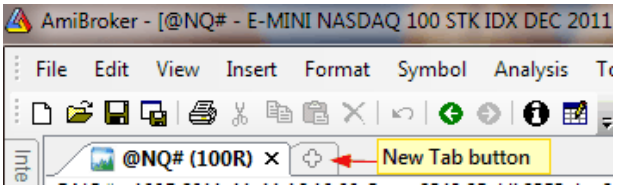
Nói chung, một batch bao gồm bất kỳ số lượng nào của các hoạt động cơ bản sau:

- **Tải một file dự án Phân tích để sử dụng** - để chạy một batch, bạn cần lưu dự án phân tích của mình vào file .APX (Dự án Phân tích) có thể sau đó được tải lên bởi bộ xử lý hàng loạt. Một Dự án Phân tích (.APX) là tự chứa và nó chứa tất cả mọi thứ cần thiết để chạy quét, thăm dò, thử nghiệm hoặc tối ưu hóa của bạn bao gồm công thức, tất cả các tùy chọn & cài đặt, cũng như việc lựa chọn và phạm vi áp dụng.
- (tùy chọn) **Đặt ký hiệu hiện tại** - nếu công thức của bạn dựa trên ký hiệu hiện tại cụ thể, bạn có thể đặt nó từ batch trước khi chạy Phân tích
- **Chạy Quét/Thăm dò/Thử nghiệm/Tối ưu hóa/Di chuyển về phía trước** - một khi dự án được tải, bạn có thể bắt đầu thực hiện bất kỳ hoạt động Phân tích nào
- (tùy chọn) **Xuất kết quả** - bằng cách Xuất ra CSV/HTML, bạn có thể xuất danh sách kết quả. Các báo cáo thử nghiệm cũng được tự động tạo sau thử nghiệm vì vậy bạn không cần xuất chúng một cách rõ ràng để xem chúng sau này trong Báo cáo Khám phá.
- (tùy chọn) **Thông báo bằng âm thanh / lời nói** - Bộ xử lý hàng loạt có khả năng phát bất kỳ file .WAV nào hoặc nói bất kỳ văn bản nào được cung cấp để bạn có thể được thông báo khi batch đã đạt đến một giai đoạn nhất định.
- (tùy chọn) **Viết văn bản vào tệp nhật ký** - (tùy chọn) Tải/Lưu cơ sở dữ liệu
- (tùy chọn) **Thực thi chương trình bên ngoài và chờ hoàn tất** - (tùy chọn) chạy lệnh Data Plugin cụ thể của nh
- (tùy chọn) **Thông báo bằng âm thanh / lời nói** - Bộ xử lý hàng loạt có khả năng phát bất kỳ file .WAV nào hoặc nói bất kỳ văn bản nào được cung cấp để bạn có thể được thông báo khi batch đã đạt đến một giai đoạn nhất định
- (tùy chọn) **Ghi văn bản** vào tệp nhật ký
- (tùy chọn) **Tải/Lưu cơ sở dữ liệu**
- (tùy chọn) **Thực thi** chương trình bên ngoài và chờ hoàn tất
- (tùy chọn) Chạy lệnh Plugin **Dữ liệu đặc thù** của nhà cung cấp
- (tùy chọn) **Nhập dữ liệu** báo giá từ tệp các ASCII

Giao Diện Người Dùng

Bạn có thể mở cửa sổ Phân Tích Mới theo một số cách:

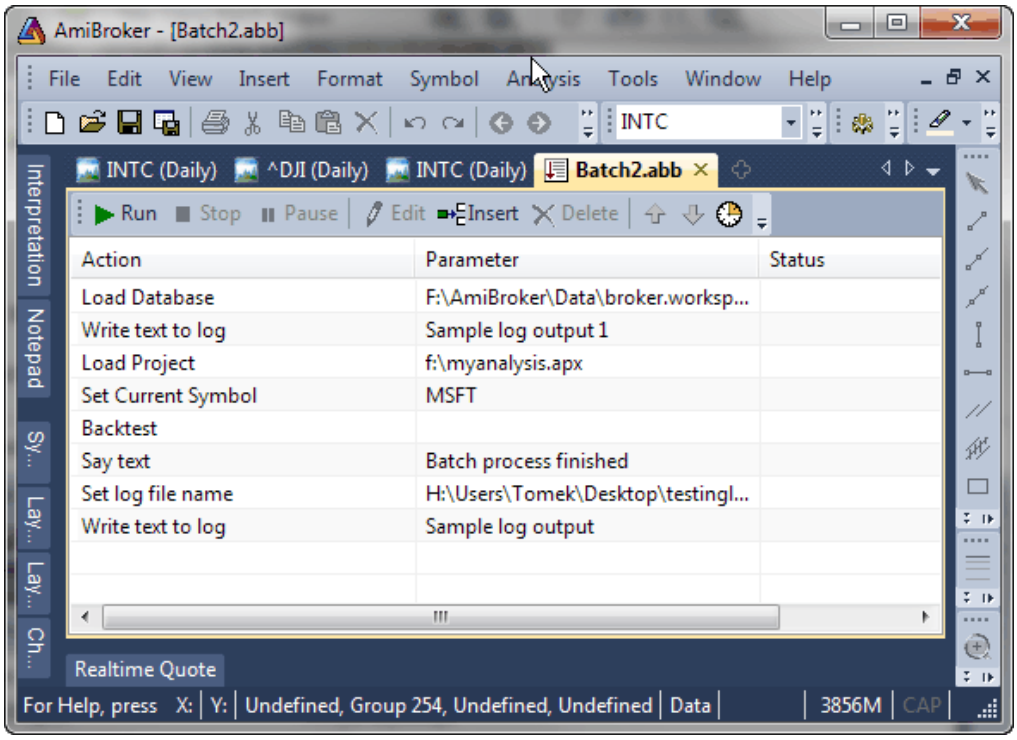
1. Nhấp vào nút Tab Mới (+) và chọn **New Batch**



hoặc

2. Chọn **Tập Tin→ Mới → New Batch.**

Giao diện của cửa sổ Batch rất đơn giản và dễ sử dụng.



Các thao tác cơ bản

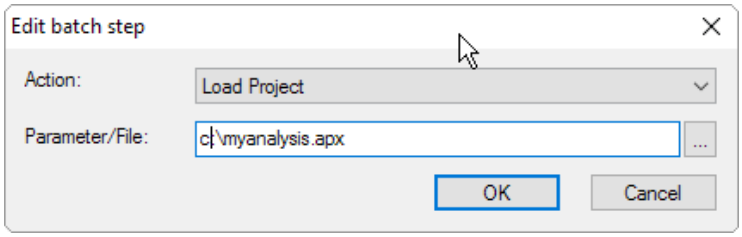
Thêm/chèn bước mới vào batch

Bạn có thể thêm một bước batch mới bằng cách nhấn vào nút **Chèn**. Nó sẽ thêm một bước mới vào cuối nếu không có gì được chọn trước khi nhấn, hoặc nó sẽ chèn một bước mới trước bước đã chọn trước đó. Bạn cũng có thể thêm một bước mới nếu bạn chỉ nhấp đúp vào danh sách sau bước được liệt kê cuối cùng.

Nếu bạn thêm/chèn bước mới, cửa sổ Chỉnh sửa bước batch sẽ xuất hiện (xem bên dưới).

Chỉnh sửa bước batch hiện tại

Để chỉnh sửa mục hiện tại, hãy nhấp đúp vào nó, hoặc sử dụng nút Chỉnh sửa. Cửa sổ Chỉnh sửa bước batch sẽ xuất hiện cho phép bạn định rõ hành động bước batch và các tham số:



Trong cửa sổ này, bạn có thể chọn một hành động để thực hiện bằng cách sử dụng combo-box **Hành động** và nhập **Tham số** hoặc **Tên tệp** để được sử dụng cho hành động đã cho.

Các hành động như **Load Project, Export to File** hoặc **Play sound (.wav)** cần một tên file để được nhập / chọn qua nút (...). Các hành động như **Set Current Symbol** hoặc **Say** cần một tham số văn bản (ký hiệu hoặc văn bản cần phát âm) để nhập. Các hành động khác (như Scan, Backtest, ...) không cần bất kỳ tham số phụ nào. Khi hành động được chọn không yêu cầu tham số, trường nhập thứ hai **(Parameter/File)** sẽ bị ẩn.

Khi đã chọn hành động mong muốn cùng với các tham số, hãy nhấn **OK** để chấp nhận bước này. Nếu bạn muốn thay đổi quyết định, hãy nhấn **Cancel.** Nếu bạn huỷ thao tác **Insert,** mục mới đã thêm sẽ bị loại bỏ khỏi danh sách.

Để xóa một bước batch hiện có, hãy chọn nó bằng cách nhấp vào danh sách và nhấn nút **Delete**.

Di chuyển / sắp xếp lại các bước batch

Bạn có thể thay đổi thứ tự các bước hiện tại bằng cách di chuyển các mục lên và xuống. Để di chuyển mục lên hoặc xuống, hãy nhấp vào nó một lần và sử dụng các nút **mũi tên up** and **down màu xanh** trong thanh công cụ. Bạn cũng có thể sắp xếp lại các mục bằng cách kéo chúng. Để làm điều này, nhấp vào mục một lần (để chọn nhiều, sử dụng phím CTRL hoặc SHIFT), sau đó nhấn và giữ nút chuột trái để kéo chúng đến vị trí mong muốn.

Chạy batch

Khi tất cả các bước batch đã được xác định, bạn có thể chạy chuỗi bằng cách nhấn nút Run (mũi tên màu xanh) trong thanh công cụ. Tất cả các bước sẽ được thực hiện theo thứ tự, bạn sẽ thấy trạng thái của các mục tiếp theo thay đổi thành "Đang thực hiện...", sau đó thành "Hoàn thành" hoặc "Thất bại". Nếu bất kỳ bước nào kết thúc với trạng thái "Thất bại", toàn bộ thực thi batch sẽ bị dừng lại.

Để tạm ngừng thực thi batch, nhấn nút **Pause**. Để hủy thực thi batch đang chạy, hãy nhấn nút **Stop**. (Những nút này chỉ hoạt động khi batch đang chạy, nếu không chúng sẽ bị vô hiệu hóa / màu xám).

Lưu batch để sử dụng sau

Một chuỗi batch có thể được lưu sử dụng **File->Save / File->Save As**... vào tệp .abb (đại diện cho AmiBroker Batch).

Khi batch đã được lưu, sau này nó có thể được tải lại bằng cách sử dụng **File->Open** hoặc sử dụng menu **File->Recent Files.**

Lệnh Thêm Vào

Chạy chương trình bên ngoài từ batch

Một lệnh batch **Execute And Wait** cho phép thực thi chương trình bên ngoài và chờ đợi cho đến khi nó hoàn tất (cho phép, giữa các điều khác, khởi chạy AmiQuote tải và chờ đợi cho đến khi nó hoàn tất

Ví dụ sau đây khởi chạy AmiQuote, tải "YourTickerList.tls", thực hiện tải và đóng AmiQuote:

ExecuteAndWait amiquote\quote.exe YourTickerList.tls /download /close

Chạy lệnh plugin dữ liệu

Một số nhà cung cấp dữ liệu có thể cung cấp các lệnh đặc thù cho dữ liệu có thể được thực hiện từ lô. Ví dụ, PremiumData sẽ cung cấp khả năng bảo trì cơ sở dữ liệu.

Để chạy lệnh plugin dữ liệu, sử dụng lệnh
`DataPluginCmd commandname`

Trong đó *commandname* là lệnh cụ thể của nhà cung cấp dữ liệu. Vui lòng tham khảo nhà cung cấp dữ liệu để biết chi tiết. Nếu bạn chỉ định lệnh không tồn tại, trạng thái của thao tác sẽ là "Thất bại" và việc thực hiện lô sẽ dừng lại.

Nhập dữ liệu báo giá từ tệp ASCII

Lệnh DataImportASCII cho phép tự động hóa việc nhập dữ liệu qua lô. Tham số là tên tệp cần nhập. Trình nhập sử dụng phần mở rộng tệp để xác định định dạng tệp được sử dụng, vì vậy nếu bạn đang nhập `File.aqh`, thì tệp định nghĩa `aqh.format` sẽ được sử dụng tự động. Việc nhập sẽ thất bại nếu tệp định nghĩa tương ứng không tồn tại trong thư mục con "Formats".

Kích hoạt chạy lô từ cấp độ AFL

Có thể khởi động lô một cách lập trình từ AFL. Hàm **ShellExecute** trong AmiBroker 6.20 hỗ trợ lệnh "runbatch".

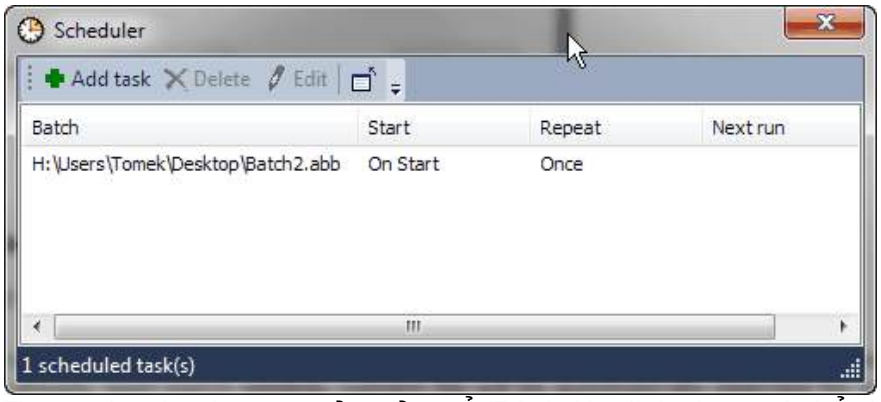
```
if( ParamTrigger("batch", "run me" ) )
{
   ShellExecute("runbatch", "path_to_batch_file.abb", "" );
}
```

Xin lưu ý rằng đây là cuộc gọi không đồng bộ, tức là ShellExecute sẽ bắt đầu một lô và trả về ngay lập tức mặc dù lô có thể (và thường sẽ) vẫn đang chạy, vì vậy đây là loại thao tác "bắn và quên". Hãy cẩn thận vì việc thực thi lặp lại AFL sẽ kích hoạt các lần chạy lô lặp lại. Sau khi chạy lô, các cửa sổ phân tích mà lô mở ra sẽ tự động đóng.

Lập lịch

Các lô có thể được lập lịch để chạy tự động vào một thời điểm nhất định hoặc khi ứng dụng khởi động. Để mở **Trình lập lịch** lô, bạn có thể nhấp vào biểu tượng Trình lập lịch (đồng hồ) trong thanh công cụ Lô hoặc sử dụng mục **Tools->Scheduler** từ menu ứng dụng chính.

Điều này sẽ mở ra cửa sổ Trình lập lịch.

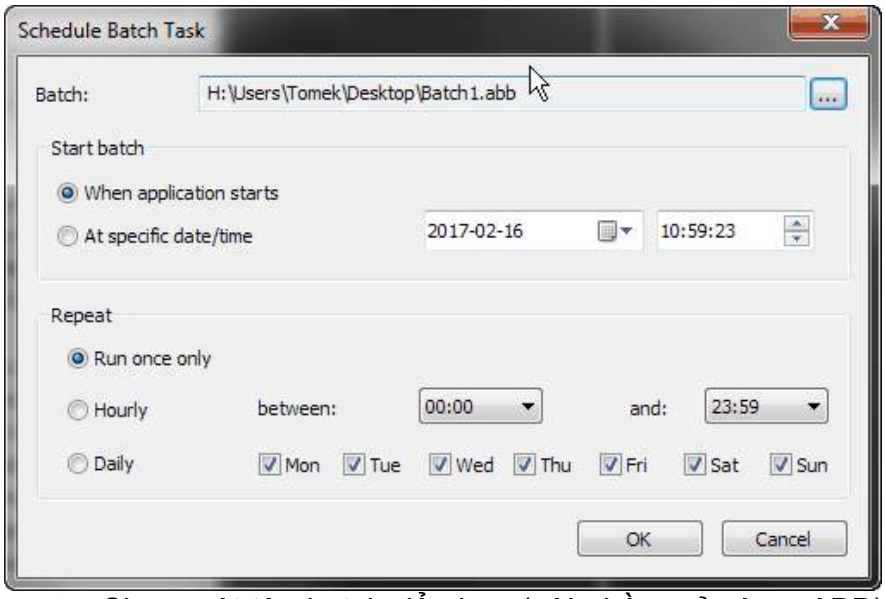


Trong thanh công cụ của cửa sổ Trình lập lịch, bạn có thể thấy các biểu tượng để: Thêm, Xóa và Chỉnh sửa các tác vụ và quyết định xem có nên đóng cửa sổ lô khi hoàn thành hay không.

Danh sách hiển thị đường dẫn đến tệp batch, ngày/giờ **Bắt đầu** và khoảng thời gian **Lặp lại** cũng như ngày/giờ của lần chạy **Tiếp theo**.Nếu batch được lên lịch chạy khi khởi động thì trường Lần chạy tiếp theo sẽ trống.

Để thêm một tác vụ, nhấp vào biểu tượng **Thêm nhiệm vụ**. Điều này sẽ mở hộp thoại sau:

Để định nghĩa lịch chạy lô, vui lòng làm theo các bước sau:



1. Chọn một tệp batch để chạy (với phần mở rộng .ABB). Tệp này có thể được tạo bằng cách đơn giản là lưu một Batch mở thông qua **Tệp->Lưu / Tệp->Lưu thành**. Nếu bạn mở bộ lập lịch từ cửa sổ batch đã mở, tệp hiện tại sẽ tự động được chọn cho bạn.
2. Xác định khi nào chạy batch: ngay khi khởi động ứng dụng hoặc vào ngày/giờ cụ thể.
3. Xác định xem có nên lặp lại các lần chạy tại các khoảng thời gian nhất định hay không. Các lần chạy có thể được thực hiện hàng ngày hoặc hàng giờ vào các ngày trong tuần cụ thể và khoảng thời gian cụ thể

Khi bạn đã hoàn thành, nhấp OK và bạn sẽ thấy nhiệm vụ đã lên lịch trong danh sách cửa sổ Bộ lập lịch.

GHI CHÚ QUAN TRỌNG**:**

- Thời gian kiểm tra là mỗi 10 giây, vì vậy nếu bạn lên lịch cho 12:14:47, nó sẽ thực sự chạy vào lúc 12:14:50.
- Nếu nhiều nhiệm vụ được lên lịch vào cùng một thời gian, chúng sẽ được chạy theo thứ tự xuất hiện trong danh sách Bộ lập lịch
- Các nhiệm vụ đã lên lịch được chạy tuần tự và nhiệm vụ tiếp theo sẽ không bắt đầu nếu nhiệm vụ trước đó chưa kết thúc. Vì lý do này, tránh các batch yêu cầu sự can thiệp của người dùng như việc sử dụng lệnh "PAUSE".
- Các nhiệm vụ đã lên lịch chỉ được bắt đầu khi giao diện người dùng không ở trạng thái modal, tức là giao diện người dùng KHÔNG đang chờ đầu vào từ người dùng (như hộp thoại mở tệp, hoặc bất kỳ hộp thoại đầu vào modal nào khác).
- Khi hộp thoại modal được đóng, các nhiệm vụ bị bỏ lỡ sẽ được chạy.
- Các tác vụ đã lên lịch chỉ được chạy khi AmIbroker đang chạy. AmiBroker không tạo ra quá trình nền / ẩn. Windows đã có quá nhiều quá trình lãng phí đang chạy suốt thời gian. Nếu bạn đóng AmiBroker, nó sẽ không chạy bất kỳ tác vụ nào cho đến khi bạn khởi động lại AmiBroker. Khi khởi động lại, tất cả các tác vụ đã bị bỏ lỡ sẽ được thực hiện một lần, và sau đó chúng sẽ được chạy theo lịch trình bình thường (nếu tùy chọn lặp lại được sử dụng).

Cách Lấy Báo Giá Từ Các Thị Trường Khác Nhau

QUAN TRỌNG: Dữ liệu được cung cấp bởi các công ty bên thứ 3. Thông tin được cung cấp dưới đây chỉ mang tính chất tham khảo. Nó có thể không còn hiện hành, vì các dịch vụ của nhà cung cấp dữ liệu thường xuyên thay đổi. Để biết thông tin chi tiết về các dịch vụ hiện tại của nhà cung cấp dữ liệu bao gồm giá cả, dữ liệu lưu trữ và phạm vi, hãy tham khảo các trang web của họ.

TRANG NÀY ĐƯỢC CẬP NHẬT ĐẾN NGÀY 2 tháng 3, 2023

Dữ Liệu Thời Gian Thực (Chỉ dành cho Phiên Bản Chuyên Nghiệp)

Quốc gia/ Sàn Giao Dịch	**Nguồn Dữ liệu**	**Loại**	**Giá**	**Tải**	**Cập Nhật**	**Bình Luận**
Tất cả thị trường Cổ Phiếu và Hợp Đồng Tương Lai Hoa Kỳ **FOREX** **Các thị trường chính ở Châu Âu**	**eSignal**	Báo giá thời gian thực luồng trực tuyến Tick, 5-, 15-giây 1-, 5-, 15-, 60-phút intraday 10-ngày tick, 60-ngày backfill thanh phút Lịch sử EOD (10 năm)	$61/tháng (TRỄ, không phải luồng trực tuyến) $204/tháng (THỜI GIAN THỰC, luồng trực tuyến) **Thêm thông tin về giá cả**	Tự động	Tự động	**Plug-in RT Dành riêng Tự động**
Cổ phiếu Mỹ, hợp đồng tương lai,lựa chọn, FOREX	DTN IQFeed	**BÁO GIÁ THỜI GIAN THỰC** 500 biểu tượng, tick, 5-giây, 15-giây, 1-phút trở lên, Dữ liệu tick hơn 100 ngày 10 năm	$99/tháng phí cơ bản (thời gian thực) Thêm thông tin về giá cả **Thêm thông tin về giá cả** (ưu đãi đặc biệt dành cho người dùng AmiBroker	Tự động	Tự động	**Plug-in RT Dành riêng Tự động**

		Dữ liệu 1 phút EOD 20+ năm (lưu ý: nguồn cấp không được lọc)	- phí thiết lập $50 BỎ QUA)			
Mỹ, Canada và Châu Âu	**Broker tương tác**	Chỉ 100 ký hiệu tương tác luồng RT, Các thanh dữ liệu 1 giây, 1 phút và các loại thanh cao hơn có sẵn. **Khách hàng của IB có thể truy cập dữ liệu lấp đầy 30 ngày**.	$10 mỗi tháng trong hoa hồng, hoặc miễn phí nếu hoa hồng hàng tháng của bạn lớn hơn 30 đô la.	Tự động	Tự động	**Plug-in RT Dành riêng Tự động**
Các sàn giao dịch khác nhau / các nguồn khác nhau (danh sách chi tiết).	**Quote Tracker**	Cung cấp báo giá trực tuyến theo thời gian thực. Dữ liệu intraday với các khoảng thời gian 1, 5, 15 và 60 phút. Lấp đầy dữ liệu có giới hạn (tối đa 5 ngày, thường là một ngày).	Nhiều tùy chọn (bao gồm cả miễn phí). Thông tin thêm về giá cả.	Tự động	Tự động	**Plug-in RT Dành riêng Tự động**

Sàn Giao dịch Chứng khoán Warsaw	**Statica**	**Dữ liệu intraday trong 30/90 ngày + chế độ kết hợp EOD.**		Tự động	Tự động	**Plug-in RT Dành riêng Tự động**
Nhiều nguồn (bất kỳ nguồn dữ liệu nào có giao diện DDE).	Liên kết DDE	Chỉ cung cấp báo giá trực tuyến, **không có lấp đầy dữ liệu**	Miễn phí	Tự động	Tự động	**Plug-in RT Dành riêng** Tự động

Dữ liệu Kết thúc Ngày (EOD) và Dữ liệu Intraday Bị Trì Hoãn

AmiBroker có thể xử lý hầu như TẤT CẢ các sàn giao dịch trên thế giới nếu chỉ có dữ liệu ASCII đơn giản cho sàn giao dịch đó.

Bảng dưới đây liệt kê một số nguồn dữ liệu. AmiBroker đi kèm với cơ sở dữ liệu mẫu các thành phần DJIA. Bạn có thể cập nhật cơ sở dữ liệu mẫu này (và bất kỳ cơ sở dữ liệu thị trường Mỹ & Canada nào) với trích dẫn mới sử dụng chương trình AmiQuote được cung cấp.

Sau trong hướng dẫn này, bạn sẽ tìm thấy hướng dẫn chi tiết về **cách sử dụng AmiQuote.**

Nguồn trích dẫn cho AmiBroker (danh sách này không hoàn thiện - hãy để ý rằng hầu hết mọi nguồn có thể được sử dụng). Sử dụng liên kết để tìm hiểu thêm (lưu ý rằng một số liên kết đòi hỏi kết nối internet)

Quốc gia/Sàn giao dịch	Nguồn dữ liệu	Loại	Giá	Tải	Cập nhật	
	Yahoo Finance	Historical + Current EOD	Miễn phí	Tự động (AmiQuote)	Tự động	

	Dịch vụ Dữ liệu Norgate và Premium (Sản phẩm di sản của Premium Data)	Historical EOD + Cập nhật hàng ngày + Sectors / Industries / v.v.+ Ký hiệu đã bị hủy	Trả phí	Tự động	Tự động (qua MS plugin)	
	Norgate Data - new offering	Historical EOD + Cập nhật hàng ngày + Sectors / Industries / v.v. + Ký hiệu đã bị hủy	Trả phí	Tự động	Tự động (qua plugin chuyên dụng)	
	TC 2000/TCNet v7 (Cổ phiếu Mỹ) TC2000 Mutual Funds	Historical + Current EOD + Sectors/Industries	Trả phí	Tự động	Tự động	
	FastTrack (quỹ tương hỗ)	Historical + Current EOD + Families	Trả phí	Tự động	Tự động	
	FOREX	Historical EOD + Intraday	Miễn phí	Tự động (AmiQuote)	Tự động	
	CSI	Historical EOD	Trả phí	Tự động	Tự động	

Úc (Sở Giao dịch Chứng khoán Úc)	Dịch vụ Dữ liệu Norgate và Premium (Sản phẩm di sản của Premium Data)	Historical EOD + Cập nhật hàng ngày + Sectors / Industries / v.v. + Ký hiệu đã bị hủy	Có phí	Tự động	Tự động (thông qua MS plugin)	
	Bản cung cấp mới của Norgate Data	Historical EOD + Cập nhật hàng ngày + Sectors / Industries / v.v. + Ký hiệu đã bị hủy	Có phí	Tự động	Tự động (qua plugin chuyên dụng)	
	BodhiFreeway	Dữ liệu lịch sử	Có phí	Tự động (Bodhi downloader)	Tự động (thông qua METASTOCK plugin)	
	Yahoo Finance Úc	Cập nhật EOD hiện tại	Miễn phí	Tự động **(AmiQuote)**	Tự động	
Hơn 50 Sở giao dịch quốc tế	Yahoo Finance	Lịch sử + Cập nhật EOD hiện tại	Miễn phí	Tự động **(AmiQuote)**	Tự động	
Ba Lan (Sở Giao dịch Chứng khoán Warsaw)	Bossa.pl	Lịch sử + Cập nhật EOD hiện tại	Miễn phí	Tự động (dựa trên kịch bản)	Tự động (dựa trên kịch bản)	
Nam Phi (Johannesburg Stock Exchange)	Sharenet	Lịch sử + Cập nhật EOD hiện tại	Có phí	Tự động **(Sharenet)**	Tự động (dựa trên kịch bản)	Mô tả tại đây

	Dữ liệu Nhà đầu tư	Dữ liệu lịch sử	Có phí	Thủ công	Thủ công	
Hà Lan (Amsterdam - Euronext)	PF-online	Dữ liệu lịch sử + Dữ liệu EOD hiện tại	Miễn phí	Thủ công	Nhập Import ASCII	

Cách thiết lập AmiBroker với nguồn dữ liệu eSignal (phiên bản RT chỉ)

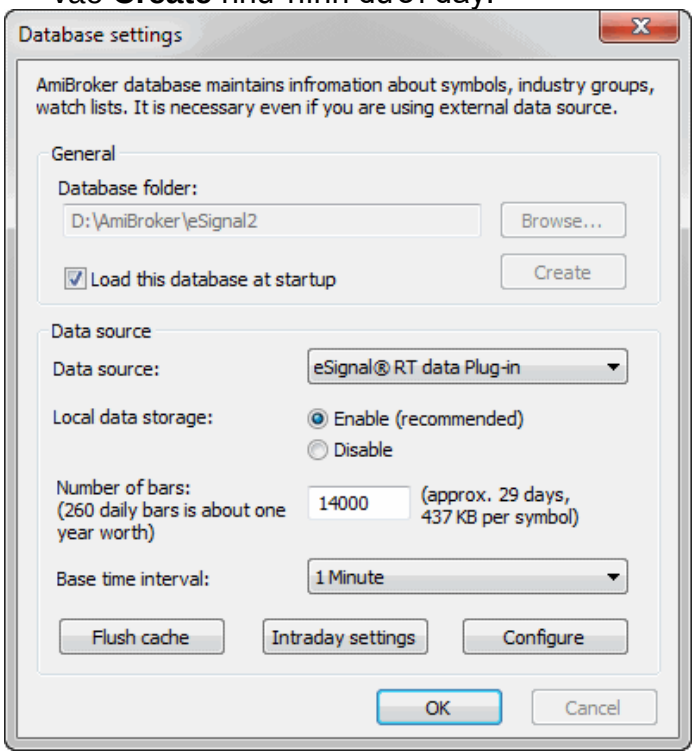
Yêu cầu

QUAN TRỌNG: **Bạn phải cài đặt ứng dụng eSignal trên máy của mình và có một đăng ký eSignal hợp lệ.**

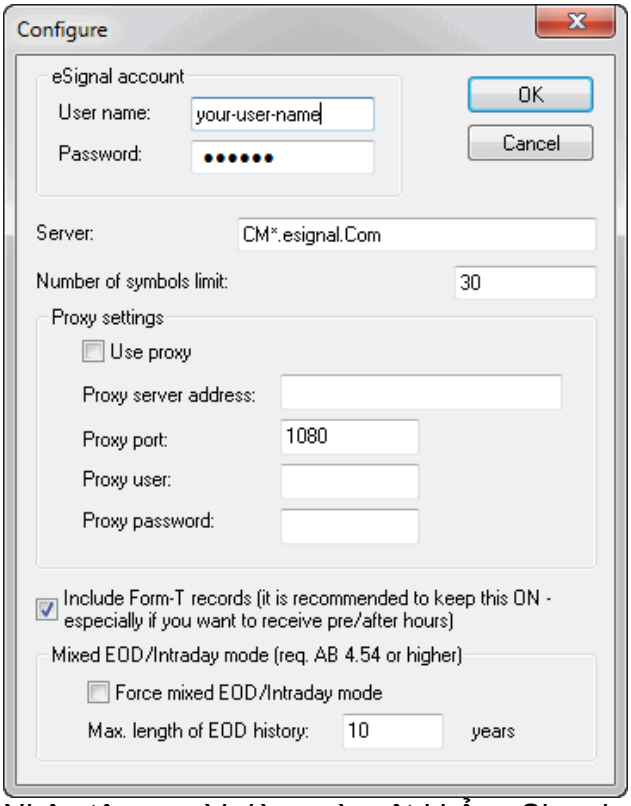
Thiết lập một lần

Để sử dụng AmiBroker với nguồn dữ liệu eSignal, bạn sẽ cần thực hiện một thiết lập một lần như mô tả dưới đây:

- Chạy AmiBroker
- Chọn **File -> New database**
- Nhập tên thư mục mới (ví dụ: C:\Program Files\AmiBroker\eSignal) và nhấp vào **Create** như hình dưới đây:



- Chọn **eSignal RT data Plug-in** từ danh sách nguồn dữ liệu và **"Enable"** từ **lưu trữ dữ liệu cục bộ**
- Nhập **số lượng thanh cần tải: 90000 cho cơ sở dữ liệu 1 phút kết hợp với cơ sở dữ liệu hàng ngày dài hạn**
- Nhấp vào nút **Configure** để hiển thị hộp thoại cấu hình plug-in như hình dưới đây



- Nhập tên người dùng và mật khẩu eSignal của bạn ở đây (nếu bạn đã cài đặt eSignal đúng cách, AmiBroker sẽ tự động điền các trường này bằng tên người dùng/mật khẩu đã nhập trong phần mềm eSignal). Bạn cũng có thể điều chỉnh Số lượng **ký hiệu**. Điều này không nên vượt quá giới hạn tài khoản của bạn, và bạn có thể xem xét giảm giá trị này nếu muốn sử dụng AmiBroker song song với ứng dụng quản lý dữ liệu khác. (Nếu bạn vượt quá giới hạn đăng ký, AmiBroker sẽ tự động điều chỉnh số này xuống)

Nhấp OK

- Bây giờ chọn Chu kỳ cơ bản. Lưu ý rằng **chu kỳ khuyến nghị là 1 phút,** nhưng bạn có thể chọn tất cả các khoảng thời gian cơ bản từ tick đến hàng giờ.

Lưu ý rằng việc chọn tick, khoảng thời gian 1 giây, 5 giây hoặc 15 giây sẽ gây ra việc truyền tải lượng dữ liệu khổng lồ từ máy chủ eSignal (đối với chứng khoán giao dịch tích cực, nó có thể lên tới vài megabyte cho chỉ một ký hiệu và rất ít ngày lịch sử). Nếu bạn đang sử dụng kết nối modem, thiết lập này được khuyến nghị không nên sử dụng. Ngoài ra, bạn nên xem xét sử dụng thanh 1 giây thay vì tick thuần túy vì chế độ này nhanh hơn.

Cũng lưu ý rằng để có được lịch sử cuối ngày dài hạn cùng với dữ liệu intraday, bạn nên vào File -> Database Settings -> Intraday Settings và bật tùy chọn **"Allow mixed EOD/intraday data".**

- Nhấp OK.

Từ giờ trở đi, AmiBroker của bạn sẽ đọc báo giá trực tiếp từ eSignal.

Để tìm hiểu cách sử dụng AmiBroker ở chế độ Thực thời, hãy đọc bài **viết hướng dẫn này.**

Cách thiết lập AmiBroker với nguồn dữ liệu myTrack (phiên bản RT chỉ)

Lưu ý: Phiên bản mới nhất của tài liệu này có thể được tìm thấy tại: http://www.amibroker.com/mytrack.html. Vui lòng kiểm tra trang này để cập nhật.

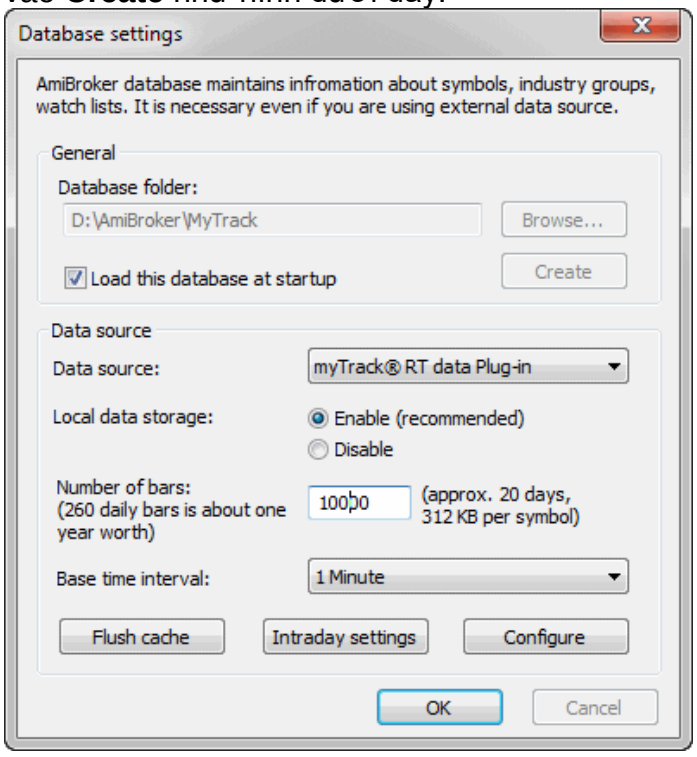
Yêu cầu

QUAN TRỌNG: Bạn phải có đăng ký myTrack với tính năng SDK được kích hoạt. Để sử dụng SDK, hãy chạy chương trình myTrack, nhấp vào CHAT, sau đó vào Entitlements và sau đó vào Features, đánh dấu ô SDK.

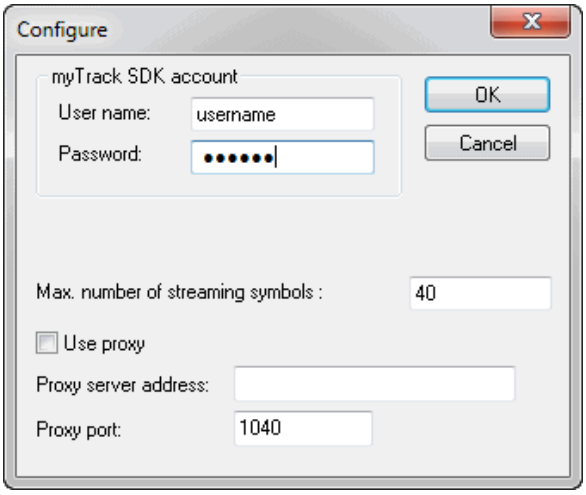
Thiết lập một lần

Để sử dụng AmiBroker với nguồn dữ liệu myTrack, bạn sẽ cần thực hiện một thiết lập một lần như mô tả dưới đây:

- Chạy AmiBroker
- Chọn **File -> New database**
- Nhập tên thư mục mới (ví dụ: C:\Program Files\AmiBroker\myTrack) và nhấp vào **Create** như hình dưới đây:



- Chọn **myTrack RT data Plug-in** từ danh sách nguồn dữ liệu và **"Enable"** từ **lưu trữ dữ liệu cục bộ**
- Nhấp vào nút **Configure** để hiển thị hộp thoại cấu hình plug-in như hình dưới đây



Nhập tên người dùng và mật khẩu myTrack của bạn ở đây. Bạn cũng có thể điều chỉnh **Số lượng ký hiệu.** Điều này không nên vượt quá giới hạn tài khoản của bạn.

Nhấp OK

- Bây giờ chọn Khoảng thời gian cơ bản. Lưu ý rằng các khoảng thời gian thanh được hỗ trợ là **1 phút** và **hàng ngày (cuối ngày).**

Nếu bạn muốn có lịch sử hàng ngày dài và biểu đồ intraday, bạn nên xem xét chạy HAI phiên bản AmiBroker. Một cho biểu đồ EOD và một cho biểu đồ intraday. Cả hai phiên bản đều có thể sử dụng myTrack làm nguồn dữ liệu.

- Nhấp OK.

Từ giờ trở đi, AmiBroker của bạn sẽ đọc báo giá trực tiếp từ myTrack.

Để tìm hiểu cách sử dụng AmiBroker ở chế độ Thực thời, hãy đọc bài viết **hướng dẫn này.**

Cách sử dụng AmiBroker với nguồn dữ liệu bên ngoài (Quote Tracker)

QUAN TRỌNG: Bạn cần QuoteTracker 2.4.9C HOẶC CAO HƠN (3.1.0 được khuyến nghị). Có thể hoạt động trên phiên bản tiêu chuẩn nhưng AmiBroker RT được khuyến nghị.

RẤT QUAN TRỌNG: QuoteTracker phải được CẤU HÌNH để máy chủ nội bộ của nó đang chạy. Nhấp vào đây để biết giải thích.

LƯU Ý: QuoteTracker nên được xem xét như một thay thế thời gian thực cho người nghèo. Hiệu suất của nó không thể so sánh với nguồn cấp dữ liệu thời gian thực thực sự như eSignal hoặc myTrack, cung cấp các cập nhật rất đáng tin cậy, lịch sử dài hạn và cập nhật tick theo tick thực sự.

Plug-in QuoteTracker hiện đang hoạt động trong HAI chế độ:

Chế độ hàng ngày - plug-in thêm và cập nhật thanh cuối cùng (hôm nay) với các báo giá mới nhất trong thời gian gần như thực - điều này có nghĩa là bạn phải sử dụng nó cùng với cơ sở dữ liệu cuối ngày đã có.

Chế độ intraday - plug-in cung cấp dữ liệu lịch sử intraday trong một ngày - nhiều ngày có thể được tích lũy nếu AmiBroker với QT được khởi động hàng ngày để AmiBroker có thể lưu trữ lịch sử vào cơ sở dữ liệu cục bộ của nó.

Thiết lập một lần

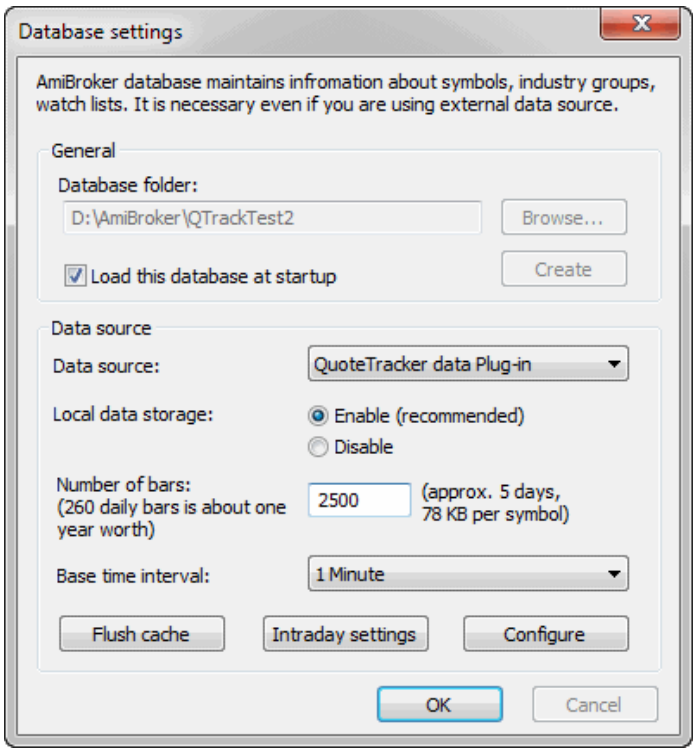
Đảm bảo rằng QuoteTracker của bạn đã bật máy chủ QT HTTP: **Options -> Edit Preferences: tab Misc: HTTP Server Settings. Nếu bạn đang sử dụng phiên bản chưa đăng ký của QuoteTracker, hãy đảm bảo rằng bạn nhấp vào quảng cáo đủ thường xuyên.**

Để sử dụng nguồn dữ liệu bên ngoài với AmiBroker, bạn sẽ cần thực hiện một thiết lập một lần như mô tả dưới đây:

• Chạy AmiBroker

• Chọn **File -> New database**

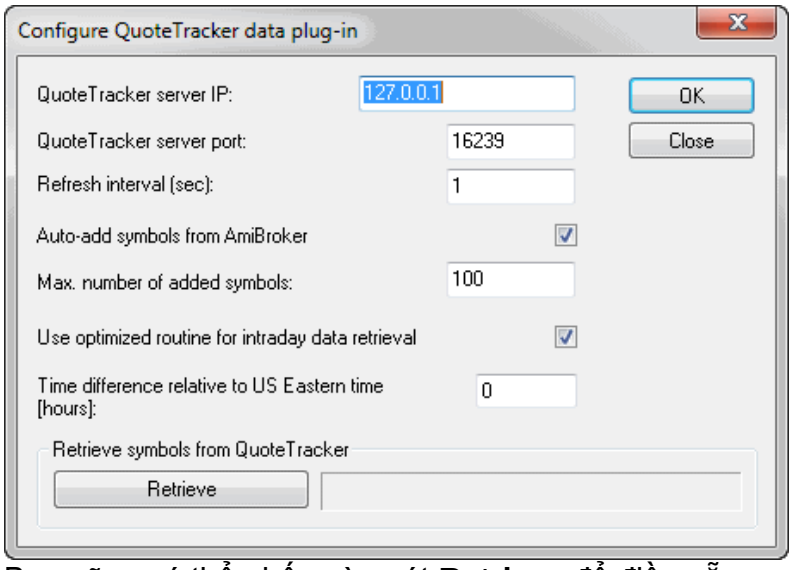
• Nhập tên thư mục mới (ví dụ: C:\Program Files\AmiBroker\NewData) và nhấp vào **Create** như hình dưới đây:



Chọn mục phù hợp từ combo nguồn dữ liệu:

- Người dùng Quote Tracker chọn **"Quote Tracker plug-in"** làm **nguồn dữ liệu** và "**Enable**" từ **bộ nhớ dữ liệu cục bộ**

Nhấp vào nút **Configure** để hiển thị hộp thoại cấu hình plugin như hình dưới đây

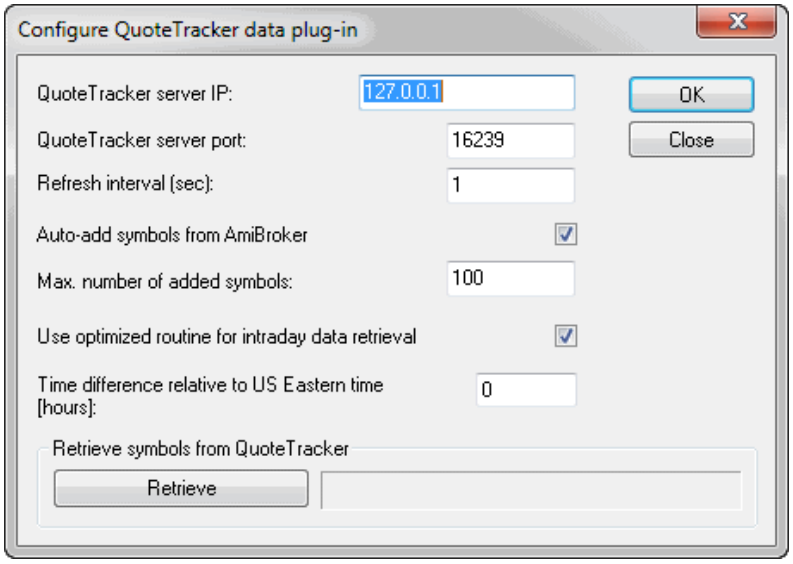


Bạn cũng có thể nhấp vào nút **Retrieve** để điền sẵn cơ sở dữ liệu AmiBroker với các ký hiệu đã có trong QuoteTracker. Từ bây giờ, AmiBroker của bạn đọc báo giá từ Quote Tracker gần như thời gian thực.

Để tìm hiểu cách sử dụng AmiBroker ở chế độ thời gian thực, hãy đọc bài viết hướng dẫn này.

Mô tả các tùy chọn cấu hình plugin QuoteTracker

QT plugin configuration dialog trông như sau:



Cổng máy chủ QuoteTracker: xác định cổng mà máy chủ HTTP nội bộ của QT có thể nhìn thấy. 16239 là giá trị mặc định được sử dụng bởi QuoteTracer và bạn không nên thay đổi điều này trong hầu hết các trường hợp. Nếu có nghi ngờ, vui lòng kiểm tra cài đặt máy chủ HTTP của QuoteTracker: **Options->Edit Preferences : Tab Misc: Menu cài đặt máy chủ HTTP của QT.**

Khoảng thời gian làm mới - định nghĩa bao lâu thì AmiBroker sẽ yêu cầu QT truy xuất báo giá. Mặc định là 5 giây. Bạn có thể xem xét việc thay đổi nó thành 10 hoặc 15 giây trong trường hợp bạn có nhiều biểu tượng và máy móc chậm.

Tự động thêm biểu tượng từ AmiBroker - nếu tùy chọn này được bật (mặc định là bật) nếu bạn chuyển đổi trong AmiBroker đến biểu tượng mà không có trong bất kỳ danh mục đầu tư QT nào - nó sẽ tự động được thêm vào danh mục đầu tư QT mặc định. Điều này cũng được áp dụng cho bất kỳ loại truy cập nào khác (ví dụ: nếu bạn cố gắng nhập biểu tượng vào AmiBroker và chúng không tồn tại trong QT - chúng sẽ được thêm nếu tùy chọn này được bật). Tắt chức năng này sẽ vô hiệu hóa chức năng thêm tự động.

Số biểu tượng tối đa được thêm - xác định số lượng tối đa các biểu tượng được thêm bằng chức năng thêm tự động được mô tả ở trên. Điều này bảo vệ QuoteTracker khỏi việc quá tải (AmiBroker có thể xử lý hàng chục ngàn biểu tượng một cách dễ dàng nhưng QuoteTracker KHÔNG thể)

Sử dụng cách tiếp cận tối ưu hóa để lấy dữ liệu trong ngày - bật tùy chọn này (mặc định, được khuyến nghị) sẽ tăng tốc đáng kể việc lấy dữ liệu trong các chế độ trong ngày. Nếu tùy chọn này được kích hoạt và AmiBroker đã có dữ liệu trong ngày từ trước đó thì AmiBroker chỉ yêu cầu QT một số bản ghi thời gian và bản ghi giao dịch cuối cùng đã xảy ra kể từ lần cập nhật cuối cùng đến thời điểm hiện tại, nếu tùy chọn này bị vô hiệu hóa thì AmiBroker luôn yêu cầu QT các bản ghi thời gian và giao dịch từ cả ngày.

Sự chênh lệch thời gian so với thời gian ở phía đông nước Mỹ - sự chênh lệch thời gian (theo giờ) giữa thời gian địa phương của bạn và thời gian ở phía đông nước Mỹ (EST). Trường này là cần thiết vì máy chủ của QuoteTracker báo cáo tất cả các thời gian theo múi giờ EST. Điều này có nghĩa là nếu bạn sống ở Australia, QuoteTracker sẽ báo cáo các báo giá ASX theo múi giờ EST và sẽ lệch 15 giờ so với giờ địa phương của bạn. Trong khi AmiBroker có thiết lập để điều chỉnh biểu đồ intraday và điều này không phải là vấn đề khi sử dụng chế độ Intraday, nó trở thành vấn đề khi sử dụng chế độ hàng ngày (EOD) vì các báo giá được báo cáo bởi QuoteTracker sẽ lệch một ngày. Thiết lập này giải quyết vấn đề này vì AmiBroker sẽ thêm số giờ đã nhập vào thời gian được báo cáo bởi QuoteTracker để có được ngày hợp lệ của báo giá trong chế độ hàng ngày. Trường này sẽ được điền với sự chênh lệch tính toán dựa trên cài đặt thời gian của Windows của bạn.

Nhận ký hiệu từ QuoteTracker - nhấn nút "Retrieve" sẽ thêm tất cả các ký hiệu có trong QuoteTracker vào danh sách ký hiệu của AmiBroker.

Cách Cài Đặt Amibroker VớI Dữ Liệu Iqfeed (Chỉ Phiên Bản Rt)

Lưu ý: phiên bản mới nhất của tài liệu này có thể được tìm thấy tại: http://www.amibroker.com/iqfeed.html . Hãy kiểm tra trang này để cập nhật.

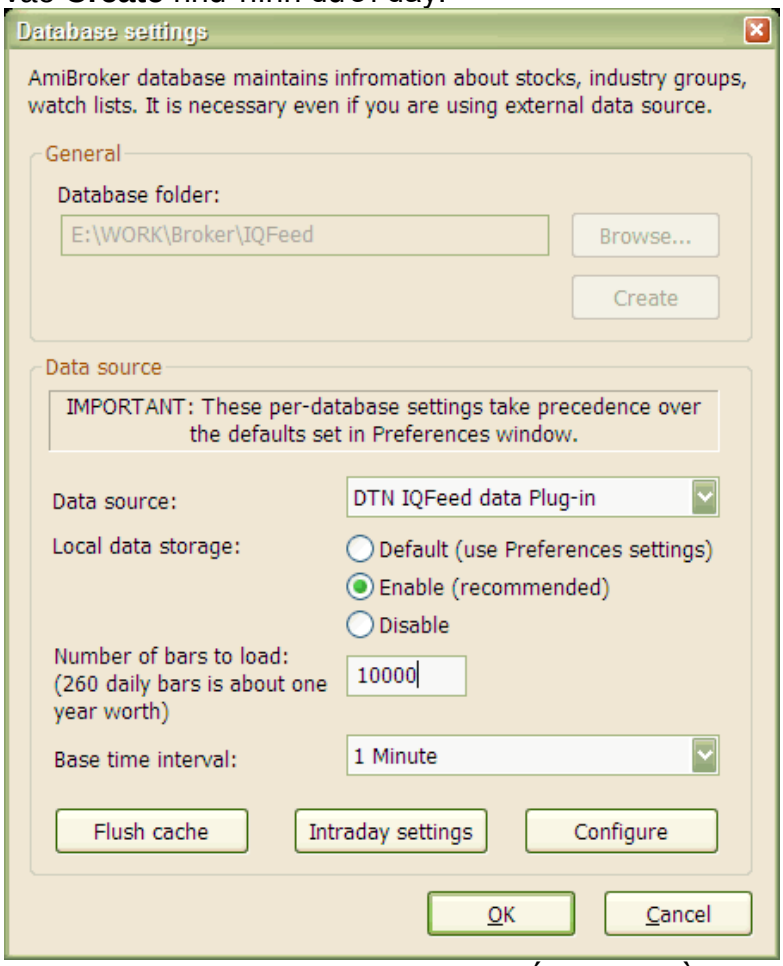
Yêu cầu

Nếu bạn chưa cài đặt IQFeed CONNECTION MANAGER, bạn cần cài đặt nó trước. Bạn có thể tải về cài đặt khách IQFeed từ đây (phiên bản 4.2.0.7).

Http://Www.Amibroker.Com/Video/Iqfeed.Html

Để sử dụng AmiBroker với IQFeed, bạn sẽ cần thực hiện cài đặt một lần được mô tả dưới đây:

- Chạy AmiBroker
- Chọn **File -> New database**
- Nhập tên thư mục mới (ví dụ: C:\Program Files\AmiBroker\NewData) và nhấp vào **Create** như hình dưới đây:



- Chọn **DTN IQFeed data Plug-in** từ Kết hợp nguồn dữ liệu và **"Enable"** từ **Lưu trữ dữ liệu địa phương**
- Bây giờ chọn Khoảng thời gian cơ sở. Chọn 1 phút
- Nhập **số lượng thanh phù hợp để tải: 10000 cho cơ sở dữ liệu 1 phút để có lịch sử tối đa (8 tháng) có sẵn từ IQFeed**

- **Nhấn vào "Cấu hình trong ngày"**. **Kiểm tra hộp "Cho phép dữ liệu EOD/Intraday hỗn hợp". Nhấn OK**
- Nhấn OK.

Từ giờ trở đi, AmiBroker của bạn sẽ đọc trực tiếp các báo giá từ IQFeed.

Để tìm hiểu cách sử dụng AmiBroker trong chế độ Real Time, hãy đọc bài hướng dẫn này.

Cách Sử Dụng Amibroker Với Interactive Brokers Tws

Lưu ý: phiên bản mới nhất của tài liệu này có thể được tìm thấy tại: http://www.amibroker.com/ib.html. Hãy kiểm tra trang này thường xuyên để cập nhật.

Tóm tắt Tính Năng Plugin Ib:

- hỗ trợ đến 100 biểu tượng luồng trực tuyến theo thời gian thực (bằng với giới hạn IB TWS)
- kết nối tự động (không cần chấp nhận kết nối đến một cách manual trong TWS)
- hỗ trợ đến **30 NGÀY dữ liệu INTRADAY với PHỤC HỒI trong khoảng thời gian bar 1 phút**
- lên đến 2000 bars backfill sử dụng khoảng thời gian bar 1-giây/5-giây/15-giây

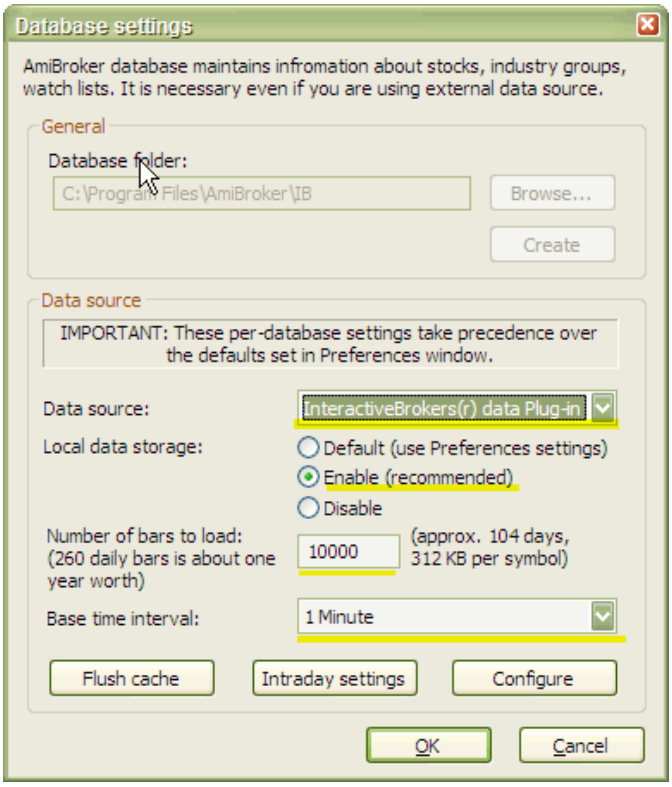
Hướng Dẫn:

LƯU Ý: Interactive Brokers TWS là ứng dụng đòi hỏi nhiều CPU, do đó để có kết quả tốt nhất, chúng tôi khuyên bạn nên sử dụng máy có bộ xử lý 1GHz hoặc nhanh hơn.

LƯU Ý 2: Có một bài hướng dẫn VIDEO hướng dẫn cách thiết lập tại http://www.amibroker.com/video/ib.html

Để sử dụng plugin dữ liệu Interactive Brokers với AmiBroker, bạn cần thực hiện như sau:

1. chạy **TWS dựa trên web hoặc tải xuống TWS độc lập**
2. trong TWS, chọn **Configure -> API -> Enable Active X and Socket clients** Cũng **nhập 127.0.0.1 trong TWS, Configure->API->Trusted IP addresses** để ngăn chặn hộp thoại "Allow incoming connection?".
3. Chạy AmiBroker và tạo cơ sở dữ liệu mới với plugin Interactive Brokers làm nguồn dữ liệu, theo các bước sau:
    - Chạy AmiBroker
    - Chọn File->New database
    - Gõ tên thư mục mới (ví dụ: C:\Program Files\AmiBroker\IB) và nhấp vào **Create** như hình dưới:



- Chọn **plug-in dữ liệu InteractiveBrokers(r)** từ danh sách nguồn dữ liệu và **"Enable"** từ **lưu trữ dữ liệu cục bộ.**
- Nhập 30000 hoặc hơn vào trường **"Số lượng thanh cần tải".**
- Bây giờ chọn Khoảng thời gian cơ bản. **Các khoảng thời gian hỗ trợ là: EOD, hàng giờ, 15 phút, 5 phút, 1 phút. Phiên bản Professional của AmiBroker cũng cho phép chọn Tick, khoảng thời gian 5 giây, 15 giây.**

  Lưu ý rằng việc điền dữ liệu (backfill) chỉ trong khoảng thời gian thanh 1 phút hoặc ít hơn (hạn chế của TWS).

  Nếu bạn muốn có lịch sử hàng ngày dài hạn và biểu đồ intraday, bạn nên xem xét chạy HAI phiên bản AmiBroker. Một cho biểu đồ EOD và một cho biểu đồ intraday. Cả hai phiên bản đều có thể sử dụng IB làm nguồn dữ liệu.

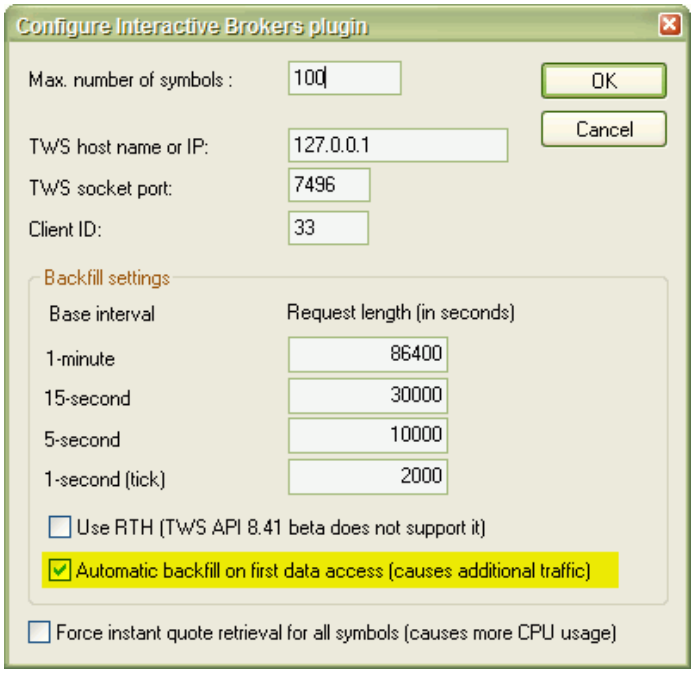
- Nhấp OK.

Từ giờ trở đi, AmiBroker của bạn sẽ đọc báo giá trực tiếp từ Interactive Brokers.

CÁCH SỬ DỤNG TÍNH NĂNG BACKFILL

Tính năng backfill trong plug-in 1.3.7 cho phép tải 24 dữ liệu lịch sử intraday để lấp đầy các khoảng trống có thể xảy ra khi AmiBroker / TWS không chạy.

Tính năng backfill của IB có thể được cấu hình từ **File -> Database Settings, Configure:**



Hai cài đặt chính liên quan đến backfill là:

1. độ dài yêu cầu
2. backfill tự động

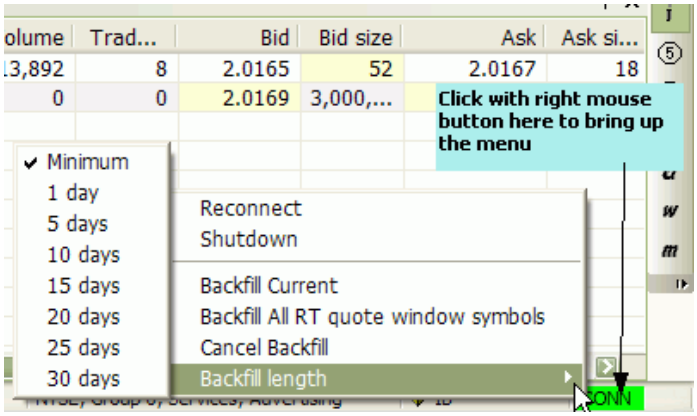
Khi xem xét độ dài yêu cầu, như đã giải thích trong TWS API Release Notes tại: **http://www.interactivebrokers.com/en/software/apiReleaseNotes/apiBetanotes.php,** tính năng backfill của IB hiện tại bị giới hạn ở một số khoảng thời gian / khoảng cách thanh cố định. Ví dụ, bạn có thể nhận tối đa 2000 tick 1 giây, tối đa 10000 giây ở khoảng thời gian 5 giây (2000 thanh), tối đa 30000 giây ở khoảng thời gian 15 giây (cũng 2000 thanh) và tối đa 5 NGÀY của các thanh 1 phút.

Theo mặc định, AmiBroker sử dụng các mức tối đa cho phép.

Đối với "backfill tự động khi truy cập dữ liệu lần đầu" - khi được chọn, AmiBroker sẽ cố gắng lấp đầy dữ liệu cho ký hiệu khi bạn hiển thị một biểu đồ cho ký hiệu đó (hoặc thực hiện backtest hoặc quét). Xin lưu ý rằng API TWS hiện chỉ cho phép một backfill tại một thời điểm, vì vậy khi có một backfill đang chạy ở chế độ nền, yêu cầu backfill tự động cho ký hiệu tiếp theo sẽ bị bỏ qua cho đến khi backfill trước đó hoàn thành.

Tùy chọn này tiện lợi, tuy nhiên có thể gây thêm tải cho kết nối internet của bạn vì dữ liệu cần tải trong quá trình backfill.

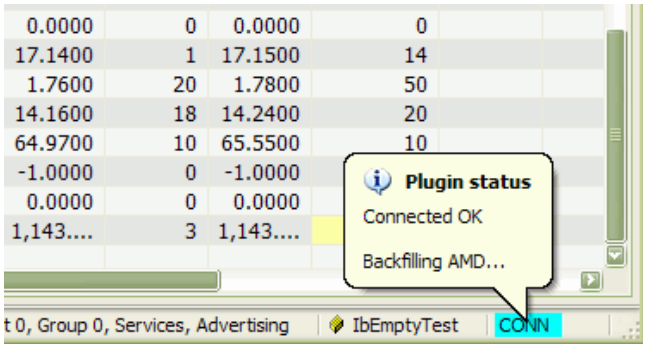
Nếu bạn tắt tùy chọn "backfill tự động khi truy cập dữ liệu lần đầu", bạn vẫn có thể lấp đầy dữ liệu cho ký hiệu hiện tại hoặc tất cả các ký hiệu trong danh sách cửa sổ báo giá thời gian thực bằng cách sử dụng các tùy chọn menu thích hợp từ menu trạng thái của plug-in.



Tùy chọn **Backfill Current** cho phép ép buộc backfill cho biểu tượng hiện đang được chọn, trong khi **Backfill All RT quote window symbols** cho phép ép buộc backfill cho tất cả các biểu tượng được liệt kê trong cửa sổ Trích dẫn thời gian thực. Việc backfill của nhiều biểu tượng được thực hiện tuần tự (một lúc một) do các hạn chế của TWS.

Menu phụ **Backfill length** cho phép chọn độ dài backfill mong muốn.

Trong quá trình backfill, một hình nhỏ sẽ xuất hiện để thông báo cho người dùng về biểu tượng đang được backfill và màu của plugin sẽ thay đổi thành màu xanh lơ nhạt (màu ngọc lục bảo) như hình bên dưới:



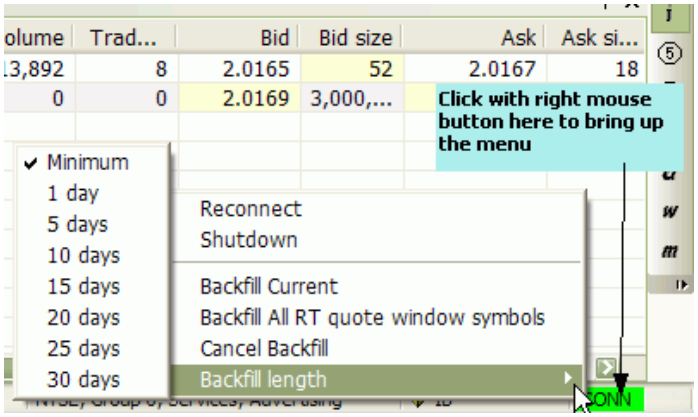
Backfill Tất Cả Biểu Tượng Cùng Một Lúc

Để backfill tất cả các biểu tượng cùng một lúc, hãy làm như sau:

1. Mở cửa sổ **Trích dẫn thời gian thực** (bằng cách chọn menu **Window->Realtime Quote**)
2. Nhấp chuột phải vào cửa sổ **Trích dẫn thời gian thực** và chọn **Add symbol / Add watch list / Type-in symbol** để thêm bất kỳ biểu tượng nào bạn muốn backfill



3. Chuột phải vào **chỉ số trạng thái của plugin** và chọn **độ dài Backfill** mong muốn



4. Chọn tùy chọn "**Backfill All RT quote window symbols"** từ cùng một menu.

Vì Interactive Brokers giới hạn nghiêm ngặt số lượng backfills mà khách hàng có thể yêu cầu trong một khoảng thời gian nhất định, nên nên sử dụng độ dài backfill ngắn nhất có thể, như 1 ngày hoặc 5 ngày và tránh khoảng dài như 30 ngày.

Ký hiệu

Định dạng ký hiệu bây giờ sử dụng chế độ ký hiệu của TWS, không phải chế độ nền tảng. Chế độ ký hiệu trong TWS có thể thấy trong tùy chọn menu **'Xem->Chế độ ký hiệu'** trong TWS.

Định dạng là: *BIỂU TƯỢNG-SỔ GIAO DỊCH-LOẠI*

nơi

SYMBOL là cùng một ký hiệu như cột ký hiệu được hiển thị trong TWS khi ở chế độ ký hiệu.

EXCHANGE (tùy chọn) là sàn giao dịch trong TWS khi ở chế độ ký hiệu.

TYPE (tùy chọn) là một trong các loại sau:

STK - cổ phiếu, FUT - hợp đồng tương lai, FOP - lựa chọn trên hợp đồng tương lai, OPT - lựa chọn, IND - chỉ số, CASH -tiền mặt (FX lý tưởng) Chú ý rằng chỉ có trường EXCHANGE và TYPE của cổ phiếu là tùy chọn. Sàn giao dịch sẽ được thiết lập thành BEST (SMART) và kiểu sẽ được thiết lập thành STK.

Hãy cẩn thận khi nhập các ký hiệu vì một số trong số chúng (hợp đồng tương lai) có NHIỀU KHOẢNG TRẮNG trong tên ký hiệu. Bạn phải nhập CHÍNH XÁC SỐ LƯỢNG khoảng trắng giống như trong các ví dụ dưới đây (xem các dấu gạch dưới tên ký hiệu giúp dễ nhận biết số lượng ký tự)

Ví dụ:

Mã	Loại	Mô tả
CSCO	Cổ phiếu	Cisco Corporation, Nasdaq
GE	Cổ phiếu	General Electric, NYSE
VOD-LSE	Cổ phiếu	VODAFONE GROUP, Sở Giao dịch Chứng khoán London
ESM4-GLOBEX-FUT	Hợp đồng tương lai	Emini ES Jun04 futures, Globex
QQQFJ-CBOE-OPT	Quyền chọn	Jun 04, 36.0 CALL option QQQFJ
INDU-NYSE-IND	Chỉ số	Dow Jones Industrials Index
YM JUN 04-ECBOT-FUT	Hợp đồng tương lai	YM Jun 04 future, ECBOT
QMN5-NYMEX-FUT	Hợp đồng tương lai	QM (Crude) June 2005 future contract, NYMEX
XAUUSD-SMART-CMDTY	Hàng hóa	London Gold Spot
IBUS500-SMART-CFD-USD	CFD (hợp đồng chênh lệch)	IB US500 contract for difference

EUR.USD-IDEAL-CASH	Forex tiền tệ	EURUSD currency pair, IDEAL EURUSD currency pair, IDEALPRO

Lại lần nữa:

Các ký hiệu hợp đồng tương lai ECBOT có độ dài 21 ký tự với 3 khoảng trắng giữa ký hiệu hợp đồng và tên tháng và một khoảng trắng giữa tháng và năm 2 chữ số

Hợp đồng		3 khoảng trống			Tháng			Khoảng trống	Năm		-	E	C	B	O	T	-	F	U	T
Z	B				J	U	N		0	4	-	E	C	B	O	T	-	F	U	T
Z	F				J	U	N		0	4	-	E	C	B	O	T	-	F	U	T
Z	N				J	U	N		0	4	-	E	C	B	O	T	-	F	U	T
Y	M				J	U	N		0	4	-	E	C	B	O	T	-	F	U	T

GHI CHÚ VỀ CÁC HẠN CHẾ CỦA IB API:

1. **Tính năng backfill chỉ có sẵn cho các tài khoản IB thực (không áp dụng cho tài khoản demo).**
2. Giá mở **(Open** price) KHÔNG được cung cấp bởi IB. Do đó, trường Giá mở sẽ trống trong cửa sổ báo giá thời gian thực.
3. Dữ liệu từ IB không bao gồm dấu thời gian (timestamp) cho các giao dịch. Thời gian hệ thống hiện tại được sử dụng để gán dấu thời gian cho từng tick.
4. **Dữ liệu phát trực tiếp từ IB TWS KHÔNG phải là theo từng giao dịch (tick-by-tick), mà là các bức ảnh chụp cách nhau 0.2-0.3 giây, bạn có thể đọc thêm chi tiết tại đây: Interactive brokers**

Cách sử dụng AmiBroker với nguồn dữ liệu DDE bên ngoài

Lưu ý: phiên bản mới nhất của tài liệu này có thể tìm thấy tại: http://www.amibroker.com/dde.html . Vui lòng kiểm tra trang này để cập nhật.

Dde Là Gì

DDE (Dynamic Data Exchange) là một giao thức của Windows được sử dụng để cho phép các ứng dụng trao đổi dữ liệu. Ví dụ, khi bạn thay đổi một biểu mẫu trong chương trình cơ sở dữ liệu của mình hoặc một mục dữ liệu trong chương trình bảng tính, chúng có thể được thiết lập để cũng thay đổi những biểu mẫu hoặc mục này ở bất cứ đâu chúng xuất hiện trong các chương trình khác mà bạn có thể sử dụng. DDE sử dụng mô hình client/server trong đó ứng dụng yêu cầu dữ liệu được coi là client và ứng dụng cung cấp dữ liệu được coi là server.

Hàng nghìn ứng dụng sử dụng DDE, bao gồm Excel của Microsoft, Word, Lotus 1-2-3, và Visual Basic.

Để biết thêm thông tin về DDE như một cơ chế giao tiếp trong Windows, vui lòng theo liên kết này:
http://msdn.microsoft.com/library/en-us/winui/WinUI/WindowsUserInterface/DataExchange/DynamicDataExchange/AboutDynamicDataExchange.asp

Dde Dành Cho Nhà Giao Dịch

DDE mang lại gì cho nhà giao dịch? Cơ bản là dữ liệu sống thời gian thực. **DDE không cung cấp BACKFILL**. Rất nhiều nhà cung cấp dữ liệu thời gian thực và các sàn giao dịch cung cấp khả năng lấy dữ liệu thời gian thực bằng cách sử dụng DDE. Bạn nên hỏi sàn giao dịch/nhà cung cấp dữ liệu thời gian thực của mình nếu họ có hỗ trợ DDE. Plugin DDE hiện có cho AmiBroker cho phép liên kết với (gần như) bất kỳ nguồn DDE (server) nào cung cấp dữ liệu sống thời gian thực. Điều này tạo ra một lựa chọn hấp dẫn cho tất cả các nguồn dữ liệu không có plugin chuyên dụng

Khi Nào Không Nên Sử Dụng Plugin Dde

Nếu bạn đang sử dụng eSignal, IQFeed, Quote.com, và bất kỳ nguồn nào khác có plugin dédicacé - bạn nên sử dụng plugin dédicacé này thay vì DDE. Điều này là do plugin dédicacé luôn là lựa chọn tốt hơn (cung cấp thêm tính năng và chúng nhanh hơn) so với DDE thông thường.

Tổng Kết Tính Năng Của Plugin Dde

DDE server/topic/item có thể xác định bởi người dùng cho mỗi trường (open, high, low, close, volume, trade size, total volume, bid, bid size, ask, ask size, time)

hỗ trợ tối đa 500 symbols phát trực tiếp thời gian thực (phiên bản 1.1.0)

hỗ trợ tất cả các khoảng thời gian cơ bản: hàng ngày, hàng giờ, 15 phút, 5 phút, 1 phút, 15 giây, 5 giây, tick

KHÔNG CÓ BACKFILL (do hầu hết các nguồn DDE không cung cấp dữ liệu backfill)

LỊCH SỬ

1.2.2 - bao gồm trường "Time shift" trong hộp thoại ngữ cảnh, lưu cấu hình theo từng cơ sở dữ liệu trong tệp dde.config thay vì trong registry, cộng với một số cải tiến nhỏ khác.

1.2.1 - đã sửa vấn đề với 'type mismatch'.

1.2.0 - plugin mặc định sử dụng định dạng số theo cài đặt vùng miền và giảm tải CPU.

1.1.0 - giới hạn ký hiệu tăng từ 40 lên 500.

1.0.0 - phát hành ban đầu (BETA).

HƯỚNG DẪN

Để sử dụng plugin dữ liệu DDE với AmiBroker, bạn cần thực hiện các bước sau:

(tùy chọn*)Tải phiên bản mới nhất của plugin DDE từ [http://www.amibroker.com/bin/DDE.dll] và sao chép nó vào thư mục con PLUGINS của thư mục AmiBroker.

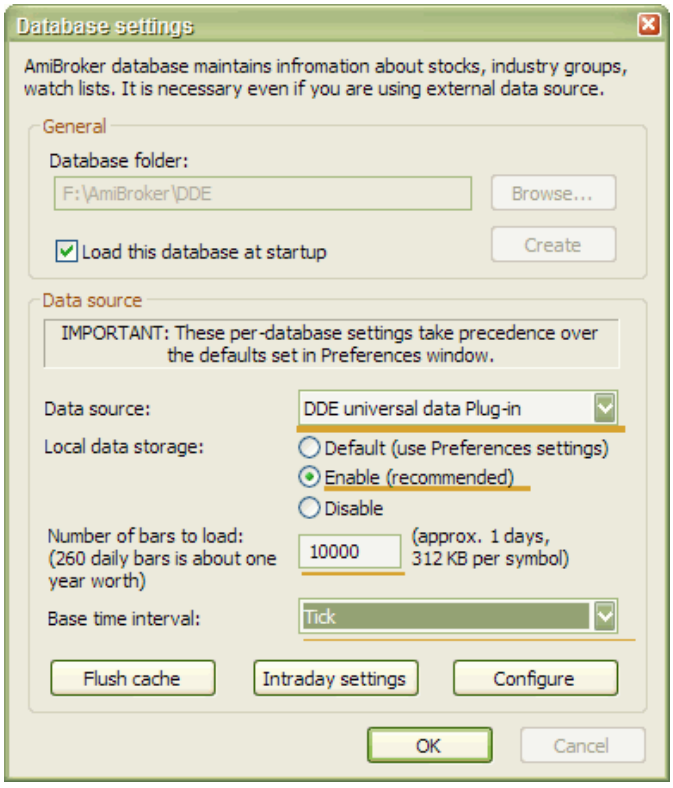
Kích hoạt DDE trong phần mềm bên thứ ba mà bạn đang sử dụng làm máy chủ DDE (tham khảo tài liệu của nhà cung cấp dữ liệu/công ty môi giới để biết chi tiết về cách kích hoạt DDE).

Chạy AmiBroker và tạo cơ sở dữ liệu mới với **"DDE universal data plugin"** làm nguồn dữ liệu, thực hiện theo các bước sau:

Chạy AmiBroker.

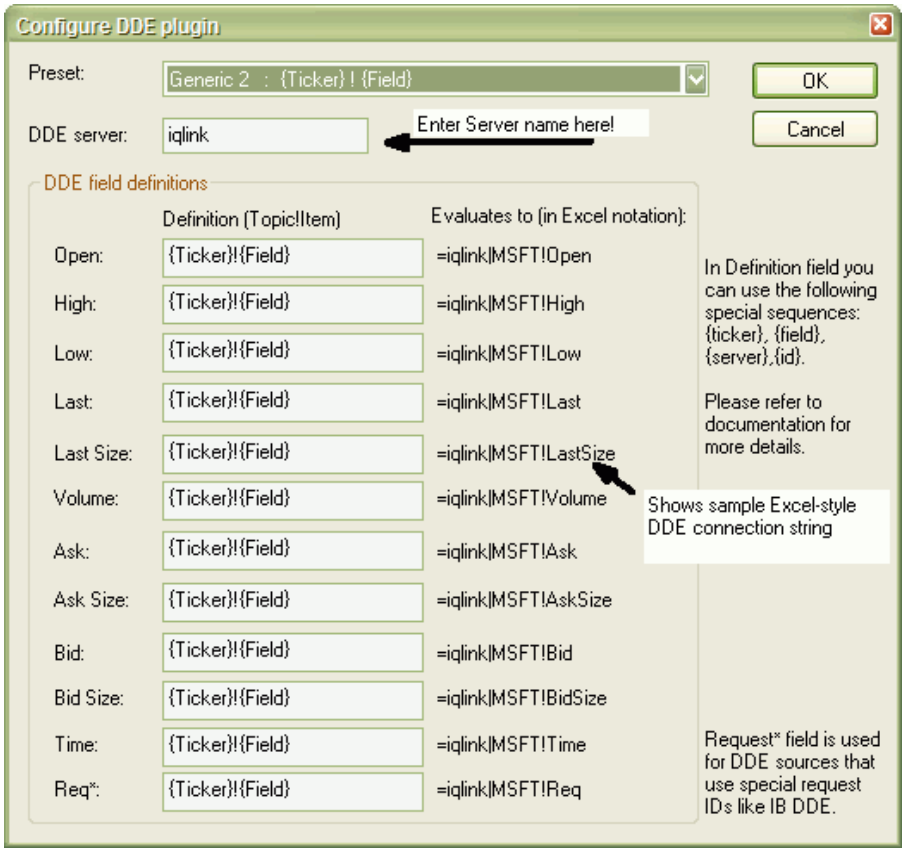
Chọn **File -> New database**

Nhập tên thư mục mới (ví dụ: C:\Program Files\AmiBroker\DDE) và nhấp vào **Create** như trong hình dưới đây:



- Chọn **DDE universal data plugin** từ danh sách nguồn dữ liệu và **"Enable"** từ **lưu trữ dữ liệu cục bộ.**
- Nhập 10000 hoặc nhiều hơn vào trường **"Number of bars to load".**
- Bây giờ chọn khoảng thời gian cơ bản. C**ác khoảng thời gian được hỗ trợ bao gồm: EOD, hàng giờ, 15 phút, 5 phút, 1 phút. Phiên bản Professional của AmiBroker cũng cho phép chọn khoảng thời gian Tick, 5 giây, 15 giây.**
- Nhấp vào nút **CONFIGURE** - QUAN TRỌNG: trong hộp thoại "CONFIGURE", bạn phải thiết lập tất cả các trường theo mô tả của nhà cung cấp dữ liệu của bạn.

Vui lòng kiểm tra thêm đoạn bên dưới (**"CẤU HÌNH PLUGIN DDE ĐỂ LÀM VIỆC VỚI NHÀ CUNG CẤP CỦA BẠN"**) để biết mô tả chi tiết. CHÚ Ý: bạn không thể bỏ qua phần này - nếu không thiết lập các trường đặc biệt cho nhà cung cấp dữ liệu của bạn, DDE SẼ KHÔNG HOẠT ĐỘNG.



- Nhấp OK

Chỉ số tình trạng Plugin nên thay đổi từ Màu vàng "WAIT" thành Màu xanh lá cây "OK" trong vòng vài giây. Nếu nó không chuyển thành trạng thái "OK" tức là có thể: a) tên máy chủ và/hoặc các trường không được thiết lập chính xác

hoặc

b) DDE server (ứng dụng của bên thứ ba) không đang hoạt động hoặc không được kích hoạt.

Nếu chỉ số hiển thị "OK" - thì dữ liệu báo giá thời gian thực được truyền vào AB. Bạn có thể kiểm tra điều này bằng cách hiển thị Window->Real time quote. Lưu ý: do không có backfill, bạn sẽ cần chờ ít nhất 3 thanh dữ liệu được thu thập trước khi biểu đồ hiển thị lên.

Cấu hình Dde Plugin để làm việc với nhà cung cấp của bạn

Các nhà cung cấp dữ liệu khác nhau sử dụng các chuỗi kết nối DDE khác nhau, dưới đây là một số ví dụ điển hình.

Hầu hết tài liệu về DDE sử dụng cú pháp DDE của Excel có dạng như sau:=SERVER|TOPIC!ITEM

Server là tên của DDE server như WINROS, IQLINK, REUTER, CQGPC, MT, MTLink, v.v.
Topic là chủ đề của cuộc trò chuyện DDE. Tuỳ thuộc vào nguồn dữ liệu, chủ đề có thể chỉ là biểu tượng ticker (như trong IQFeed), hoặc tên trường (như trong winros).

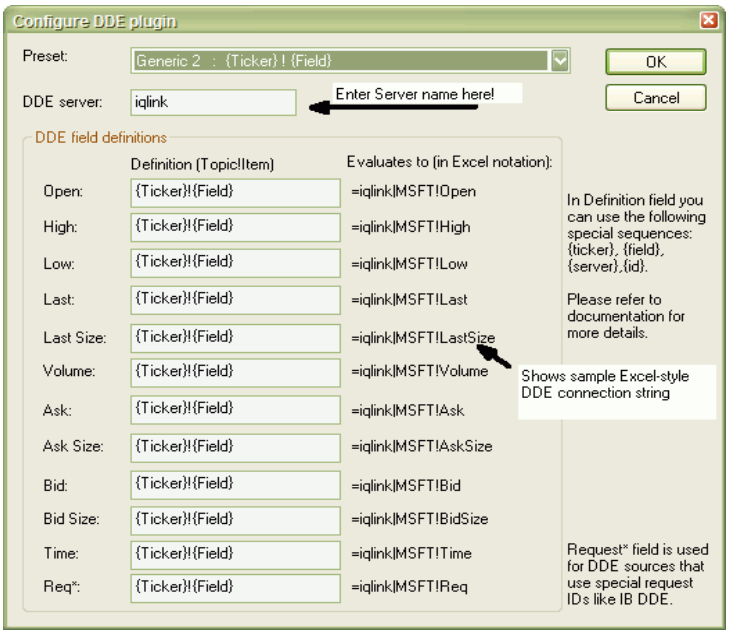
Item là mục của cuộc trò chuyện DDE. Tùy thuộc vào nguồn dữ liệu, nó có thể là tên trường (như trong IQFeed) hoặc biểu tượng ticker (như trong Winros).

Vì vậy, chuỗi kết nối DDE theo hai chuẩn phổ biến nhất có dạng như sau:

=WINROS|LAST!MSFT

=IQLINK|MSFT!LAST

Bây giờ màn hình cấu hình plugin DDE trông như thế này:



Ở phía TRÊN của hộp thoại, bạn có thể thấy trường "DDE Server". Trong trường này, bạn nên nhập phần SERVER của chuỗi kết nối DE (=SERVER|TOPIC!ITEM) mà không có dấu bằng và không có ký tự |.

Bên dưới, bạn có thể thấy 12 ô nhập văn bản nơi bạn có thể xác định chủ đề DDE và mục cho mỗi trường dữ liệu mà nguồn dữ liệu của bạn cung cấp. Ở đây, bạn nên nhập cặp TOPIC!ITEM của chuỗi kết nối DDE (=SERVER|TOPIC!ITEM) với dấu chấm than giữa chủ đề DDE và mục DDE.

Như bạn có thể thấy trong hình trên, plugin DDE cho phép bạn sử dụng một số chuỗi đặc biệt, cụ thể là: {Ticker}, {Field}, {FieldSp}, {Server}, {Id} được đánh giá trong thời gian chạy cho mỗi biểu tượng riêng biệt cho phép xây dựng chuỗi DDE động (tùy thuộc vào mã chứng khoán đã chọn ví dụ) được yêu cầu bởi hầu hết nguồn dữ liệu:

{Ticker} - đánh giá để biểu tượng chứng khoán của bảo mật đã cho

{Field} - đánh giá để tên trường tương ứng (không có dấu cách), tức là Open, High, Low, Last,LastSize, Volume, Ask, AskSize, Bid, BidSize, Time, Req

{FieldSp} - tương tự như {field} nhưng tên trường 2 từ có dấu cách, cụ thể là: "Last Size", "Ask Size", "Bid Size"

{Server} - đánh giá để tên máy chủ

{Id} - đánh giá để ID duy nhất (bộ đếm chạy tăng thêm 1 với mỗi biểu tượng)

Tất cả các văn bản khác đều được sao chép nguyên bản, vì vậy nếu bạn viết ví dụ:

PREFIX_{Ticker }_SUFFIX!MYTEXT

Nó sẽ được đánh giá thành =SERVER|PREFIX_MSFT_SUFFIX!MYTEXT (giả sử rằng biểu tượng hiện tại là MSFT).

Bên cạnh các định nghĩa trường, chúng ta có thể thấy rằng định nghĩa nhất định sẽ được đánh giá như thế nào (theo ký hiệu Excel). Điều này giúp dễ dàng xác minh xem định nghĩa có đúng hay không.

Đánh giá mẫu luôn sử dụng "MSFT" làm {Ticker}, và 34 là {id}.

Nếu nguồn dữ liệu của bạn không cung cấp tất cả các trường, bạn có thể để trống trường nhất định. Lưu ý rằng để hoạt động đúng, giá "Cuối cùng" (giá của giao dịch cuối cùng) là bắt buộc. Nếu nguồn dữ liệu của bạn không cung cấp giá "cuối cùng" (hầu hết các nguồn forex không có "last"), bạn có thể buộc plugin DDE sử dụng "Bid" thay thế. Để làm được điều đó, bạn nên để trống trường "Last" và cung cấp cặp Topic!Item phù hợp trong trường "Bid". Lưu ý rằng các cặp Topic!Item nên được đánh giá thành các giá trị duy nhất.

Ở phần trên của hộp thoại, bạn có thể thấy combo-box "Preset".

Tính đến thời điểm hiện tại, nó cho phép đặt trước các trường bằng hai lược đồ chung: a) {Field}!{Ticker} - "last price" được đánh giá thành =SERVER|Last!MSFT b) {Ticker}!{Field} - "last price" được đánh giá thành =SERVER|MSFT!Last Trong tương lai, hộp "Preset" sẽ chứa nhiều cài đặt sẵn hơn cho các nguồn DDE mà bạn gửi.

Vài Ví dụ

Các ví dụ về kết nối được trình bày trên trang web: http://www.amibroker.com/dde.html

Nền tảng Kiểm tra

Plugin DDE đã được kiểm tra và nó đã được biết là hoạt động đúng trên Windows XP (32 bit DDE) và Windows 9x (16 bit DDE). Các máy chủ DDE sau đây đã được xác nhận bởi chúng tôi là hoạt động đúng:

IQLINK(DTN)

WINROS(eSignal)

MT (Metaquote)

DDE plugin KHÔNG hoạt động với các máy chủ DDE sau đây:

VTSPOT (Visual Trader) - do lỗi lập trình trong VisualTrader gây ra việc thư viện Microsoft DDEML DdeConnect bị treo ngay từ lần kết nối đầu tiên.

Tất cả các máy chủ DDE khác không được liệt kê ở trên nên hoạt động đúng. Liên hệ hỗ trợ tại amibroker.com nếu có vấn đề.

Giúp Chúng Tôi Giúp Người Khác:

Để giúp những người khác cấu hình plugin DDE cho nhà cung cấp dữ liệu của họ, khi bạn đã thành công trong việc kết nối với nhà cung cấp cụ thể của mình, hãy gửi cho chúng tôi một ghi chú kèm theo ảnh chụp màn hình của hộp thoại CẤU HÌNH và tên của nguồn dữ liệu. Thông tin này sẽ được đưa vào tài liệu sau này như một tham khảo cách sử dụng các nguồn dữ liệu khác nhau. Các thiết lập hoạt động cũng sẽ được thêm vào danh sách "cài đặt trước" để dễ dàng cấu hình chỉ với một cú nhấp chuột.

Ghi chú về Plugin Dde:

KHÔNG CÓ BACKFILL trong Plugin Dde. Bạn có thể sử dụng ASCII importer (bao gồm AmiQuote) để nhập dữ liệu lịch sử ngay vào cơ sở dữ liệu mà sau đó bạn sẽ cập nhật theo thời gian thực bằng cách sử dụng Plugin Dde.

Các trường Thay đổi, % thay đổi hiện không có sẵn (tuy nhiên có thể thay đổi trong tương lai)

Các trường Thời gian (Time) và hiện đang được bỏ qua (điều này cũng có thể thay đổi trong tương lai)

Thời gian hệ thống hiện tại được sử dụng để đánh dấu mỗi tick.

Khi nguồn của bạn không cung cấp giá "LAST" (như một số nguồn Forex), bạn nên để trường "Last" TRỐNG trong hộp thoại cấu hình. Điều này sẽ chỉ cho plugin sử dụng trường "BID" thay thế.

Trạng thái plugin (kết nối / ngắt kết nối) luôn xuất hiện lúc đầu với trạng thái "Wait" (đèn báo màu vàng). Điều này có nghĩa là chưa có cuộc trò chuyện DDE nào được thiết lập. Nếu ít nhất MỘT cuộc trò chuyện DDE bắt đầu thành công, nó sẽ chuyển sang trạng thái "OK" (đèn chỉ báo màu xanh lá cây). Nếu máy chủ DDE không hoạt động khi cố gắng kết nối lần đầu, plugin sẽ KHÔNG tự động thử kết nối lại. Thay vào đó, bạn nên buộc việc kết nối lại theo cách thủ công (xem điểm7). Đèn báo có thể chuyển sang "Ngắt kết nối" (màu đỏ) chỉ trong hai trường hợp:

a) bạn đã kết nối một cách đúng đắn nhưng máy chủ DDE (ứng dụng của bên thứ 3) đã được đóng

b) bạn đã chọn "tắt" từ menu trạng thái plugin

Bạn có thể kết nối lại bất cứ lúc nào bằng cách chọn "kết nối lại" từ menu trạng thái plugin.

Cách Làm Việc Với Trình Bổ Sung Dữ Liệu Thời Gian Thực Lần Đầu Thiết Lập

Thiết lập một lần

Để sử dụng AmiBroker với bất kỳ nguồn dữ liệu thời gian thực nào, bạn phải thiết lập cơ sở dữ liệu với trình bổ sung dữ liệu phù hợp trước. Điều này chỉ yêu cầu một lần lúc tạo cơ sở dữ liệu. Hướng dẫn thiết lập có sẵn tại đây: eSignal, myTrack, IQFeed, QuoteTracker.

Kiểm tra trang nguồn dữ liệu trực tuyến tại http://www.amibroker.com/quotes.html cho các trình bổ sung mới.

Thêm Biểu Tượng

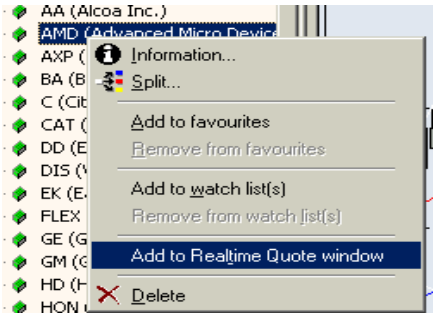
Bây giờ bạn có thể thêm các biểu tượng vào cơ sở dữ liệu của mình. Để làm như vậy, vui lòng vào menu **Biểu tượng->Mới.** Trong hộp thoại thêm biểu tượng, nhập một hoặc nhiều mã giao dịch (phân cách bằng dấu phẩy) mà bạn muốn thêm vào cơ sở dữ liệu của mình. Nếu bạn muốn xem biểu đồ cho biểu tượng mới được thêm vào, chỉ cần chọn nó từ cây biểu tượng ở cửa sổ không gian làm việc. Hãy cho phép vài giây (tùy thuộc vào tốc độ kết nối internet của bạn) để lấp đầy dữ liệu lịch sử.

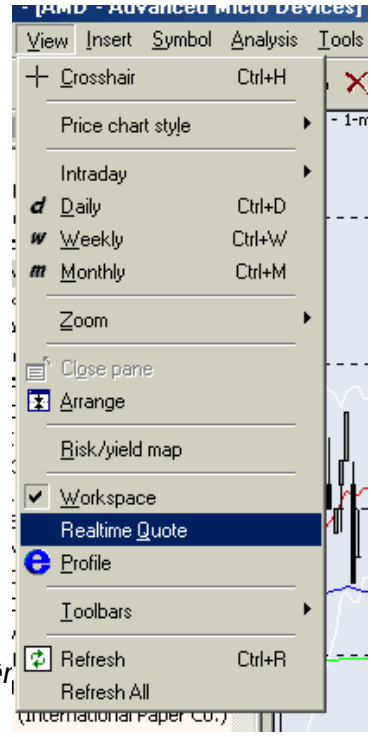
Bạn có thể thêm nhiều mã giao dịch hơn tài khoản RT của bạn cho phép. AmiBroker sẽ tự động chuyển đổi/cập nhật/tải lại các biểu tượng để các biểu tượng được sử dụng gần đây nhất hoạt động và những cái hơn tuổi tự động được gỡ bỏ khỏi Data manager. Tuy nhiên, làm như vậy có thể dẫn đến một số vấn đề nếu bạn vượt quá giới hạn thuê bao quá nhiều. Vì vậy, bạn nên sử dụng tính năng này một cách có trách nhiệm và không kỳ vọng có 500 biểu tượng trong khi thuê bao của bạn chỉ giới hạn ở 50.

Lưu ý rằng cơ chế ở trên không áp dụng cho cửa sổ báo giá thời gian thực và nó không thể chứa nhiều hơn số biểu tượng mà giới hạn thuê bao của bạn.

Hiển Thị Cửa Sổ Trích Dẫn Thời Gian Thực

AmiBroker RT có tính năng cửa sổ xem thời gian thực cho phép bạn xem các báo giá phát trực tiếp. Để hiển thị cửa sổ này, chọn menu **Window->Realtime Quote.** (xem hình ảnh bên phải ---->)

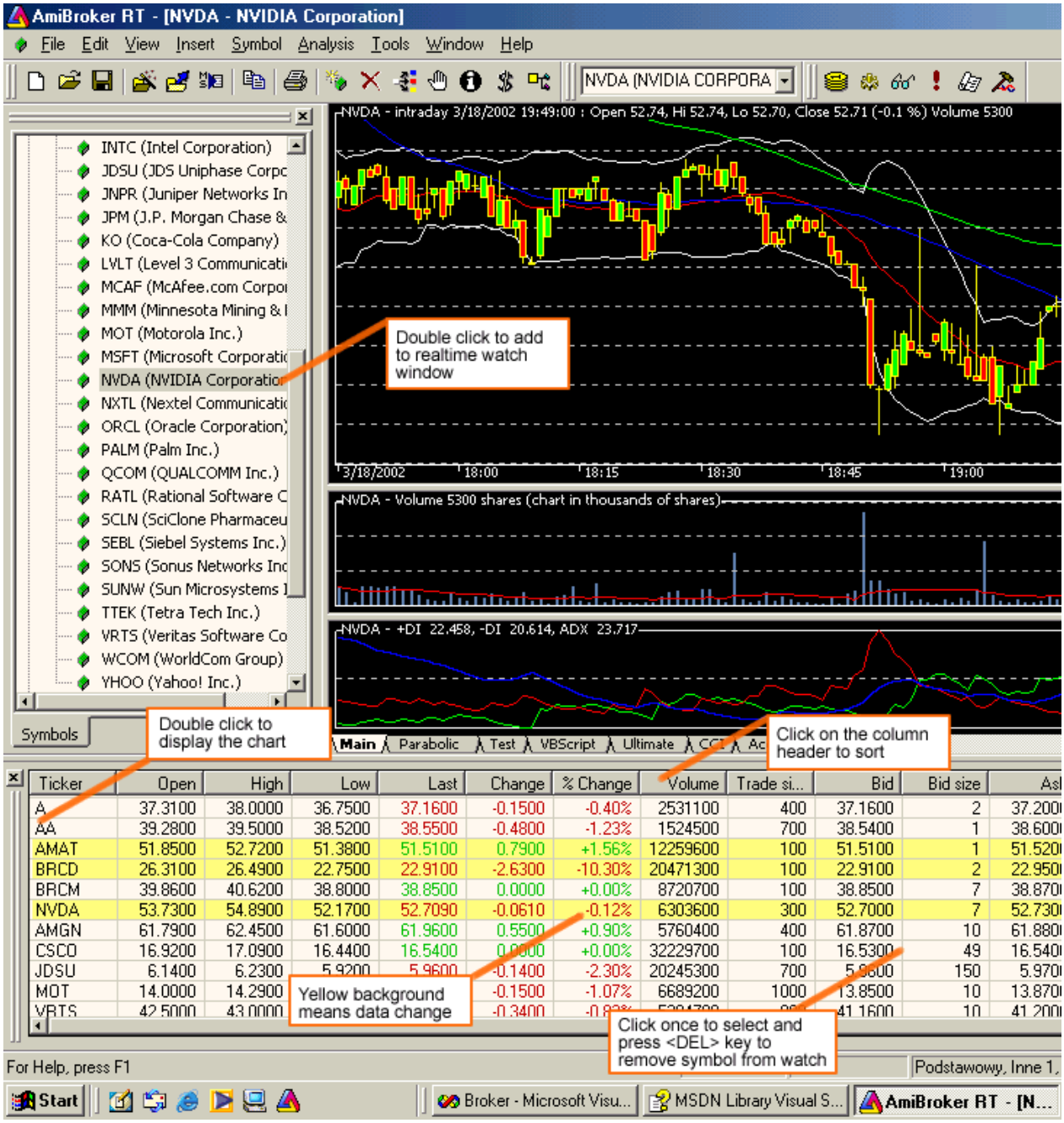




Để thêm các biểu tượng vào cửa sổ báo giá thời gian thực, bạn có thể nhấp đúp vào cây biểu tượng hoặc sử dụng menu click chuột phải tùy chọn "Add to Realtime quote" như hình ảnh trên.

Làm việc với cửa sổ Báo giá Thời gian thực

Cửa sổ báo giá RT cung cấp báo giá trực tuyến theo thời gian thực và một số dữ liệu cơ bản. Nó khá dễ sử dụng như hình dưới đây



Bạn cũng có thể hiển thị menu ngữ cảnh bằng cách nhấn chuột phải vào cửa sổ báo giá thời gian thực (RT quote window).



Menu ngữ cảnh cho phép bạn truy cập các tùy chọn sau:

- **Thời gian & Giao dịch**

Mở cửa sổ Thời gian & Giao dịch, cung cấp thông tin về từng lệnh mua, lệnh bán và giao dịch đang diễn ra trên thị trường.

- **Cảnh báo dễ dàng**

Mở cửa sổ Cảnh báo dễ dàng, cho phép bạn định nghĩa các cảnh báo thời gian thực được thực hiện khi các giá trị lệnh mua/lệnh bán/giao dịch cuối cùng và các trường khác đạt đến mức người dùng xác định.

- **Thêm ký hiệu**

Thêm ký hiệu hiện tại vào danh sách báo giá thời gian thực.

- **Thêm danh sách theo dõi**

  Thêm toàn bộ danh sách theo dõi vào cửa sổ báo giá thời gian thực.

- **Xóa ký hiệu**

  Xóa dòng đã chọn (ký hiệu) khỏi danh sách báo giá thời gian thực.

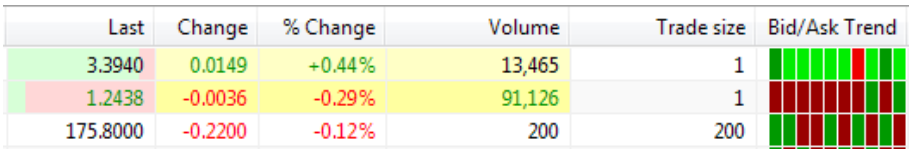
- **Xóa tất cả**

  Xóa tất cả các ký hiệu khỏi danh sách báo giá thời gian thực.

- **Ẩn**

  Ẩn danh sách báo giá thời gian thực.

Chỉ báo xu hướng lệnh mua/lệnh bán



Phiên bản 5.90 thêm chỉ báo xu hướng Lệnh Mua/Lệnh Bán - một chỉ báo đồ họa hiển thị hướng đi của 10 thay đổi gần đây nhất trong giá lệnh mua/lệnh bán theo thời gian thực. Ô bên phải nhất là thay đổi gần đây nhất, và khi có các báo giá lệnh mua/lệnh bán mới đến, chúng sẽ được dịch sang bên trái. Mã màu như sau:

- Xanh đậm: lệnh mua > lệnh mua trước đó HOẶC lệnh bán > lệnh bán trước đó

- Xanh sáng: lệnh mua > lệnh mua trước đó VÀ lệnh bán > lệnh bán trước đó

- Đỏ đậm: lệnh mua < lệnh mua trước đó HOẶC lệnh bán < lệnh bán trước đó

- Đỏ sáng: lệnh mua < lệnh mua trước đó VÀ lệnh bán < lệnh bán trước đó

- Hộp đỏ/xanh: lệnh bán < lệnh bán trước đó VÀ lệnh mua > lệnh mua trước đó

- Hộp xanh/đỏ: lệnh bán > lệnh bán trước đó VÀ lệnh mua < lệnh mua trước đó

Nếu giá lệnh mua/lệnh bán không thay đổi, sẽ không có ô mới. LƯU Ý: Cột này chỉ hoạt động nếu có các báo giá thời gian thực đang chạy (thị trường đang mở).

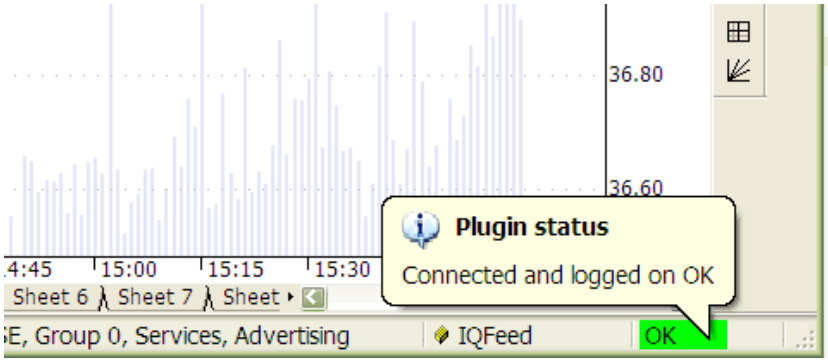
Làm việc với biểu đồ intraday và hàng ngày

Nếu nguồn dữ liệu của bạn hỗ trợ chế độ EOD/Intraday hỗn hợp (như eSignal hoặc IQFeed), bạn có thể sử dụng một cơ sở dữ liệu cho cả hai loại biểu đồ.

Tuy nhiên, nếu nguồn dữ liệu của bạn không hỗ trợ chế độ EOD/Intraday hỗn hợp và nếu bạn muốn có lịch sử hàng ngày dài và các biểu đồ intraday, bạn nên xem xét việc chạy HAI phiên bản của AmiBroker. Một phiên cho biểu đồ EOD và một cho biểu đồ intraday. Cả hai có thể sử dụng cùng một nguồn dữ liệu thời gian thực.

Hiển thị trạng thái kết nối

Trạng thái kết nối của plug-in dữ liệu được hiển thị trong khu vực hiển thị trạng thái plug-in nằm ở góc dưới bên phải của cửa sổ chính AmiBroker như trong hình dưới đây. Khi trạng thái kết nối thay đổi, AmiBroker phát ra âm beep và hiện lên một chú thích thông báo để thông báo về sự thay đổi trạng thái.



Chú thích cung cấp thông tin chi tiết hơn và tự động biến mất sau 2 giây.

Nếu bạn muốn hiển thị lại, chỉ cần di chuột qua khu vực hiển thị trạng thái plug-in.

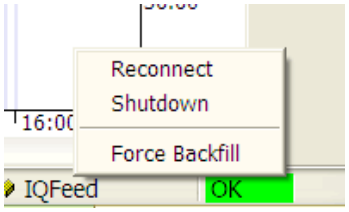
Để dễ dàng kiểm tra trạng thái kết nối, AmiBroker hiển thị thông tin mã màu:

- **OK (đèn xanh)** có nghĩa là kết nối đang hoạt động tốt và chỉ ra sự hoạt động đúng đắn của plug-in.
- **WAIT (đèn vàng)** có nghĩa là kết nối đang được thiết lập ngay bây giờ hoặc plug-in chỉ được kết nối một phần (tới một vài trong số nhiều máy chủ). Thông thường, trạng thái này là tạm thời và trong vài giây, trạng thái sẽ quay lại "OK".

- **ERR (đèn đỏ)** có nghĩa là kết nối bị đứt. Điều này có thể có nghĩa là tên người dùng/mật khẩu không hợp lệ cho đăng ký của bạn, hoặc thực tế là một thành phần/chương trình bên thứ ba nào đó cần thiết không đang chạy (ví dụ: nếu QuoteTracker không đang chạy và bạn đang sử dụng plug-in QuoteTracker). Trạng thái này thường yêu cầu người dùng can thiệp, chẳng hạn như kiểm tra/sửa chữa tên người dùng/mật khẩu trong Tệp -> Cài đặt cơ sở dữ liệu -> Cấu hình hoặc chạy thành phần cần thiết. Khi bạn sửa lỗi, plug-in sẽ tự động cố gắng kết nối lại (và nếu kết nối lại thành công thì "OK" sẽ được hiển thị).
- **SHUT (đèn tím)** có nghĩa là đã xảy ra một vấn đề nghiêm trọng và plug-in sẽ không cố gắng kết nối lại tự động. Trong hầu hết các trường hợp, bạn phải đầu tiên sửa vấn đề gây ra trạng thái này và sau đó kết nối lại thủ công bằng cách sử dụng menu ngữ cảnh plug-in được mô tả bên dưới. Hoặc bạn có thể chỉ cần khởi động lại AmiBroker.

Sử dụng menu ngữ cảnh của plugin

Các plugin thời gian thực cung cấp một số điều khiển bổ sung qua menu ngữ cảnh của plugin. Menu ngữ cảnh này có sẵn khi bạn nhấn chuột phải vào khu vực hiển thị trạng thái của plugin. Nếu bạn làm vậy, menu như sau sẽ được hiển thị:



Xin lưu ý rằng các plugin khác nhau cung cấp các tùy chọn khác nhau trong menu này, tuy nhiên hầu hết các plugin cung cấp ít nhất 3 tùy chọn cơ bản và hữu ích:

- **Kết nối lại -** tùy chọn này cho phép bạn kết nối lại thủ công. Hầu hết các plugin RT cố gắng kết nối lại tự động, nhưng đôi khi việc kết nối lại thủ công là cần thiết.
- **Tắt (Ngắt kết nối)**- tùy chọn này cho phép tắt plugin RT. Điều này hữu ích khi bạn muốn dừng việc truyền dữ liệu báo giá.
- **Làm đầy dữ liệu lại** - tùy chọn này khiến plugin tải lại toàn bộ lịch sử (intraday) từ máy chủ. Thông thường, plugin tự động xử lý tất cả các dữ liệu đã làm đầy lại, vì vậy bạn không cần phải kích hoạt việc làm đầy lại bằng tay. Nếu plugin phát hiện bạn có một số báo giá bị thiếu từ thanh cuối cùng có sẵn đến thời điểm hiện tại, nó sẽ tự động kích hoạt việc làm đầy lại. Nhưng... trong ít nhất hai trường hợp, tùy chọn này là hữu ích:
    - Làm đầy lại nhiều thanh hơn sau khi thay đổi cài đặt (khi bạn mở rộng 'số thanh để tải' trong hộp thoại Cài đặt cơ sở dữ liệu, bạn phải làm đầy lại cho các ký hiệu đã được làm đầy trước đó với số thanh nhỏ hơn).
    - Dọn dẹp các tick lỗi (khi bạn thấy một tick lỗi, bạn có thể thử làm đầy lại với hy vọng rằng nhà cung cấp dữ liệu đã dọn dẹp cơ sở dữ liệu của mình và bạn sẽ nhận được dữ liệu đã được sửa chữa - điều này hoạt động tốt với eSignal, nơi thực sự có vẻ sửa chữa các tick lỗi sau khi chúng xảy ra).

Những điều bạn KHÔNG nên làm, hoặc cần làm rất cẩn thận

Bạn nên lưu ý rằng khi sử dụng plugin dữ liệu, plugin sẽ kiểm soát cơ sở dữ liệu báo giá (xem bài viết Hiểu về các khái niệm cơ sở dữ liệu), vì vậy bạn KHÔNG nên nhập báo giá từ các tệp ASCII (bao gồm AmiQuote) cho các ký hiệu đã có sẵn trong cơ sở dữ liệu thời gian thực.

Nếu bạn làm vậy, plugin cuối cùng sẽ ghi đè lên các dữ liệu nhập của bạn bằng dữ liệu thời gian thực hoặc cơ sở dữ liệu của bạn sẽ bị hỏng (nếu bạn nhập dữ liệu cuối ngày vào cơ sở dữ liệu intraday).

Vì vậy, xin đừng nhập dữ liệu ASCII (đặc biệt là dữ liệu EOD) vào cơ sở dữ liệu intraday thời gian thực được cung cấp bởi plugin.

Bạn có thể hỏi: tại sao điều này không bị vô hiệu hóa hoàn toàn. Câu trả lời là đôi khi điều này hữu ích và đôi khi nó sẽ hoạt động (nhưng những trường hợp này rất hiếm). Ví dụ, nó sẽ hoạt động nếu bạn nhập dữ liệu INTRADAY vào cơ sở dữ liệu intraday được cung cấp bởi plugin QuoteTracker và cả hai dữ liệu và dữ liệu nhập đều có cùng khoảng thời gian thanh.

Nó cũng hoạt động nếu bạn nhập dữ liệu cho các ký hiệu KHÔNG có trong cơ sở dữ liệu. Trong trường hợp này, các ký hiệu được nhập mới sẽ được đánh dấu bởi bộ nhập ASCII là "chỉ sử dụng cơ sở dữ liệu cục bộ cho ký hiệu này" (Xem cửa sổ Thông tin để biết chi tiết), vì vậy chúng sẽ bị LOẠI TRỪ khỏi cập nhật thời gian thực. Điều này hữu ích nếu bạn muốn nhập một số dữ liệu khác (ngay cả khi không phải là dữ liệu báo giá) và truy cập nó thông qua hàm ngoại trong khi sử dụng cơ sở dữ liệu thời gian thực của bạn.

Vì vậy, việc nhập ASCII không bị vô hiệu hóa trong cơ sở dữ liệu thời gian thực, nhưng bạn phải sử dụng nó với sự cẩn thận tối đa và biết những gì bạn đang làm.

Điều thứ hai là sử dụng Trình chỉnh sửa báo giá.

Mặc dù dữ liệu được kiểm soát bởi plugin, nhưng trong hầu hết các trường hợp, bạn có thể sử dụng Trình chỉnh sửa báo giá. Tuy nhiên, xin lưu ý rằng bạn chỉ có thể chỉnh sửa dữ liệu 1 phút hoặc khoảng thời gian cao hơn, và bạn chỉ có thể chỉnh sửa các ký hiệu đã được làm đầy hoàn toàn (không có việc làm đầy lại đang diễn ra cho ký hiệu cụ thể đó) và bạn sẽ KHÔNG thể chỉnh sửa ba thanh cuối cùng. Điều này là do ba thanh cuối cùng được lưu trong bộ nhớ cache của plugin. Vì vậy, bạn sẽ chỉ có thể chỉnh sửa chúng khi các thanh mới đến, khiến chúng trở thành 'cũ' hơn ba thanh cuối cùng.

Tính năng “CHỜ ĐỂ LÀM ĐẦY LẠI”

Người dùng của các plugin thời gian thực eSignal, myTrack và IQFeed giờ đây có thể đánh dấu ô "chờ để làm đầy lại" trong cửa sổ Phân tích Tự động và tất cả các quét, khám phá và làm đầy lại sẽ chờ cho quá trình làm đầy lại hoàn thành cho ký hiệu đã cho. Cờ này không ảnh hưởng đến các cơ sở dữ liệu không sử dụng plugin (các nguồn dữ liệu bên ngoài) hoặc sử dụng các plugin cuối ngày (như FastTrack, QP2, TC2000/TCNet, v.v.). Cờ này cũng không ảnh hưởng khi sử dụng plugin QT do thực

tế là QuoteTracker tự quản lý việc làm đầy lại và không cung cấp bất kỳ điều khiển nào cho quá trình làm đầy lại cho các ứng dụng bên thứ ba.

LÀM ĐẦY LẠI TẤT CẢ CÁC KÝ HIỆU CÙNG LÚC

Để làm đầy lại tất cả các ký hiệu cùng lúc trong nguồn dữ liệu hỗ trợ tính năng "Chờ để làm đầy lại" (IQFeed, eSignal), bạn có thể sử dụng cửa sổ Phân tích. Quy trình như sau:

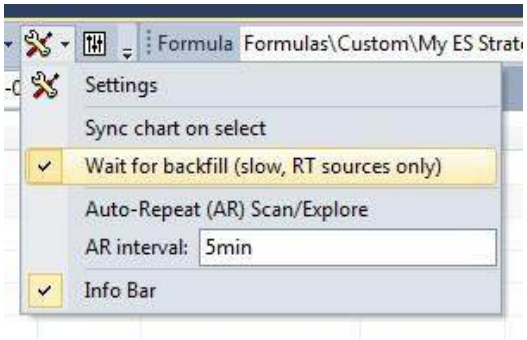
1. Mở Trình chỉnh sửa Công thức và nhập một quy tắc đơn giản như sau và chọn **Công cụ -> Gửi đến Phân tích**

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

2. Trong cửa sổ Phân tích, chọn **Áp dụng cho: *Tất cả các ký hiệu và Phạm vi: 1 thanh gần đây**



và bật tùy chọn **Chờ để làm đầy lại.**



3. Nhấn nút **Quét**.

Cửa sổ Phân tích sẽ lặp qua tất cả các ký hiệu, yêu cầu làm đầy lại cho mỗi ký hiệu và chờ cho đến khi dữ liệu đến, vì vậy cuối cùng tất cả các ký hiệu sẽ được làm đầy lại.

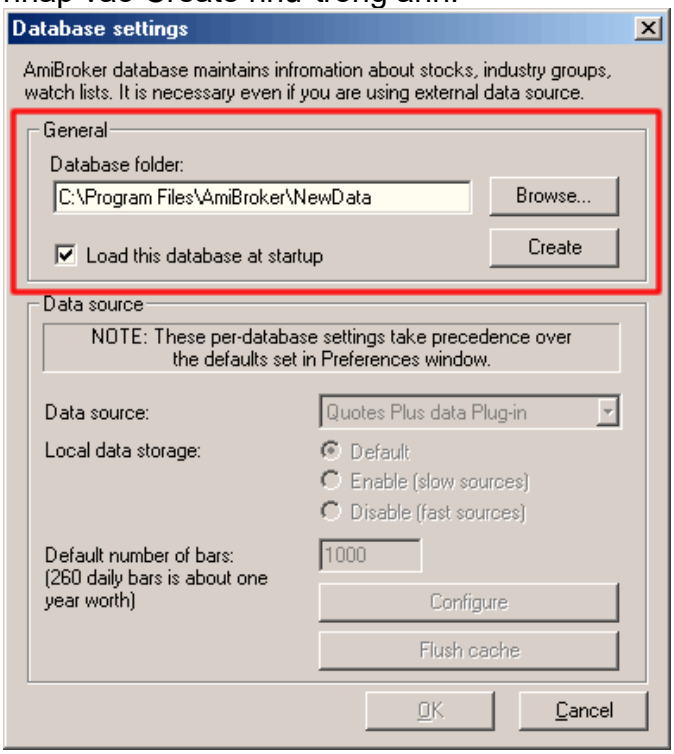
Cách Sử Dụng Amibroker Với Nguồn Dữ Liệu Ngoại Vi (Quotes Plus, Tc2000/Tcnet/Tc2005, Fasttrack, Metastock)

Một trong những tính năng mới được giới thiệu trong phiên bản AmiBroker 3.90 là khả năng đọc trực tiếp các cơ sở dữ liệu ngoại vi. Điều này được thực hiện nhờ vào các DLL plug-in dữ liệu cho phép kết nối cơ sở dữ liệu AmiBroker với một nguồn ngoại vi. Xin lưu ý rằng mặc dù bạn sẽ sử dụng cơ sở dữ liệu ngoại vi, bạn vẫn cần có một cơ sở dữ liệu AmiBroker để lưu trữ thông tin bổ sung không được hỗ trợ bởi nguồn ngoại vi như các nghiên cứu tự vẽ, gán cho các nhóm, danh sách theo dõi, hợp thành và vân vân. Bạn có thể tìm thêm thông tin về việc xử lý cơ sở dữ liệu AmiBroker tại đây.

Thiết Lập Một Lần

Để sử dụng một nguồn dữ liệu ngoại vi với AmiBroker, bạn sẽ cần thực hiện một thiết lập một lần được mô tả dưới đây:

- Chạy AmiBroker
- Chọn File->New database
- Nhập một tên thư mục mới (ví dụ: C:\Program Files\AmiBroker\NewData) và nhấp vào Create như trong ảnh:



- Chọn mục phù hợp từ combo Data source:
    - **Người dùng Quotes Plus** chọn **"Quotes Plus plug-in"** làm **Data Source** và "**Disable**" từ **Local data storage**
    - **Người dùng TC2000/TCNet** chọn **"TC2000/TCNet plug-in"** làm **Data Source** và **"Enable"** từ **Local data storage**
    - **Người dùng TC2000 cho Mutual Funds** chọn **"TC2000 Mutual Funds plug-in**" làm **Data Source** và "**Enable**" từ **Local data storage**

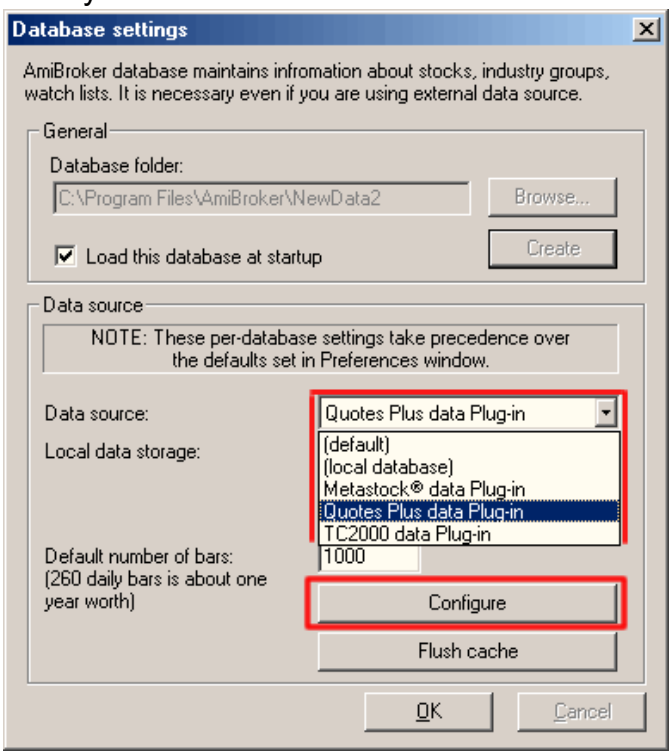
- **Người dùng TC2005** chọn "**TC2000/TCNet plug-in**" làm **Data Source** và "**Enable**" từ **Local data storage**

  Lưu ý: Người dùng TC2005 có thể cần **tuân theo hướng dẫn này (nhấn vào đây)** nếu plugin TC2000 không xuất hiện.

- **Người dùng FastTrack** chọn **"FastTrack plug-in**" làm **Nguồn Dữ liệu** và chọn **"Vô hiệu hóa"** từ **Lưu trữ dữ liệu Cục bộ**

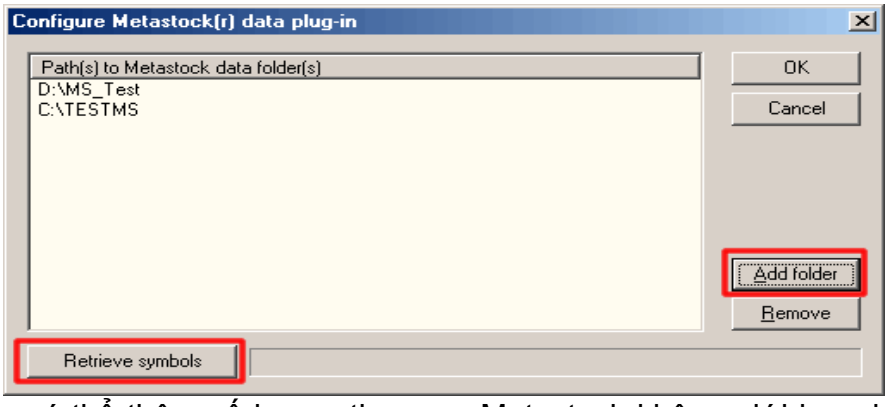
- **Người dùng Metastock** chọn "**Metastock plug-in**" làm **Nguồn Dữ liệu** và chọn "**Vô hiệu hóa**" từ **Lưu trữ dữ liệu Cục bộ**

- Nhấp vào nút Cấu hình để hiển thị hộp thoại cấu hình plugin như hình dưới đây.



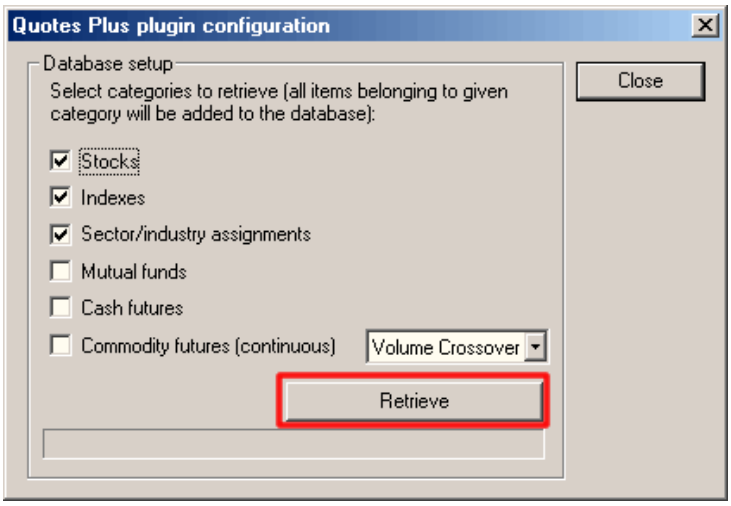
- Chỉ áp dụng cho Metastock plug-in (bỏ qua điểm này trong trường hợp của TC2000, Quotes Plus, FastTrack):

- Nhấn vào nút "Thêm thư mục" để thêm thư mục cơ sở dữ liệu Metastock làm nguồn dữ liệu của bạn (tìm kiếm tập tin MASTER của Metastock và nhấn OK) như hình dưới đây:



- Bạn có thể thêm số lượng thư mục Metastock không giới hạn, hiệu quả vượt qua giới hạn 4096 ký hiệu của MS.

- Nhấn nút **Retrieve** - thao tác này sẽ thiết lập một cơ sở dữ liệu mới với tất cả các ký hiệu và tên đầy đủ. Plugins Quotes Plus và TC2000 cũng sẽ thiết lập tên và phân công của lĩnh vực/ngành của bạn, như hình dưới



Từ bây giờ, AmiBroker của bạn đọc trực tiếp các báo giá từ nguồn dữ liệu ngoài. Không cần nhập/cập nhật báo giá nữa. **Tất cả các báo giá mới sẽ xuất hiện tự động mà không cần can thiệp của người dùng.**

QUAN TRỌNG: Nếu có **ký hiệu mới** được thêm hoặc **ký hiệu cũ** bị xóa khỏi nguồn dữ liệu ngoài, bạn sẽ cần đi tới File->Database Settings->Configure và bấm "RETRIEVE" một lần nữa để lấy các ký hiệu mới.

Ghi Chú Về Hiệu Suất Plugin

Sử dụng cơ sở dữ liệu gốc của AmiBroker sẽ đem lại hiệu suất tốt nhất (mất ít hơn 2 mili giây để lấy 1000 dữ liệu).

Plugin Metastock cũng khá nhanh, vì nó có thể lấy 1000 dữ liệu trong khoảng 6-7 mili giây (bao gồm việc tìm kiếm ký hiệu trong 5 thư mục khác nhau). Thực tế, AmiBroker có thể truy cập dữ liệu Metastock nhanh hơn chính Metastock :-)

Hiệu suất của Quotes Plus phụ thuộc vào nhiều yếu tố - lần truy cập đầu tiên có thể chậm hơn nhiều (0.1-0.2 giây cho 1000 dữ liệu) nhưng các lần truy cập sau nhanh hơn (xuống còn 5 mili giây). Plugin FastTrack nhanh chóng như plugin Quotes Plus.

TC2000 không nhanh bằng, đặc biệt nếu bạn chỉ sử dụng dữ liệu từ CD. Vì vậy, bạn nên sao chép cơ sở dữ liệu của mình tới ổ đĩa cứng để có hiệu suất tốt hơn. Nhưng vẫn vậy, ngay cả khi chỉ sử dụng dữ liệu từ CD, AmiBroker có thể truy cập 1000 dữ liệu từ TC2000 trong khoảng 0.25 giây (lần truy cập đầu tiên) và 0.015 giây (các lần truy cập sau). Ngoài ra, cũng nên kích hoạt "Local data storage" khi sử dụng plugin TC2000 vì nó sẽ tăng tốc đáng kể (>10 lần) (khi bạn truy cập dữ liệu TC2000, AmiBroker sẽ lưu chúng vào cơ sở dữ liệu gốc của mình để lấy dữ liệu nhanh hơn).

Thời gian là tương đối và không bao gồm quá trình khởi tạo plugin one-time. Các phép đo được thực hiện trên máy tính dựa trên Celeron 600 với RAM 196KB và CD-ROM 24x

Bộ Nhớ Cache Trong Bộ Nhớ

Theo mặc định, AmiBroker chỉ giữ dữ liệu của 10 ký hiệu được truy cập gần đây nhất trong RAM. Điều này chiếm khoảng 320 KB (đúng, kilobyte) bộ nhớ cho mỗi ký hiệu với 1000 thanh dữ liệu được tải. Bạn có thể tăng "Bộ nhớ cache trong" (Tools -> Preferences : tab "Data") lên 100 (khoảng 3,2MB tiêu thụ RAM bổ sung) hoặc 1000 (khoảng 32MB tiêu thụ RAM bổ sung) hoặc thậm chí nhiều hơn để cải thiện hiệu suất khi truy cập dữ liệu sau này (một khi dữ liệu đã có trong RAM, AmiBroker không cần phải yêu cầu plugin nhiều lần).

Hướng Dẫn Tự Động Cập Nhật Trích Dẫn US Bằng Amiquote

Bắt Đầu Nhanh

Chạy AmiBroker

Chọn **Công cụ-> Tự động cập nhật trích dẫn (Mỹ & Canada)**

Cách Thức Hoạt Động

AmiQuote tải (hoặc truy xuất từ AmiBroker) một tệp danh sách ticker (.TLS) đơn giản là tệp ASCII với các biểu tượng ticker, sau đó phân tích nó và tạo ra các URL đến trang tài chính Yahoo! dựa trên tên ticker, chế độ (trích dẫn hiện tại hoặc lịch sử), quốc gia và từ / đến ngày. Sau đó, khi bạn bắt đầu quá trình tải xuống, nó yêu cầu dữ liệu từ Yahoo và lưu trữ dữ liệu đã tải xuống trong các tệp .AQD (hàng ngày) hoặc .AQH (lịch sử) riêng biệt cho mỗi ticker. Sau khi tải xuống, nếu AmiBroker đang chạy, AmiQuote sẽ tự động nhập các trích dẫn vào AmiBroker.

Cách Sử Dụng

Cập nhật tự động

Phương pháp dễ nhất để làm việc với AmiQuote và AmiBroker là sử dụng thủ tục được đưa ra trong mục **Bắt đầu nhanh** của tài liệu này. Chỉ cần chạy AmiBroker và AmiQuote và chọn **Công cụ -> Tự động cập nhật cơ sở dữ liệu AmiBroker.** Phương pháp này cập nhật trích dẫn lịch sử từ ngày cuối cùng hiện diện trong AmiBroker đến ngày hôm nay. Khi thực hiện cập nhật tự động, AmiQuote thực hiện 4 bước:

a) truy xuất danh sách ticker từ AmiBroker (tất cả các biểu tượng hiện được tải trong AmiBroker);

b) nhận ngày trích dẫn cuối cùng có sẵn trong AmiBroker;

c) thực hiện tải xuống lịch sử từ ngày cuối đến hôm nay;

d) hướng dẫn AmiBroker để nhập các tệp đã tải xuống.

Xin hãy chú ý rằng quy trình này chỉ hoạt động cho thị trường Mỹ & Canada, vì Yahoo chỉ cung cấp trích dẫn lịch sử cho những thị trường đó.

Lưu ý rằng AmiQuote hiện hỗ trợ một tham số dòng lệnh mới: /autoupdate. Tùy chọn này buộc AmiQuote thực hiện quy trình cập nhật tự động mà không cần sự can thiệp của người dùng. Theo mặc định, menu công cụ của AmiBroker được cấu hình như sau:

C:\Program Files\AmiBroker\AmiQuote\Quote.EXE /autoupdate

Vì vậy, bạn có thể cập nhật cơ sở dữ liệu Mỹ của mình chỉ với một cú nhấp vào Công cụ-> Tự động cập nhật trích dẫn (Mỹ & Canada) trong Hướng dẫn sử dụng AmiBroker.

Hoạt động thủ công

Chế độ tự động rất tiện lợi, nhưng có những trường hợp bạn cần thực hiện một số tác vụ thủ công. Có một tài liệu cũ mô tả cách hoạt động này tại: Cách tải dữ liệu báo giá thủ công bằng AmiQuote. Tất cả những gì viết trong tài liệu này vẫn còn đúng, với một ngoại lệ - bây giờ việc nhập dữ liệu vào AmiBroker được thực hiện tự động nếu bạn đã đánh dấu vào **ô nhập tự động**.

Ngoài ra, cũng có một số trường hợp bạn muốn thực hiện thủ công, vì vậy đừng quên các công cụ hữu ích có sẵn trong tầm tay bạn:

File -> Mở (Open), File -> Save, File -> Lưu như (Save As)

Những chức năng này giúp bạn tải và lưu lại danh sách các biểu đồ được chỉnh sửa cho việc sử dụng lặp lại trong tương lai.

Edit -> Thêm Tickers (Add Tickers)

Chức năng này cho phép bạn thêm các biểu đồ vào danh sách. Chỉ cần gõ các biểu đồ cần thêm vào trường sẽ xuất hiện khi bạn chọn chức năng này.

Edit -> Xóa Tickers (Delete Tickers)

Chức năng này cho phép bạn xóa các biểu đồ khỏi danh sách. Chỉ cần chọn các mục bạn muốn xóa từ chế độ xem danh sách (có thể chọn nhiều mục bằng cách giữ phím SHIFT hoặc CTRL khi nhấp vào các mục), sau đó chọn chức năng này.

Edit -> Đánh dấu Tất cả (Mark All), Edit -> Bỏ Đánh dấu Tất cả (Unmark All), Edit -> Chuyển đổi (Toggle), Edit -> Đánh dấu Lựa chọn (Mark Selection), Edit -> Bỏ Đánh dấu Lựa Chọn (Unmark Selection)

Các chức năng này cho phép bạn đánh dấu các biểu đồ cần tải về. Lưu ý rằng AmiQuote sẽ đánh dấu trước tên mỗi biểu đồ trong danh sách. CHỈ CÓ các mục ĐÃ ĐÁNH DẤU mới được tải về. Điều này cho phép bạn thực hiện việc tải về/ cập nhật theo lựa chọn.

View -> Làm mới (Refresh)

Cơ bản, AmiQuote tự điều khiển việc làm mới khi cần thiết. Ví dụ, nếu bạn thay đổi khoảng thời gian, danh sách sẽ được làm mới trước khi bắt đầu tải về. Tuy nhiên, có những trường hợp bạn có thể muốn làm mới danh sách theo cách của bạn. Chẳng hạn, nếu bạn tải về và nhập dữ liệu một lần và muốn làm điều này một lần nữa, bạn cần chọn chức năng này. Chức năng Làm mới đơn giản chỉ áp dụng tất cả các cài đặt về ngày và loại cho các URL được liệt kê và ĐÁNH DẤU tất cả các biểu đồ cho lượt tải về mới.

Tools -> Nhập vào Amibroker (Import Into Amibroker)

Chức năng này hữu ích nếu bạn muốn nhập các báo giá vừa tải về vào AmiBroker nhưng bạn đã bỏ chọn hộp **Automatic import**.

Tools -> Lấy Tickers từ Amibroker (Get Tickers From Amibroker)

Chức năng này lấy tất cả các biểu ngữ từ cơ sở dữ liệu AmiBroker hiện tại đã tải lên và điền danh sách biểu đồ AmiQuote với chúng.

Tools -> Lấy Ngày Cập nhật Cuối cùng (Get Last Update Date)

Chức năng này lấy ngày của bảng báo giá gần đây nhất của biểu tượng đầu tiên hiện diện trong cơ sở dữ liệu AmiBroker hiện tại đã tập hợp và đặt ngày **From** cho ngày này.

Tools -> Cài đặt (Settings)

Hiển thị cửa sổ cài đặt nơi bạn có thể xác định thư mục đích để lưu trữ tất cả các tệp tải về. Lưu ý rằng nếu thư mục đích để trống, các tệp tải về sẽ được lưu trữ trong thư mục làm việc hiện tại (trong hầu hết các trường hợp, đây là thư mục từ nơi tệp .TLS hiện tại được tải lên).

Trong cửa sổ này, bạn cũng có thể thay đổi chế độ ghi các tệp. Theo mặc định, các tệp lịch sử sẽ bị ghi đè trong khi các tệp hàng ngày sẽ được nối thêm. Đây là cài đặt được đề xuất. Việc nối thêm các tệp hàng ngày đơn giản cho phép bạn tạo các tệp lịch sử trong ngày khi bạn thực hiện các cập nhật hàng ngày. Bạn có thể thay đổi hành vi này theo mục đích cụ thể của mình.

Hướng Dẫn Tải Dữ Liệu Báo Giá Thủ Công Sử Dụng Amiquote

Giới thiệu

Mục đích của tài liệu này là giải thích cách sử dụng AmiQuote và AmiBroker để nhận dữ liệu báo giá từ các trang Yahoo Finance và Quote.com. AmiQuote là một chương trình đồng hành với phần mềm phân tích biểu đồ AmiBroker. Mục đích chính của AmiQuote là đơn giản hóa và tự động hóa việc tải dữ liệu báo giá hàng ngày và lịch sử từ Yahoo! Finance (Mỹ, các sàn giao dịch chính ở châu Âu và một số quốc gia khác), Quote.com (chỉ Mỹ), MSN (Mỹ và một số sàn châu Âu), Integratir (cổ phiếu Mỹ), Forex (trang không giới hạn của Finam)

Yahoo cung cấp dữ liệu ở hai chế độ "Lịch Sử" và "Hiện Tại" của AmiQuote. Quote.com cung cấp dữ liệu ở chế độ "Trong ngày" của AmiQuote.

Chuẩn Bị Danh Sách Mã Chứng Khoán

Danh sách mã chứng khoán là một tệp văn bản đơn giản liệt kê từng dòng mã bạn muốn nhập. Tệp danh sách mã chứng khoán của AmiQuote có phần mở rộng .TLS. AmiQuote đi kèm với danh sách mã chứng khoán được viết sẵn cho các thành phần của chỉ số chính NYSE và NASDAQ và một số chỉ số / thị trường châu Âu. Các danh sách mã chứng khoán bổ sung có sẵn trên trang khởi động tại: **http://www.amibroker.com/starter/**. Bạn có thể sử dụng các danh sách mã chứng khoán được viết trước hoặc bạn có thể tùy chỉnh hoặc viết danh sách riêng của mình. Để chỉnh sửa tệp .TLS hiện có hoặc viết một cái hoàn toàn mới, tất cả những gì bạn cần là một trình soạn thảo văn bản thuần túy như Notepad hoặc bất kỳ trình biên tập ASCII thuần túy khác (không phải MS Word!). Tất cả những gì bạn phải làm là viết mã chứng khoán bạn muốn nhập từng dòng (mỗi dòng một mã) và lưu tệp.Hãy chắc chắn rằng bạn đang lưu tệp với phần mở rộng .TLS. Nếu không, AmiQuote sẽ không tải tệp này.

Xin lưu ý rằng Yahoo sử dụng các hậu tố cho các cổ phiếu không phải của Mỹ. Vì vậy, để lấy báo giá cho biểu tượng không phải của Mỹ, bạn sẽ cần thêm hậu tố thích hợp vào mã chứng khoán. Các hậu tố theo thứ tự chữ cái là:.AS - Amsterdam, .AX - Australia (ASX), .BC - Barcelona, .BE - Berlin, .BO - Bombay, .BM - Breman, .BR - Brussels, .BA - Buenos Aires, .CL - Calcuta, .CR - Caracas, .V - CDNX, .CO - Copenhagen, .D - Dusseldorf, .F - Frankfurt, .H - Hamburg, .HA - Hanover, .HK - Hong Kong, .I - Ireland, .JK - Jakarta, .KA - Karachi, .KQ - Kosdaq, .KS - KSE, .KL - Kuala Lumpur, .L - London, .LM- Lima, .LS - Lisbon, .MA - Madrid, .MX - Mexico, .MI - Milan, .MU - Munich, .NS - NSE, .NZ - New Zeland, .OL - Oslo, .PA - Paris , .SN - Santiago, .SS - Shanghai, .SZ Shenzhen, .ST - Stockholm, .SG - Stutgart, .TW- Taiwan, .TA - Tel Aviv, .TO - Toronto, .VA - Valencia, .VI - Viena, .DE - XETRA, .S - Zurich.

Xin lưu ý rằng Yahoo và Quote.com cũng sử dụng các biểu tượng khác nhau cho các chỉ số. Sự khác biệt chính là Yahoo sử dụng ^ (dấu gạch ngang) tiền tố và Quote.com sử dụng $ (dấu hiệu đô la) tiền tố.

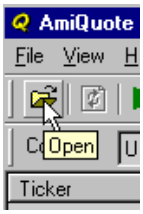
Để xem danh sách các chỉ số được cung cấp bởi **Yahoo**, vui lòng nhấn đây

Để xem danh sách các chỉ số do **Lycos/Quote.com** cung cấp, vui lòng nhấp vào đây. Xin lưu ý rằng gần đây Lycos/Quote.com đã ngừng cung cấp báo giá miễn phí và bạn

cần có đăng ký Livecharts (9,95 USD/tháng) để sử dụng dịch vụ này. Để biết thêm chi tiết, hãy xem bài viết trong Kho kiến thức này.

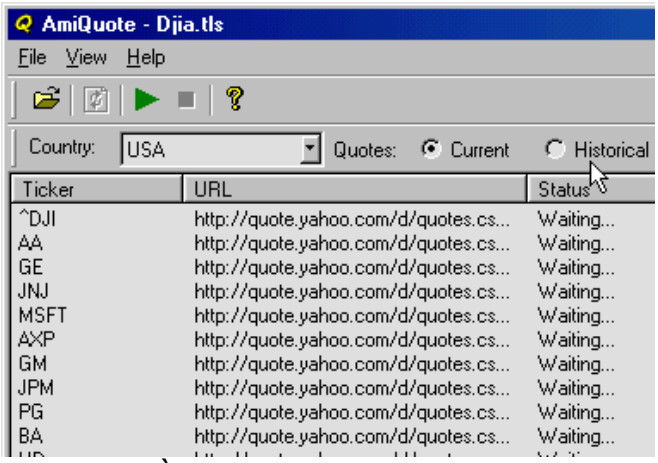
Để xem danh sách các ký hiệu do **MSN** cung cấp, vui lòng nhấp vào đây.

Tải Dữ liệu



Để tải dữ liệu hãy mở AmiQuote. Sau đó hãy nhấn vào nút "Mở" trên thanh công cụ (hoặc chọn **File->Open Menu**) như hình bên phải.

Từ hộp thoại tệp hãy chọn một tệp .TLS (ví dụ: DIJA.TLS) và nhấn nút Mở. Sau đó bạn sẽ thấy màn hình chính của AmiQuote được điền đầy danh sách các mã chứng khoán đã tải, như hình dưới đây.



Chọn Nguồn Dữ liệu phù hợp.

- **Yahoo Historical** - cho phép bạn tải lịch sử cuối ngày đến ngày hiện tại (dữ liệu ngày hiện tại xuất hiện vài giờ sau phiên kết thúc)

- Yahoo Current - cho phép bạn tải báo giá ngày hiện tại (trễ 15 phút) trong phiên giao dịch

- **Lycos/Quote.com Intraday** - cho phép bạn tải dữ liệu lịch sử intraday và hàng ngày (1-phút bars và trở lên) - chỉ cho cổ phiếu/futures của Mỹ. Nếu bạn đã chọn chế độ này, bạn cũng nên chọn khoảng thời gian bar (xem các hạn chế được mô tả dưới đây) - cần đăng ký Livecharts ($9.95/tháng)

- **MSN Historical** - cho phép bạn tải lịch sử cuối ngày đến ngày hiện tại (dữ liệu ngày hiện tại xuất hiện vài giờ sau phiên kết thúc)

- **Forex** - cho phép bạn tải lịch sử cuối ngày và intraday (phiên bản đã đăng ký) cho các cặp tiền tệ sau: EURCHF, EURGBP, EURJPY, EURUSD, GBPUSD, USDCHF, USDJPY.

Sau khi chọn đúng các tùy chọn, vui lòng nhấp vào mũi tên màu xanh (hoặc sử dụng **File -> Start Download** menu). Quá trình tải xuống sẽ bắt đầu. AmiQuote sẽ hiển thị các thông điệp tiến trình và thông tin trạng thái bao gồm số lượng tải xuống đã hoàn thành và số lượng tệp còn lại. Bất cứ lúc nào bạn cũng có thể dừng quá trình tải xuống với nút "dừng" (hộp màu đỏ). Sau khi hoàn thành việc tải xuống, **AmiQuote sẽ tự**

động cập nhật các báo giá trong AmiBroker (nếu chỉ AmiBroker đang chạy song song và hộp "nhập tự động" trong AmiQuote được tích).

Hạn chế

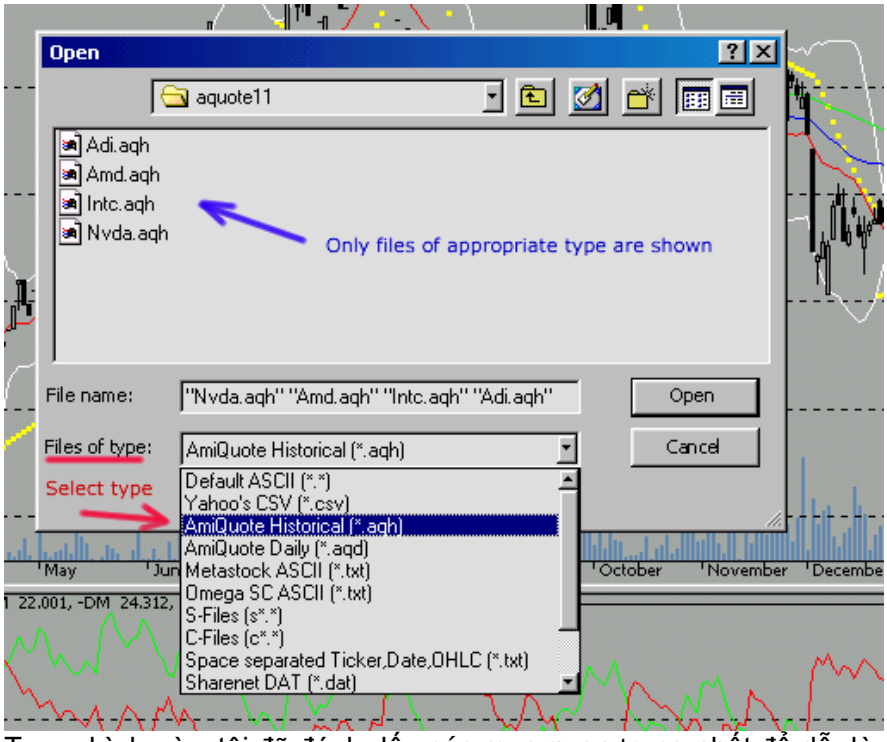
Dữ liệu bar khoảng thời gian intraday (1-phút, 5-phút, 15-phút, 60-phút và 120-phút) chỉ có sẵn cho các chứng chỉ tài sản của Mỹ. Dữ liệu lịch sử cho các sàn giao dịch quốc tế thường ngắn hơn nhiều so với thị trường Mỹ.

Bởi vì dữ liệu bar intraday được tải từ máy chủ của Quote.com nên các ký hiệu chứng khoán cho chỉ số khác nhau so với những ký hiệu được dùng bởi Yahoo. Để tham khảo đầy đủ, vui lòng kiểm tra tại http://finance.lycos.com/home/misc/symbol_search.asp?options=i

Dữ liệu thanh trong ngày bị giới hạn ở mức 500 thanh, bất kể khoảng thời gian của thanh. Nói cách khác, bạn luôn nhận được dữ liệu 500 thanh, bất kể lúc đó là dữ liệu 1 phút, 5 phút, 15 phút, 60 phút hay 120 phút - vì vậy, bằng cách chọn khoảng thời gian lớn hơn, bạn sẽ nhận được dữ liệu từ nhiều ngày hơn. Đây là giới hạn do máy chủ Livecharts đặt ra.

Nhập trích dẫn vào AmiBroker

LƯU Ý: Bước này không còn cần thiết nếu bạn sử dụng tính năng "nhập tự động" của AmiQuote. Các giải thích chỉ được cung cấp cho người dùng muốn nhập một cách lựa chọn hoặc nhập lại các tệp đã tải về trong quá khứ**.**
Đầu tiên, hãy khởi động AmiBroker. Trong menu File, vui lòng chọn Nhập từ ASCII. Bạn sẽ thấy hộp thoại tệp tiếp theo:



Trong hình này, tôi đã đánh dấu các mục quan trọng nhất để dễ dàng xác định. Được đánh dấu bằng màu đỏ là hộp combo chọn kiểu ("b). Để nhập các tệp AmiQuote (những tệp có

phần mở rộng .AQH và .AQD, bạn nên chọn AmiQuote History hoặc AmiQuote Daily, hoặc AmiQuote Intraday (.AQI) hoặc AmiQuote MSN (.AQM) hoặc AmiQuote eSignalCentral (.AQE) từ hộp combo (mũi tên màu đỏ cho thấy những tùy chọn đó).

Sau khi chọn kiểu phù hợp, bạn sẽ chỉ thấy các tệp của kiểu phù hợp trong danh sách tệp (mũi tên màu xanh dương cho thấy điều đó). Bây giờ bạn có thể chọn một hoặc nhiều tệp từ danh sách. Chọn nhiều tệp cùng một lúc có thể thực hiện bằng cách giữ phím CTRL trong khi chọn các mục bằng chuột (bạn cũng có thể nhấn SHIFT để chọn một phạm vi các tệp với một lần nhấp chuột). Giờ khi bạn đã chọn xong các tệp muốn nhập, chỉ cần nhấp vào nút "Mở". Quá trình nhập sẽ bắt đầu và bạn sẽ thấy thanh tiến trình cho thấy AmiBroker đang nhập dữ liệu. Sau khi kết thúc quá trình nhập, AmiBroker sẽ tự động làm mới danh sách biểu tượng và bạn sẽ thấy các dấu và biểu đồ được cập nhật. Nếu có gì đó không đúng trong quá trình nhập, AmiBroker sẽ ghi một tệp nhật ký gọi là "import.log" và nằm trong thư mục chính của AmiBroker. Bạn có thể xem tệp nhật ký này nếu muốn tìm hiểu vấn đề gì đã xảy ra (Vì import.log là tệp văn bản đơn giản, bạn có thể mở nó bằng bất kỳ trình chỉnh sửa văn bản nào)

Các câu hỏi thường gặp

Câu hỏi	Trả lời
Làm thế nào để chỉnh sửa danh sách ticker của tôi (.TLS)?	Bạn có thể tạo hoặc chỉnh sửa .TLS bằng Windows Notepad. Khi lưu một tập tin, chỉ cần đặt phần mở rộng .TLS cho tập tin (thay vì .TXT mặc định).
Về các danh sách ticker hoàn chỉnh sẵn có cho NYSE, NASDAQ, AMEX?	Sau đây là các danh sách ticker sẵn có để tải về: - DJIA.TLS (30 cổ phiếu) - DJTA.TLS (20 cổ phiếu) - DJUA.TLS (15 cổ phiếu) - NASDAQ100.TLS (100 cổ phiếu) - NYSE.TLS (2612 cổ phiếu) - NASDAQ.TLS (4464 cổ phiếu) - AMEX.TLS (794 cổ phiếu)

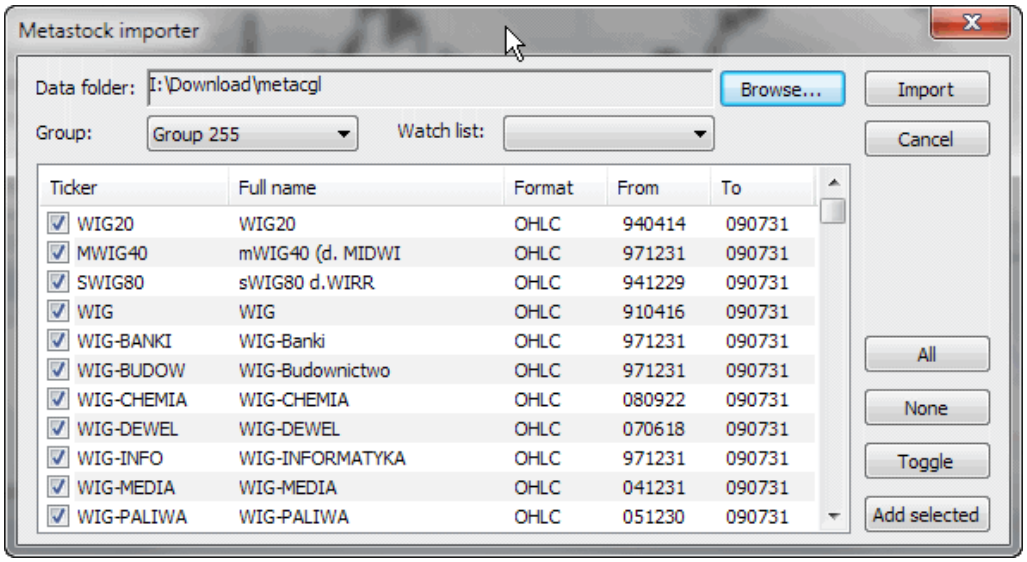
Thông tin khác

Để biết thêm thông tin, vui lòng tham khảo phần "Quản lý dữ liệu - Nhập dữ liệu từ tập tin ASCII" trong Hướng dẫn sử dụng AmiBroker. Trong trường hợp có bất kỳ câu hỏi, bình luận hay đề xuất nào khác, vui lòng sử dụng trang hỗ trợ khách hàng tại http://www.amibroker.com/support.html

Cửa sổ nhập khẩu Metastock.

LƯU Ý QUAN TRỌNG: Trình nhập khẩu Metastock chỉ nên được sử dụng nếu bạn muốn nhập dữ liệu MS vào cơ sở dữ liệu AmiBroker cục bộ một lần. Nếu bạn muốn AmiBroker đọc TRỰC TIẾP cơ sở dữ liệu Metastock mà không cần nhập dữ liệu mới nhiều lần, hãy thiết lập cơ sở dữ liệu của bạn với PLUGIN METASTOCK như đã mô tả trong Hướng dẫn.

LƯU Ý 2: Nếu bạn đã thiết lập cơ sở dữ liệu của mình với plugin MS, bạn KHÔNG nên sử dụng trình nhập khẩu Metastock, vì sẽ không có lý do để sử dụng nó khi dữ liệu của bạn đã được cung cấp bởi plugin.



Trình nhập khẩu Metastock mở ra cho AmiBroker một nguồn dữ liệu lịch sử phong phú. Trình nhập khẩu hỗ trợ cả định dạng Metastock cũ 6.5 và định dạng mới 7.x (XMASTER).

Cơ bản, dữ liệu Metastock bao gồm:

- Tệp MASTER/EMASTER chứa thông tin chung về các mã chứng khoán, tên cổ phiếu, v.v.

- Các tệp F1.DAT, F2.DAT,..., Fxx.DAT chứa dữ liệu báo giá thực tế.

Tệp MASTER/EMASTER rất quan trọng vì nó chứa tham chiếu đến các tệp Fxx.DAT. Các tệp Fxx.DAT chỉ lưu trữ báo giá theo định dạng 5 trường (ngày/cao/thấp/đóng/khối lượng), 6 hoặc 7 trường (ngày/mở/cao/thấp/đóng/khối lượng/lợi tức mở). Như bạn thấy, các tệp MASTER/EMASTER và Fxx.DAT có liên kết chặt chẽ và bạn cần tất cả chúng để nhập dữ liệu.

Cách sử dụng:

Để nhập dữ liệu Metastock, bạn cần thực hiện các bước sau:

- Chọn nhập khẩu Metastock từ menu.

- Sử dụng trình chọn thư mục (Browse...) để chọn vị trí dữ liệu ở định dạng Metastock (thư mục chứa các tệp MASTER/EMASTER và Fxx.DAT).
- Sau khi chọn thư mục phù hợp, AmiBroker sẽ hiển thị danh sách các ký hiệu và khoảng thời gian có sẵn. Theo mặc định, tất cả các ký hiệu có sẵn sẽ được đánh dấu để nhập (đánh dấu ở đầu danh sách).Bạn có thể loại trừ một số ký hiệu khỏi danh sách nhập bằng cách nhấp vào mục thích hợp trong danh sách (đánh dấu sẽ chuyển đổi khi bạn nhấp).
- Bạn có thể quyết định nhóm và danh sách theo dõi mà các biểu tượng mới sẽ được thêm vào bằng cách sử dụng Group và Watch List combos.
- Sau khi thực hiện lựa chọn của mình, nhấn nút 'Import' để bắt đầu quá trình nhập.
- Trong quá trình đó, bạn có thể hủy bỏ thao tác bằng cách nhấp vào nút 'Abort' trong cửa sổ tiến trình.

Hiểu về cơ sở dữ liệu Amibroker

Giới thiệu

Một ứng dụng Windows tiêu chuẩn, ví dụ như Paint, chỉ làm việc với MỘT file. Bạn chỉ cần mở và lưu file đó (.BMP trong Paint, hoặc .DOC trong MS WORD), và file đó chứa tất cả thông tin cần thiết.

Tuy nhiên, AmiBroker là một phần mềm phức tạp hơn. Nó sử dụng lượng dữ liệu lớn (tất cả báo giá từ các loại ticker khác nhau, các nghiên cứu được vẽ bằng tay, phân chia thành các nhóm, thị trường, danh sách giám sát, các mục ưa thích, ngành công nghiệp, các lĩnh vực, v.v.), vì vậy nó phải quản lý nhiều file.

Trên thực tế, có thể lưu tất cả thông tin này vào một file duy nhất, nhưng nếu vậy, file sẽ (a) rất lớn, và (b) mất thời gian để cập nhật một cách chọn lọc. Vì vậy, AmiBroker sử dụng nhiều file để lưu trữ tất cả dữ liệu. Có rất nhiều file được liên kết với bất kỳ cơ sở dữ liệu nào. Các file cho một cơ sở dữ liệu cụ thể được lưu trữ trong một thư mục (và các thư mục con của nó) dành riêng cho cơ sở dữ liệu đó.Trong tài liệu AmiBroker, thư mục đó được gọi là "thư mục cơ sở dữ liệu" (phiên bản 3.9 trở đi) hoặc "thư mục workspace" (phiên bản AmiBroker trước đó).

Khi bạn cài đặt AmiBroker lần đầu tiên, một thư mục cơ sở dữ liệu mặc định được tạo ra, được gọi là 'data', trong thư mục AmiBroker. Thư mục cơ sở dữ liệu này chứa một cơ sở dữ liệu Dow Jones Industrial Average mẫu.

Trong menu và các lựa chọn hộp thoại cơ sở dữ liệu của AmiBroker, bạn đang chọn hoặc tạo một thư mục cơ sở dữ liệu, không phải một tệp cá nhân.

Cấu trúc cơ sở dữ liệu Amibroker

Một cơ sở dữ liệu (hoặc một workspace) là một thư mục chứa một tập hợp các file nhị phân, được lưu trữ trong các thư mục con 0-9, a-z, '_'. Những file nhị phân này chứa báo giá, thông tin biểu tượng, các nghiên cứu của bạn (đường xu hướng, số Fibonacci). Thông tin của mỗi biểu tượng được lưu trữ riêng biệt trong file có tên biểu tượng ticker nằm trong thư mục con tương ứng với ký tự đầu tiên của biểu tượng, vì vậy dữ liệu / nghiên cứu báo giá IBM được lưu trữ trong file 'IBM' nằm trong thư mục con 'I'.

Cơ sở dữ liệu mặc định cho AB là thư mục 'data'. Nó chứa dữ liệu mẫu DJIA. Bạn có thể tạo thêm các cơ sở dữ liệu trong các thư mục khác thông qua menu File -> "New database".

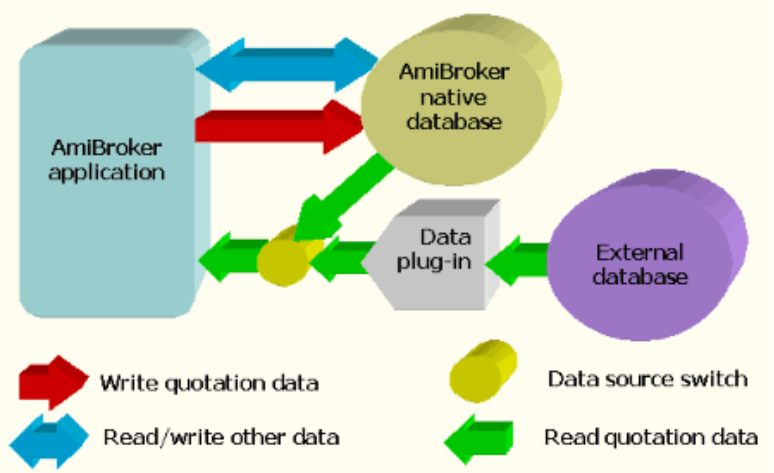
Ngoài các thư mục và tệp con này, AmiBroker cũng tạo ra hai tệp bổ sung: broker.workspace và broker.master. Tệp đầu tiên được sử dụng để lưu trữ tên danh mục và thông tin về các vấn đề tăng/giảm/không thay đổi. Tệp thứ hai lưu trữ bảng tất cả các ký hiệu, được sử dụng để tải nhanh cơ sở dữ liệu.Hai tệp này nằm trong thư mục gốc của mỗi cơ sở dữ liệu, trong thư mục "data" theo mặc định.

Trong hầu hết các trường hợp, bạn KHÔNG nên chạm vào các tệp trong cơ sở dữ liệu AmiBroker, vì chương trình quản lý chúng tự động và không cần sự can thiệp của người dùng.

Dữ liệu dạng External?

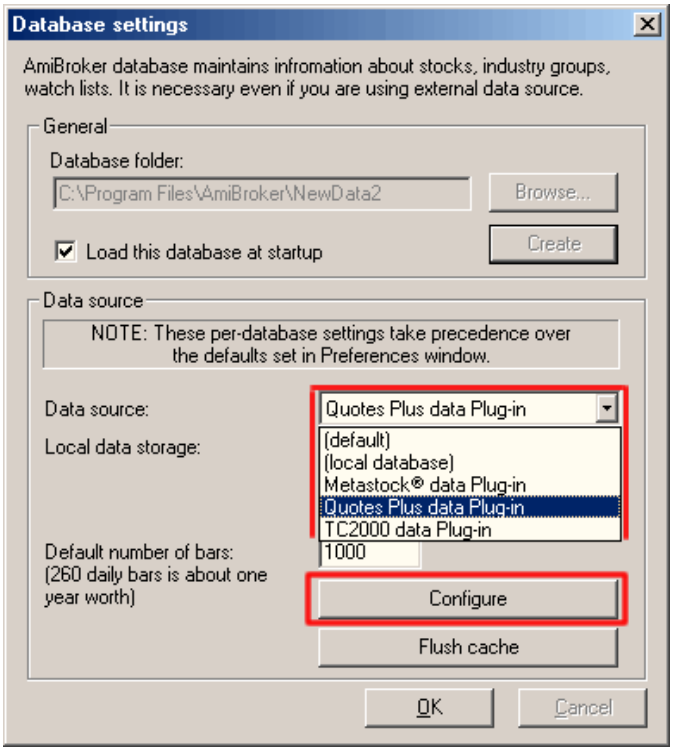
AmiBroker 3.9 có khả năng đọc báo giá TRỰC TIẾP từ một nguồn dữ liệu bên ngoài. Hiện tại, AmiBroker có thể đọc trực tiếp từ cơ sở dữ liệu Quotes Plus (QP2), TC2000 (TC2K) và Metastock (MS). Điều này được thực hiện thông qua các plugin dữ liệu mà AmiBroker sử dụng để đọc dữ liệu từ nguồn dữ liệu bên ngoài. Khi người dùng quyết định muốn sử dụng một cơ sở dữ liệu bên ngoài thay vì đọc báo giá từ cơ sở dữ liệu của chính mình, AmiBroker sẽ chỉ cần yêu cầu plugin cung cấp báo giá cho bất kỳ ký hiệu nào. Plugin sẽ đọc cơ sở dữ

liệu bên ngoài và cung cấp dữ liệu cho AmiBroker. Toàn bộ quá trình được minh họa trong hình dưới đây:



Như bạn thấy, các plugin dữ liệu cung cấp quyền truy cập chỉ đọc vào báo giá trong cơ sở dữ liệu bên ngoài. Điều này có nghĩa là các nguồn dữ liệu bên ngoài của bạn sẽ không bao giờ bị sửa đổi bởi AmiBroker. Các thay đổi hoặc bổ sung mà bạn thực hiện đối với dữ liệu và biểu đồ (như các nghiên cứu vẽ tay, phân loại, v.v.) sẽ luôn được lưu trong cơ sở dữ liệu của AmiBroker. Vì vậy, AmiBroker vẫn sử dụng cơ sở dữ liệu riêng của mình (để lưu các thay đổi, như một bộ nhớ cache để tăng tốc độ truy cập và cho các tác vụ khác), ngay cả khi sử dụng nguồn dữ liệu bên ngoài.

Công tắc Nguồn dữ liệu được biểu diễn trong hình trên có thể được người dùng thiết lập để truy cập các cơ sở dữ liệu bên ngoài khác nhau. Các nguồn dữ liệu bên ngoài được chọn bằng cách vào hộp thoại File -> "Database settings", như được hiển thị dưới đây.



Bạn cũng có thể chọn lưu trữ các báo giá lấy từ nguồn bên ngoài vào cơ sở dữ liệu của AmiBroker để truy xuất nhanh hơn trong các lần truy cập tiếp theo. Nếu bạn muốn làm điều này, bạn nên chuyển cài đặt "Local data storage" sang "Enabled".

Chú ý: Các thiết lập tương tự có thể được tìm thấy tại Tools->Preferences trong tab 'Data'. Tuy nhiên, những thiết lập này chỉ là mặc định được sử dụng khi tạo cơ sở dữ liệu mới. Các cấu hình trong File->Database Settings luôn được ưu tiên so với những cấu hình đã được thực hiện trong Preferences -- TRỪ trong những trường hợp sau: Nếu bạn chọn mục 'Default' trong danh sách drop down cùng tên (được hiển thị ở trên), hoặc nút radio 'Default' cho Local Data Storage (cũng được hiển thị ở trên), AmiBroker sẽ sử dụng các thiết lập trong Preferences cho những mục đó.

Hiểu về Danh mục

AmiBroker có khả năng phân loại các biểu tượng vào các danh mục khác nhau, cho phép bạn (khi được cài đặt đúng cách) thu hẹp các tìm kiếm phân tích của mình đến các biểu tượng đáp ứng một số tiêu chí lựa chọn nhất định (nhờ tính năng lọc có sẵn trong cửa sổ Đánh giá nhanh và Phân tích tự động). Việc thiết lập ban đầu của các danh mục có thể hơi phức tạp, đặc biệt khi bạn muốn theo dõi hàng nghìn biểu tượng.



Các danh mục xuất hiện trong cửa sổ Symbols. Điều quan trọng nhất để nhớ là danh mục KHÔNG hoạt động như các thư mục và cửa sổ Workspace KHÔNG hoạt động như Windows Explorer.

Sự khác biệt là cơ bản. Trong Windows Explorer, tệp thường chỉ xuất hiện một lần trong cành cây đã cho. Trong cây biểu tượng, biểu tượng đã cho xuất hiện nhiều lần vì nó xuất hiện trong mỗi cành danh mục mà nó thuộc về, ngay cả khi đó là cùng một biểu tượng và chỉ tồn tại duy nhất.

Cửa sổ Symbols được chia thành ba phần:

a) hộp tìm kiếm

b) cây danh mục

c) danh sách biểu tượng

Hộp tìm kiếm cho phép thực hiện tìm kiếm toàn văn (bao gồm cả phù hợp với ký tự đại diện) đối với biểu tượng và tên đầy đủ trong danh mục đã chọn. Ví dụ, nếu bạn chọn ngành "Công nghệ" và gõ A (chữ 'A' và ký tự đại diện), danh sách biểu tượng sẽ hiển thị tất cả các biểu tượng thuộc ngành Công nghệ có biểu tượng hoặc tên đầy đủ bắt đầu bằng chữ 'A'. Ví dụ khác có thể là tping -A0-FX - điều này sẽ trả về tất cả các biểu tượng forex trên cơ sở dữ liệu eSignal (những biểu tượng kết thúc bằng chuỗi -A0-FX).

Cây danh mục (xem hình) hiển thị các loại danh mục khác nhau.

Danh sách biểu tượng (phần dưới) hiển thị danh sách các biểu tượng thuộc danh mục đã chọn. Danh sách biểu tượng có thể được sắp xếp theo biểu tượng hoặc theo tên đầy đủ. Để sắp xếp, chỉ cần nhấp vào hàng đầu của danh sách. Một khi bạn chọn thứ tự sắp xếp mong muốn, nó sẽ được giữ cho tất cả các lựa chọn danh mục và tìm kiếm tiếp theo. Cũng order của các cột có thể được thay đổi để cột Tên đầy đủ xuất hiện đầu tiên. Để sắp xếp lại cột, nhấp vào tiêu đề cột, giữ nút chuột và kéo cột đến vị trí mong muốn. Sau đó, thả nút chuột.

Một ký hiệu đơn có thể thuộc NHIỀU danh mục cùng một lúc. Ví dụ, AAPL (Apple Inc.) sẽ thuộc về:

- Danh mục nhóm Cổ phiếu

- Danh mục thị trường Nasdaq

- Danh mục ngành Thông tin

- Danh mục ngành Sản xuất Máy tính (Comp-Computer Mfg)

Và cũng có thể thuộc về nhiều danh sách theo dõi và danh mục yêu thích cùng một lúc. Đó là lý do tại sao một biểu tượng sẽ xuất hiện trong nhiều lá của cây biểu tượng workspace. Bây giờ nếu bạn xóa BIỂU TƯỢNG, nó sẽ chắc chắn biến mất khỏi TẤT CẢ các danh mục vì bạn đã xóa chính biểu tượng, chứ không phải việc gán biểu tượng cho danh mục.

Có hai loại danh mục:

1. với thành viên độc lập lẫn nhau: nhóm, thị trường, ngành/nghiệp vụ, GICS - nghĩa là biểu tượng phải thuộc về một nhóm duy nhất, một thị trường duy nhất và một ngành/nghiệp vụ duy nhất trong một thời điểm. Bạn có thể di chuyển biểu tượng từ một nhóm/thị trường/ngành/nghiệp vụ này sang nhóm/thị trường/ngành/nghiệp vụ khác nhưng bạn không thể xóa việc gán này - bạn nên tạo nhóm/thị trường/ngành/nghiệp vụ "Chưa phân loại" và di chuyển các biểu tượng 'chưa được gán' vào đó.
2. với thành viên tự do tham gia: danh sách theo dõi/yêu thích/chỉ số- nghĩa là một biểu tượng có thể thuộc về BẤT KỲ số lượng (bao gồm cả không) của danh sách theo dõi (và danh mục yêu thích/chỉ số). Trong trường hợp này, bạn có thể xóa việc gán này bằng Danh sách theo dõi->Xóa
   Danh sách theo dõi được mô tả chi tiết trong Hướng dẫn sử dụng: Hướng dẫn: Làm việc với Danh sách theo dõi.

Cũng có một danh mục đặc biệt gọi là "ALL" xuất hiện trong cây biểu tượng workspace. Nó đơn giản là liệt kê TẤT CẢ các biểu tượng hiện có trong cơ sở dữ liệu.

Làm việc với Cơ bản về Ngành và Nghiệp vụ

Cơ bản - các ngành và lĩnh vực được định nghĩa sẵn

Bây giờ chúng ta sẽ tập trung vào việc thiết lập ngành và nghiệp vụ và gán các biểu tượng cho chúng. Đầu tiên, hãy để tôi thảo luận về một số ý tưởng cơ bản.

AmiBroker đi kèm với ví dụ cơ sở dữ liệu Dow Jones Industrials chứa tất cả 30 thành phần của trung bình thị trường nổi tiếng nhất thế giới. Chúng được gán cho các ngành và nghiệp vụ đã định trước. Những ngành và nghiệp vụ này hoàn toàn giống như được sử dụng trên trang web tài chính Yahoo và dưới đây là bảng liệt kê tất cả chúng:

Lĩnh vực	Lĩnh vực
Vật liệu cơ bản (0)	Sản xuất Hóa chất
	Hóa chất - Nhựa & Cao su
	Đồ chứa và bao bì
	Nhựa và cao su chế tạo

	Lâm nghiệp và sản phẩm gỗ
	Vàng và Bạc
	Sắt và Thép
	khai thác kim loại
	Sản phẩm chế tạo khác
	Khai thác Không kim loại
	Giấy và sản phẩm giấy
Hàng hóa vốn (1)	Hàng không & Quốc phòng
	Cung cấp & Thiết bị xây dựng
	Máy móc xây dựng & Nông nghiệp
	Nguyên liệu xây dựng
	Dịch vụ xây dựng
	Hàng hóa vốn khác
	Nhà di động & RVs
Tập đoàn (2)	Tập đoàn
Ngành tiêu dùng chu kỳ (3)	Thời trang/Phụ kiện
	Thiết bị & Dụng cụ
	Thiết bị Âm thanh & Video
	Nhà sản xuất Ô tô và xe tải
	Phụ tùng Ô tô & Xe tải
	Giày dép

	Đồ nội thất & Thiết bị
	Trang sức & Đồ bạc
	Nhiếp ảnh
	Sản phẩm Giải trí
	Vải - không phải trang phục
	Lốp
Ngành tiêu dùng không chu kỳ (4)	Đồ uống (có cồn)
	Đồ uống (Không có cồn)
	Nông sản
	Thủy sản/Động vật
	Chế biến thực phẩm
	Đồ dùng văn phòng
	Sản phẩm cá nhân & Gia đình
	Thuốc lá
Năng lượng (5)	Than
	Dầu & Khí - Tích hợp
	Hoạt động Dầu & Khí
	Dịch vụ & Thiết bị Giếng dầu
Tài chính (6)	Dịch vụ Tài chính tiêu dùng
	Bảo hiểm (Tai nạn & Sức khỏe)
	Bảo hiểm (Cuộc sống)

	Bảo hiểm (Khác)
	Bảo hiểm (Tài sản & Trách nhiệm)
	Dịch vụ Đầu tư
	Dịch vụ Tài chính khác
	Ngân hàng Trung tâm
	Ngân hàng Khu vực
	Ngân hàng Tiết kiệm/S&Ls
Chăm sóc sức khỏe (7)	Công nghệ sinh học & Thuốc
	Cơ sở Chăm sóc sức khỏe
	Thuốc chính
	Thiết bị và vật tư y tế
Dịch vụ (8)	Quảng cáo
	Truyền hình & Cáp TV
	Dịch vụ Kinh doanh
	Casino và trò chơi
	Dịch vụ Truyền thông
	Khách sạn & Nhà nghỉ
	Phim ảnh
	Dịch vụ Cá nhân
	In ấn & Xuất bản

	Dịch vụ In ấn
	Hoạt động Bất động sản
	Hoạt động Giải trí
	Cho thuê và thuê
	Nhà hàng
	Bán lẻ (Thời trang)
	Bán lẻ (Đặt hàng qua Catalog & Gửi)
	Bán lẻ (Cửa hàng & Giảm giá)
	Bán lẻ (Thuốc)
	Bán lẻ (Thực phẩm)
	Bán lẻ (Cải thiện nhà ở)
	Bán lẻ (Đặc sản)
	Bán lẻ (Công nghệ)
	Trường học
	Hệ thống & Dịch vụ An ninh
	Dịch vụ Quản lý Chất thải
Công nghệ (9)	Thiết bị Truyền thông
	Phần cứng máy tính
	Mạng Máy tính

Quan trọng là phải hiểu sự khác biệt giữa một lĩnh vực và một ngành: các ngành "thuộc" lĩnh vực. Ví dụ: "Air Courier", "Airline", "Railroads", "Trucking" là các ngành thuộc lĩnh vực "Transportation". Vì vậy, nếu một biểu tượng được gán cho một ngành cụ thể, nó cũng "tự động" được gán cho lĩnh vực tương ứng.

Trong cơ sở dữ liệu DJIA, mỗi cổ phiếu đều được gán cho một ngành cụ thể, ví dụ GM (General Motors) được gán cho ngành "Auto & Truck Manufacturers", và điều này ngụ ý rằng GM thuộc về lĩnh vực "Consumer/Cyclical".

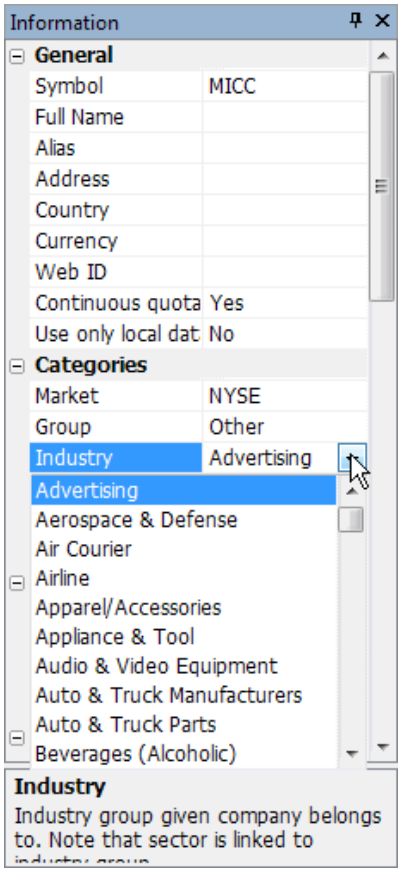
AmiBroker có thể xử lý tối đa 32 lĩnh vực và tối đa 256 ngành.

Làm thế nào để gán biểu tượng cho ngành?

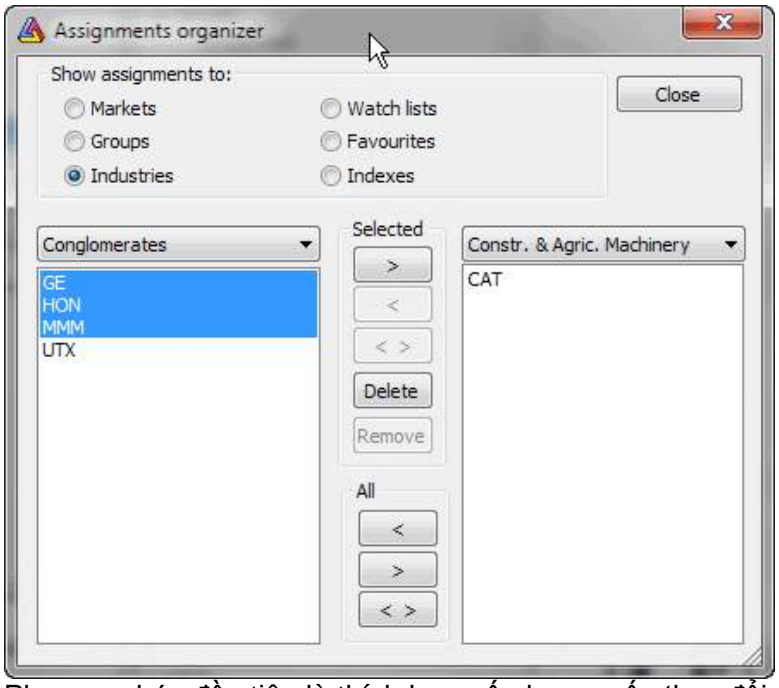
Bạn có thể thay đổi ngành mà biểu tượng cho trước đang gán bằng cách sử dụng Window->Information dialog (Hộp combo ngành)

Cách gán biểu tượng cho ngành?

Bạn có thể thay đổi ngành mà biểu tượng đã cho được gán bằng cách sử dụng hộp thoại Cửa sổ -> Thông tin (combo box Ngành).



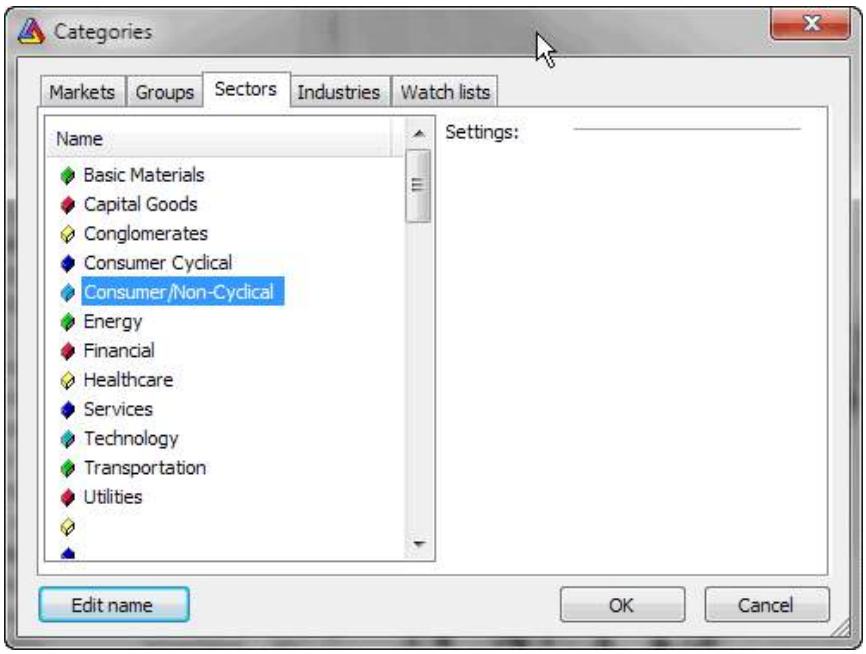
hoặc sử dụng Biểu tượng -> Sắp xếp Gán



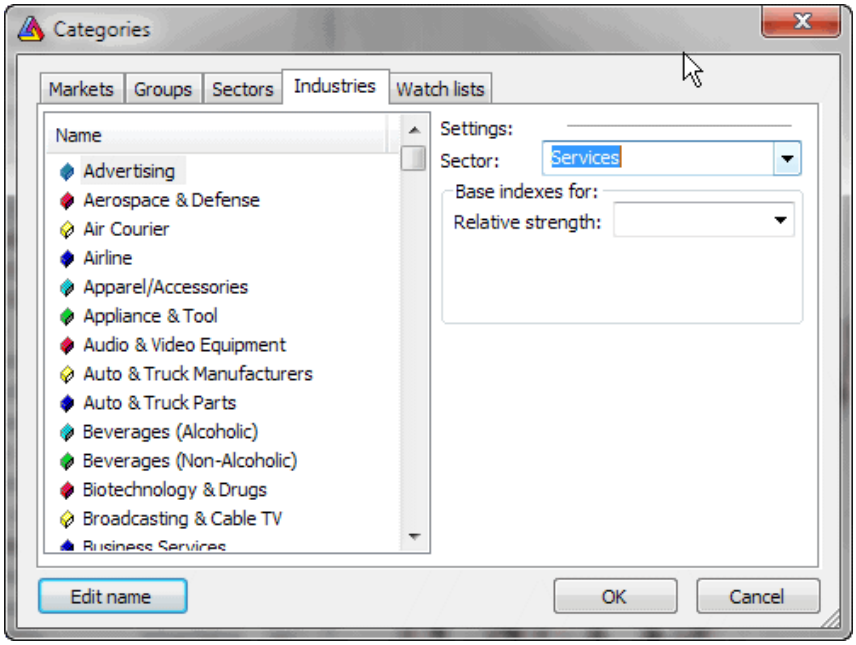
Phương pháp đầu tiên là thích hợp nếu bạn muốn thay đổi cài đặt của một biểu tượng đơn lẻ. Phương pháp sau thì tốt hơn nếu bạn muốn chuyển nhiều biểu tượng từ một danh mục này sang danh mục khác.

Cách định nghĩa các ngành và lĩnh vực của riêng bạn.

Vui lòng chuyển đến hộp thoại Symbol->Categories, hai tab cuối cùng là "Sectors" và "Industries". Đầu tiên, chuyển sang tab "Sectors" và bạn sẽ thấy danh sách 32 tên ngành. Bây giờ bạn có thể chọn ngành bằng cách nhấp một lần vào tên của nó và chỉnh sửa tên ngành bằng cách nhấn ENTER hoặc nhấp vào nút "Edit name". Nhấn ENTER lần nữa để chấp nhận thay đổi tên.



Sau khi bạn đã đổi tên các ngành, bạn có thể chuyển sang tab "Industries". Tương tự như tab trước, bạn có thể chọn công nghiệp trong danh sách và chỉnh sửa tên của nó theo cùng một cách. Ở đây bạn cũng có thể gán công nghiệp cho ngành bằng cách sử dụng combo "Sector". Chỉ cần chọn ngành mà bạn muốn gán cho công nghiệp hiện được chọn.



Nơi lưu trữ thông tin về Ngành và Lĩnh vực?

Thông tin về ngành và lĩnh vực thường được lưu trong cơ sở dữ liệu AmiBroker. Các tên ngành và lĩnh vực cùng với các cài đặt của chúng được lưu trong tệp broker.workspace (trong thư mục workspace), các tệp dữ liệu ký hiệu chỉ chứa thông tin về việc gán ký hiệu cho ngành đã cho (IndustryID).

Khi bạn tạo ra một workspace mới (một cơ sở dữ liệu), AmiBroker thiết lập các ngành và lĩnh vực của bạn theo các mẫu được lưu trong các tệp "broker.sectors" và "broker.industries". Đây là các tệp văn bản đơn giản có thể được chỉnh sửa bằng trình chỉnh sửa văn bản đơn giản (như Notepad). Các tệp này cũng có thể được sử dụng để thiết lập nhanh, tự động các lĩnh vực và ngành. AmiBroker đi kèm với broker.sectors và broker.industries được định nghĩa sẵn theo quy ước được mô tả ở trên (xem bảng). Bạn có thể ghi đè các tệp broker.sectors và broker.industries để xác định lược đồ mặc định của riêng bạn. Vì vậy, các tệp "broker.sectors" và "broker.industries" được sử dụng như một mẫu khi tạo workspace mới. Khi workspace được tạo, các tệp này không được xem xét. Bằng cách này, bạn có thể có các danh mục khác nhau trong mỗi workspace. Nếu bạn muốn AmiBroker tải chúng vào workspace đã tồn tại, hãy xóa tệp broker.workspace trước khi mở workspace. Nếu bạn mở workspace sau đó, AmiBroker sẽ đọc broker.sectors và broker.industries.

Cấu trúc của tệp broker.sectors rất đơn giản: đó là tệp văn bản đơn giản chứa các tên lĩnh vực được viết từng dòng như sau:

Basic Materials

Capital Goods

Conglomerates

Consumer Cyclical

Consumer/Non-Cyclical

Energy

Financial

Healthcare

Services

Technology

Transportation

Utilities

Cấu trúc của broker.industries tương tự, nhưng cũng có một số ở đầu mỗi dòng:

8 Advertising

1 Aerospace & Defense

10 Air Courier

10 Airline

3 Apparel/Accessories

3 Appliance & Tool

3 Audio & Video Equipment

3 Auto & Truck Manufacturers

3 Auto & Truck Parts

4 Beverages (Alcoholic)

4 Beverages (Non-Alcoholic)

7 Biotechnology & Drugs

8 Broadcasting & Cable TV

8 Business Services

8 Casinos & Gaming

0 Chemical Manufacturing

0 Chemicals - Plastics & Rubber

5 Coal

9 Communications Equipment

Các số phía trước tên ngành là "Sector IDs". Các số này quyết định ngành công nghiệp thuộc về nhóm nào. Vì một số ngành có thể thuộc về một nhóm - bạn có thể cần điền cùng một số cho sector Id. Sector IDs được đếm từ số 0, tức là số 0 đề cập đến dòng đầu tiên

(tên nhóm) của tệp "broker.sectors", trong khi 7 đề cập đến dòng thứ tám của tệp này. Trong ví dụ trên: Ngành "Advertising" thuộc về nhóm "Services", trong khi ngành "Aerospace & Defence" thuộc về nhóm "Capital Goods".

Nếu bạn không muốn cài đặt thông tin chi tiết về ngành công nghiệp và chỉ muốn gán biểu tượng cho các nhóm, bạn có thể định rõ mối quan hệ một-một giữa 32 ngành đầu tiên để chúng tương đương với nhóm. Sử dụng tệp broker.sectors như đã trình bày trước đó trong bài viết này, tệp broker.industries 1-1 sẽ trông như sau:

0 Basic Materials

1 Capital Goods

2 Conglomerates

3 Consumer Cyclical

4 Consumer/Non-Cyclical

5 Energy

6 Financial

7 Healthcare

8 Services

9 Technology

10 Transportation

11 Utilities

Hãy chú ý rằng tệp này về cơ bản giống như tệp broker.sectors với sự khác biệt duy nhất là chúng tôi đã thêm các số liên tiếp vào mỗi dòng. Sử dụng cấu hình này, việc thiết lập ngành công nghiệp sẽ tương đương với việc thiết lập nhóm.

Tự động hóa việc này

Như đã mô tả ở trên, tên biểu tượng và tên ngành công nghiệp cũng như mối quan hệ của chúng có thể được thiết lập nhanh chóng bằng cách sử dụng các tệp "broker.sectors" và "broker.industries". Việc này sẽ tiết kiệm một ít công việc cần thiết khác để nhập thông tin này vào cửa sổ Symbol->Categories.

Thật không may, còn rất nhiều công việc cần thiết để gán tất cả các biểu tượng cho ngành công nghiệp ngay cả khi sử dụng hộp thoại Symbol->Organize Assignments.

May mắn thay, có một cách để thiết lập và cập nhật cơ sở dữ liệu một cách tự động.

Trong phiên bản trước 5.60, nó vẫn yêu cầu viết kịch bản và rất nhiều công việc (xem số 4 của AmiBroker Tips newsletter) nhưng phiên bản 5.60 mang đến những cách hoàn toàn mới để thiết lập cơ sở dữ liệu một cách tự động.

Nhập khẩu ASCII cải tiến trong v5.60 cho phép nhập biểu tượng, ngành công nghiệp và các tên nhóm và xây dựng cơ sở dữ liệu hoàn chỉnh chỉ trong một bước

Giả sử rằng chúng ta có một file CSV trông như sau:

"DDD","3D Systems Corporation","Công nghệ","Phần mềm máy tính: Phần mềm đóng gói"

"MMM","3M Company","Chăm sóc sức khỏe","Dụng cụ y tế/Nha khoa"

"SVN","7 Days Group Holdings Limited","Dịch vụ tiêu dùng","Khách sạn/Khu nghỉ dưỡng"

"AHC","A.H. Belo Corporation","Dịch vụ tiêu dùng","Báo/ Tạp chí"
"AIR","AAR Corp.","Hàng tiêu dùng","Hàng không" "AAN","Aaron's, Inc.","Công nghệ","Dịch vụ thương mại đa dạng"

Giờ chúng ta có thể nhập nó vào AmiBroker và tự động thiết lập tất cả các ngành và công nghiệp sử dụng định dạng này

```
$FORMAT Ticker, FullName,SectorName,IndustryName
 $SEPARATOR ,
 $AUTOADD 1
 $NOQUOTES 1
 $OVERWRITE 1
 $CLEANSECTORS 1
 $SORTSECTORS 1
```

Hai lệnh cuối cùng ($CLEANSECTORS và $SORTSECTORS) hướng dẫn AmiBroker dọn dẹp (xóa) tên ngành/tên công nghiệp hiện có trước khi nhập và sắp xếp các ngành được nhập mới sau khi nhập để chúng xuất hiện theo thứ tự chữ cái.

AmiBroker sẽ đọc từng tệp ASCII một, sau đó nó sẽ kiểm tra bất cứ khi nào tên ngành/tên công nghiệp đã được đưa ra, nếu không - nó sẽ tạo ngành mới/công nghiệp mới. Sau đó, nó sẽ gán biểu tượng đã cho cho ngành/công nghiệp chỉ định.

Kết quả sẽ là một cơ sở dữ liệu với cấu trúc ngành mới/công nghiệp đang được thiết lập và biểu tượng được gán cho các ngành và công nghiệp thích hợp.

Chức năng được mô tả được sử dụng để triển khai công cụ "Cập nhật danh sách ký hiệu Hoa Kỳ và các loại hình" trong menu "Công cụ".

Chức năng "Cập nhật danh sách ký hiệu và danh mục Hoa Kỳ chỉ với một cú nhấp chuột"

Chức năng tự động thiết lập và cập nhật cơ sở dữ liệu chứng khoán Hoa Kỳ có sẵn từ menu Công cụ -> Cập nhật danh sách ký hiệu và danh mục Hoa Kỳ. Chức năng này được triển khai bằng cách sử dụng lệnh mới #import và lệnh nhập ASCII mới được mô tả ở trên.

Lệnh này sẽ tải danh sách ký hiệu, lĩnh vực và ngành từ trang web amibroker.com và tạo hoặc cập nhật cơ sở dữ liệu hiện tại với các cổ phiếu được niêm yết trên NYSE, Nasdaq và AMEX. Nó cũng sẽ tạo cấu trúc lĩnh vực và ngành, và gán các cổ phiếu vào các ngành phù hợp.

LƯU Ý: Hãy cẩn thận vì việc sử dụng công cụ này sẽ XÓA (xóa) bất kỳ lĩnh vực/ngành nào hiện có và thay thế chúng bằng các lĩnh vực/ngành được nhập tự động.

Ghi chú về Gics

GICS là viết tắt của Global Industry Classification Standard (http://en.wikipedia.org/wiki/Global_Industry_Classification_Standard).

AmiBroker cũng cho phép hệ thống phân loại GICS 4 cấp, nhưng cơ sở dữ liệu demo không có biểu tượng được phân loại theo tiêu chuẩn đó. Bạn có thể tìm thấy các mã phân loại GICS trong tệp GICS.txt bên trong thư mục AmiBroker.

Ghi chú về ICB

ICB nghĩa là Industry Classification Benchmark (**http://en.wikipedia.org/wiki/Industry_Classification_Benchmark**).

AmiBroker cũng hỗ trợ hệ thống phân loại 4 cấp của ICB, nhưng cơ sở dữ liệu demo không có các biểu tượng được phân loại theo tiêu chuẩn này. Bạn có thể tìm mã phân loại ICB trong tệp ICB.txt nằm trong thư mục AmiBroker

Làm việc với danh sách theo dõi

AmiBroker 5.00 giờ đây sử dụng hệ thống danh sách theo dõi mới. Danh sách theo dõi khác với các loại danh mục khác (như nhóm, thị trường, ngành, lĩnh vực) ở chỗ bạn có thể gán một biểu tượng đơn lẻ cho nhiều danh sách theo dõi hơn.

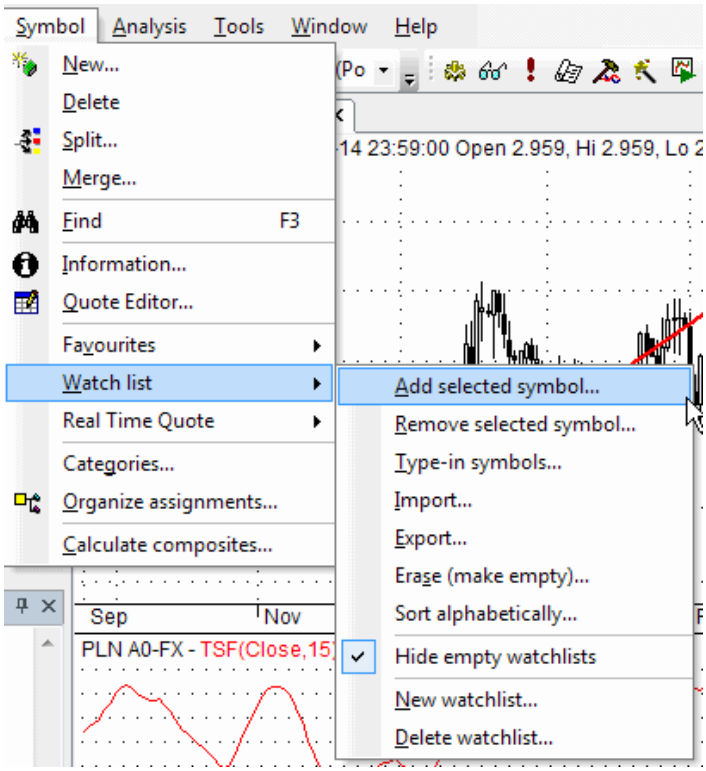
Bạn có thể sử dụng SỐ danh sách theo dõi KHÔNG GIỚI HẠN với tên của họ có thể định rõ trong cửa sổ Symbol->Categories. Các thành viên của mỗi danh sách theo dõi được hiển thị trong cây biểu tượng phía dưới "Danh sách theo dõi".

Danh sách theo dõi giờ đây được lưu trữ dưới dạng các tệp văn bản bên trong thư mục "Danh sách theo dõi" trong cơ sở dữ liệu. Thư mục chứa bất kỳ số lượng .TLS nào với các danh sách theo dõi chính và index.txt xác định thứ tự của các danh sách theo dõi. Bạn có thể thêm tệp .tls của riêng mình (một biểu tượng trên mỗi dòng) và AmiBroker sẽ cập nhật tự động index.txt (thêm bất kỳ danh sách theo dõi mới nào ở cuối)Các tệp .TLS cũng có thể được mở trong AmiQuote.

Danh sách theo dõi nhớ thứ tự mà các biểu tượng được thêm vào, vì vậy ví dụ nếu bạn sắp xếp danh sách kết quả AA theo một thứ tự nào đó và sau đó bạn "thêm biểu tượng vào danh sách theo dõi" thứ tự sẽ được giữ trong danh sách theo dõi.

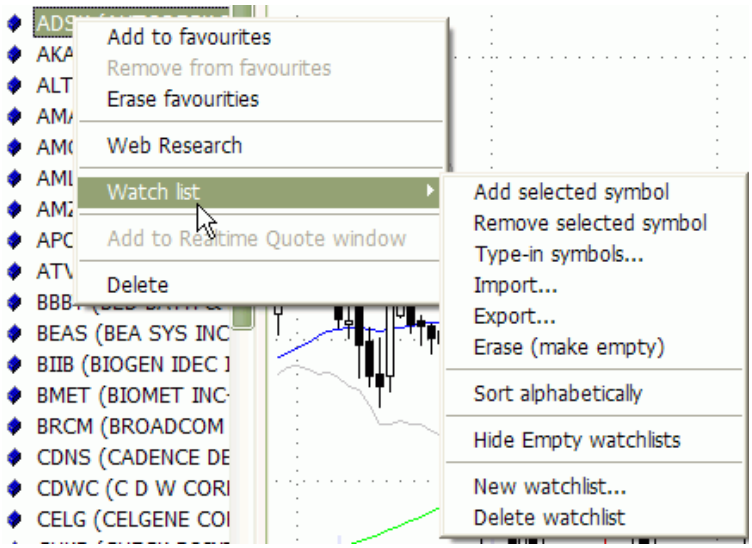
Thêm / xóa danh sách theo dõi

Bạn có thể thêm/xóa danh sách theo dõi bằng cách sử dụng Symbol->Watch List->New Watchlist và Symbol->Watch List->Delete Watchlist từ menu danh sách theo dõi hoặc từ menu ngữ cảnh của danh sách theo dõi. Lưu ý rằng nếu bạn đã tùy chỉnh bất kỳ mục nào trong menu, bạn có thể cần phải di chuyển đến Tools->Customize, chọn "Menu Bar" và nhấn nút "Reset" đối với mục menu mới này để xuất hiện.

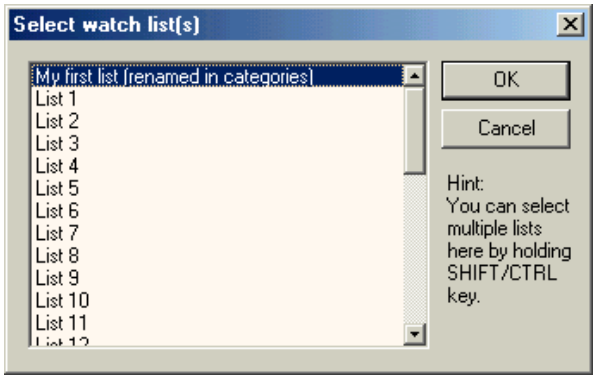


Thêm ticker vào danh sách theo dõi

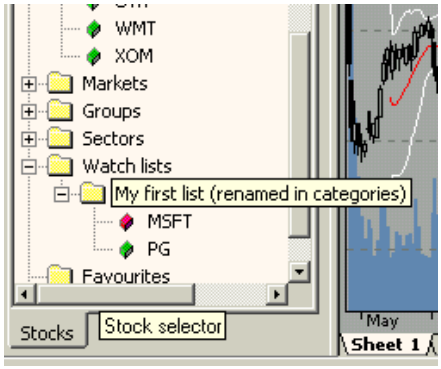
Bạn có thể dễ dàng thêm một ticker vào danh sách theo dõi chỉ bằng cách nhấn chuột phải lên mục trong cây biểu tượng và chọn tùy chọn:



Sau khi chọn tùy chọn này, một cửa sổ chọn danh sách theo dõi sẽ xuất hiện:



Tại đây, bạn nên chọn danh sách mà bạn muốn thêm ký hiệu vào. Lưu ý rằng bạn có thể thêm một ký hiệu vào nhiều danh sách cùng một lúc bằng cách giữ phím CTRL trong khi nhấp vào các mục trong danh sách. Sau khi nhấp vào OK, ký hiệu đã chọn (ví dụ: MSFT) sẽ xuất hiện trong danh sách theo dõi mà bạn đã chọn.



Bạn cũng có thể nhập trực tiếp các ký hiệu vào danh sách theo dõi bằng cách sử dụng tùy chọn **Ký hiệu -> Danh sách theo dõi -> Nhập vào**. Các ký hiệu nên được phân tách bằng

dấu phẩy. Ngoài ra, bạn có thể nhấp chuột phải vào tên danh sách theo dõi trong cây không gian làm việc để nhập trực tiếp các ký hiệu vào danh sách theo dõi.

Sắp xếp các ký hiệu trong danh sách theo dõi:

Bây giờ bạn có thể sắp xếp theo thứ tự bảng chữ cái các ký hiệu trong danh sách theo dõi - nhấp vào danh sách theo dõi bằng nút chuột PHẢI và chọn "Sắp xếp theo thứ tự bảng chữ cái"

Xóa mã chứng khoán khỏi danh sách theo dõi

Xóa ký hiệu khỏi danh sách theo dõi cũng dễ như thêm chúng. Chỉ cần nhấp vào thành viên danh sách bằng nút chuột phải và chọn Xóa khỏi danh sách theo dõi. Sau đó, cửa sổ chọn danh sách tương tự sẽ xuất hiện, chỉ hiển thị những danh sách mà ký hiệu hiện đang được chọn thuộc về. Bây giờ bạn có thể chọn một hoặc nhiều danh sách và ký hiệu sẽ bị xóa khỏi danh sách.

Xóa danh sách theo dõi

Đôi khi bạn có thể muốn xóa (hoặc xóa) toàn bộ danh sách theo dõi. Sau đó, chỉ cần chọn tùy chọn Biểu tượng->Theo dõi

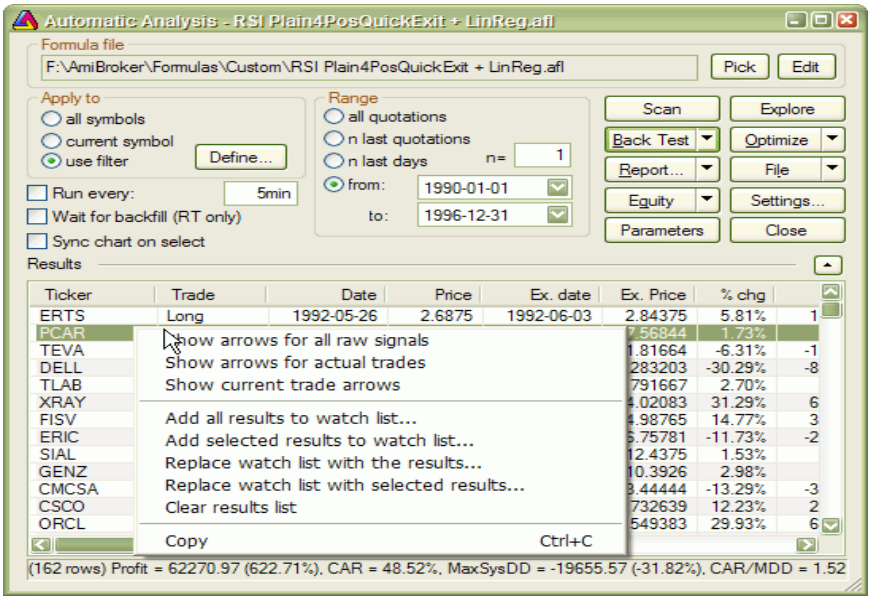
danh sách->Xóa (trống). Trong cửa sổ chọn danh sách theo dõi, hãy đánh dấu danh sách(các) bạn muốn xóa và nhấp vào OK. Theo cách này, danh sách theo dõi đã chọn sẽ trở nên trống rỗng.

Ẩn/Hiển thị danh sách theo dõi trống

Theo mặc định, danh sách theo dõi trống được hiển thị trong cây ký hiệu nhưng bạn có thể ẩn chúng bằng cách nhấp chuột phải vào danh sách theo dõi trong cây ký hiệu và chọn menu "Ẩn danh sách theo dõi trống". Để hiển thị, hãy chọn lại tùy chọn này.

Sử dụng danh sách theo dõi trong cửa sổ phân tích tự động

AmiBroker cung cấp cho bạn một cách rất dễ dàng để lưu trữ kết quả quét, kiểm tra ngược và khám phá vào danh sách theo dõi chỉ bằng một cú nhấp chuột - chỉ cần chạy công thức AFL yêu thích của bạn trên toàn bộ cơ sở dữ liệu và nhấp vào danh sách kết quả bằng nút chuột phải để xem menu sau:



Khi bạn chọn Thêm tất cả/kết quả đã chọn vào danh sách theo dõi, một bộ chọn danh sách theo dõi sẽ xuất hiện, tại đó bạn chọn ký hiệu danh sách nào sẽ được thêm vào, sau đó sau khi nhấp vào OK, tất cả các ký hiệu được lọc theo quy tắc giao dịch của bạn sẽ tự động xuất hiện trong danh sách theo dõi mà bạn chọn.

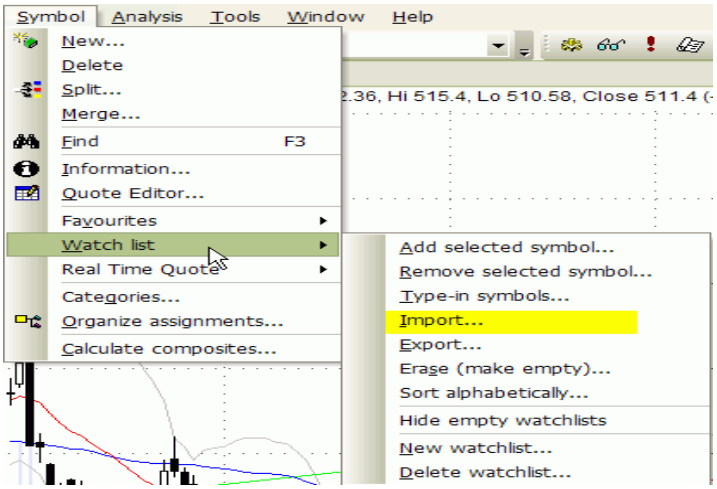
Bạn cũng có thể sử dụng tùy chọn Thay thế danh sách theo dõi bằng kết quả/kết quả đã chọn

Tùy chọn mới này sẽ làm trống danh sách theo dõi trước khi thêm kết quả. Thứ tự của các ký hiệu trong danh sách kết quả được bảo toàn trong danh sách theo dõi.

Cách nhập/xuất danh sách theo dõi từ/vào tệp

NHẬP DANH SÁCH THEO DÕI TỪ TỆP

1. Chọn Symbol->Watch List->Import menu hoặc nhấp chuột phải vào danh sách theo dõi trong cây và chọn Import.



2. Chọn danh sách theo dõi đích

3. Trong hộp thoại tệp sẽ xuất hiện, hãy chọn tệp .TLS, .LST, .TXT hoặc .CSV

.TLS, .CSV, .TXT phải có một ký hiệu mã cổ phiếu trên mỗi dòng và không có trường nào khác.

.LST là tệp danh sách chuẩn của Quotes-Plus, được phân tách bằng dấu phẩy có ký hiệu mã cổ phiếu ở cột đầu tiên và một số dữ liệu bổ sung ở các cột còn lại. AmiBroker chỉ đọc cột đầu tiên và bỏ qua phần còn lại.

4. Nhấp vào OK.

XUẤT DANH SÁCH THEO DÕI SANG TỆP

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

hoặc nhấp chuột phải vào danh sách theo dõi trong cây và chọn Export.

2. Chọn danh sách theo dõi nguồn và chuyển sang "Nguồn dữ liệu ngoài"

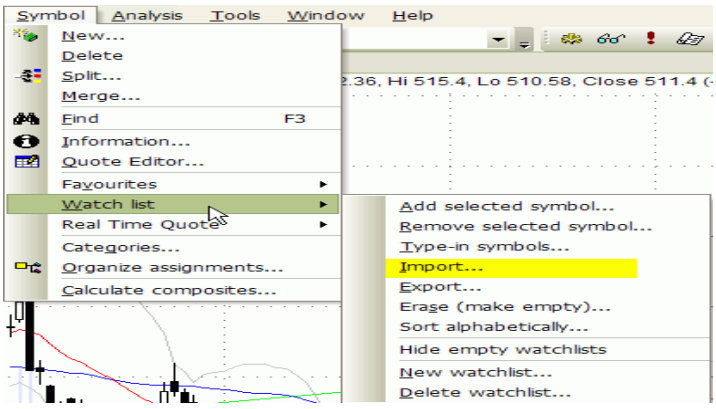
3. Trong hộp thoại tệp, hãy chọn tệp để xuất sang. Tệp được tạo sẽ là tệp ASCII đơn giản, mỗi dòng có một ký hiệu ticker.

Cách nhập/xuất danh sách theo dõi từ/sang cơ sở dữ liệu ngoài

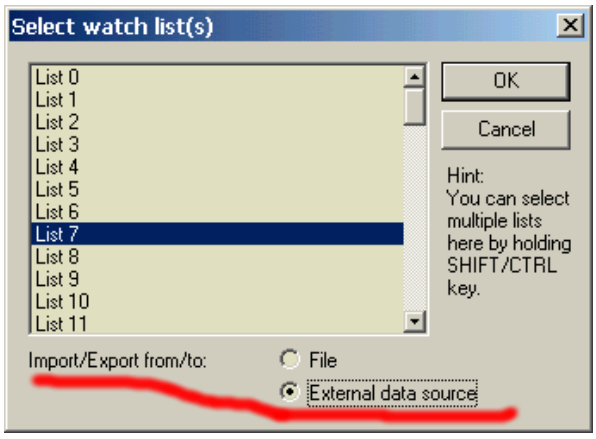
LƯU Ý: Chỉ hoạt động nếu bạn đặt "Nguồn dữ liệu" thành plugin "Fast Track" trong Tệp->Cài đặt cơ sở dữ liệu (và tất nhiên là nếu bạn đã cài đặt cơ sở dữ liệu FastTrack).

NHẬP GIA ĐÌNH TỪ FASTTRACK

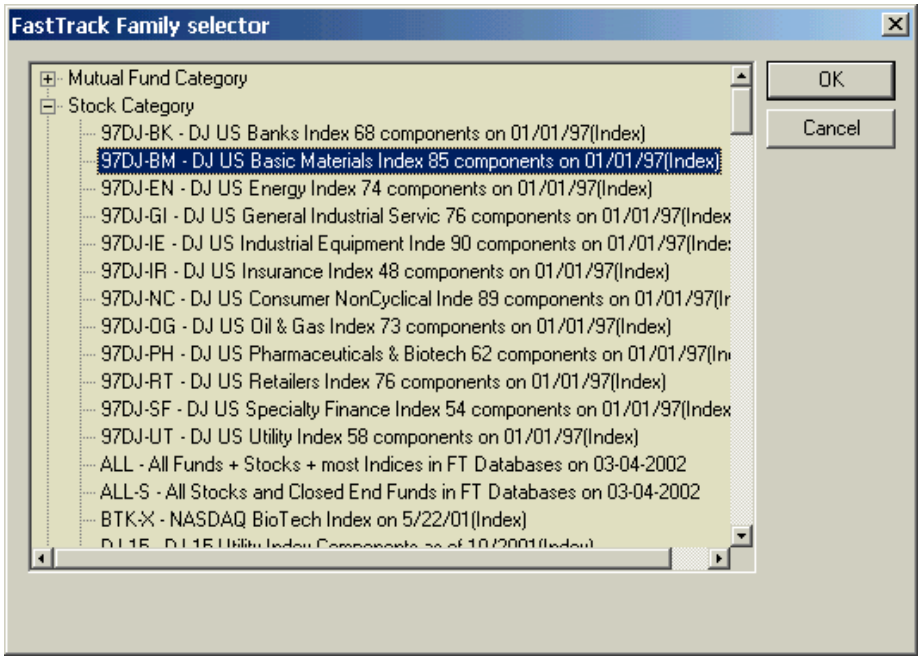
1. Chọn Biểu tượng->Danh sách theo dõi->menu Nhập hoặc nhấp chuột phải vào danh sách theo dõi trong cây và chọn Nhập.



2. Chọn danh sách theo dõi đích và chuyển sang "Nguồn dữ liệu ngoài"



3. Trong hộp thoại sẽ xuất hiện, hãy mở rộng một danh mục và chọn gia đình mà bạn muốn nhập ký hiệu từ đó:



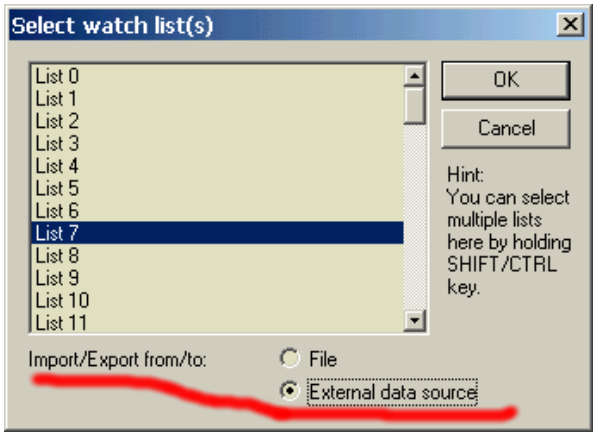
4. Nhấp vào OK.

XUẤT DANH SÁCH THEO DÕI SANG GIA ĐÌNH FASTTRACK

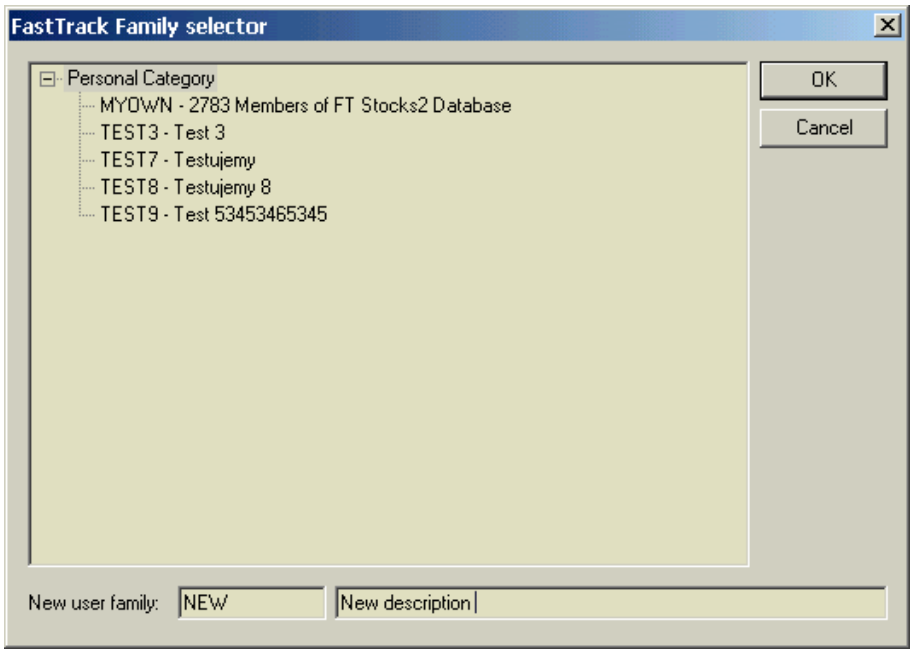
1. Chọn Biểu tượng->Danh sách theo dõi->menu Xuất.

hoặc nhấp chuột phải vào danh sách theo dõi trong cây và chọn Xuất.

2. Chọn danh sách theo dõi nguồn và chuyển sang "Nguồn dữ liệu ngoài"



3. Bây giờ, NHẬP họ cá nhân mới vào "Họ người dùng mới" (và mô tả trong tệp bên phải) HOẶC chọn họ cá nhân hiện có từ danh sách.



Hiểu cách thức hoạt động của AFL

Giới thiệu

Một trong những khía cạnh quan trọng nhất của AFL là nó là một ngôn ngữ xử lý mảng. Nó hoạt động trên các mảng (hoặc hàng/vectơ) dữ liệu. Cách hoạt động này khá giống với cách hoạt động của các bảng tính phổ biến (như Microsoft Excel). Bất kỳ ai quen thuộc với MS Excel đều có thể dễ dàng nắm bắt AFL. - Trên thực tế, tất cả các ví dụ trong bài viết này đều được tạo bằng MS Excel.

Mảng là gì?

Mảng chỉ đơn giản là một danh sách (hoặc hàng) giá trị. Trong một số cuốn sách, nó có thể được gọi là vectơ. Mỗi hàng giá trị được đánh số trong ví dụ biểu thị một mảng riêng lẻ. Amibroker đã lưu trữ trong cơ sở dữ liệu của mình 6 mảng cho mỗi ký hiệu. Một mảng cho giá mở cửa, một mảng cho giá thấp, một mảng cho giá cao, một mảng cho giá đóng cửa và một mảng cho khối lượng (xem các hàng được gắn nhãn 1-5 bên dưới) và một mảng cho lãi suất mở. Những mảng này có thể được tham chiếu trong AFL là mở, thấp, cao, đóng, khối lượng, openint hoặc o, l, h, c, v, oi.

Tất cả các chỉ số mảng trong AFL đều bắt đầu từ số 0, tức là đếm các thanh bắt đầu từ thanh 0 (cũ nhất). Thanh mới nhất (mới nhất) có chỉ số là (BarCount-1). Mảng 10 phần tử sẽ có BarCount = 10 và các chỉ số đi từ 0 đến 9 như minh họa bên dưới

		(cũ nhất)	---->dấu thời gian tăng dần---->								(mới nhất)
	Bar	0	1	2	3	4	5	6	7	8	9
1	Mở	1.23	1.24	1.21	1.26	1.24	1.29	1.33	1.32	1.35	1.37

Hình 1. Mở mảng giá

Bất kỳ mảng nào khác được tính toán từ 6 mảng này bằng các công thức được tích hợp sẵn trong AFL. Các mảng này không được lưu trữ trong cơ sở dữ liệu nhưng được tính toán khi cần thiết.

Mỗi giá trị riêng lẻ trong một mảng đều có ngày liên kết với nó. Nếu bạn bật tùy chọn chú giải công cụ (Tùy chọn -> Tab Khác -> Chú giải công cụ dữ liệu giá), khi bạn di chuyển con trỏ qua nến trên biểu đồ nến hàng ngày, một hình chữ nhật nhỏ màu vàng sẽ xuất hiện. Sau đó, AFL sẽ tra cứu các giá trị mở, thấp, cao, đóng, khối lượng trong mảng thích hợp và hiển thị chúng bên trong chú giải công cụ.

Xử lý mảng - tại sao AFL lại nhanh như vậy?

Hãy xem cách xử lý câu lệnh sau:

MyVariable = (High + Low)/2;

Khi AFL đang đánh giá câu lệnh như thế này (High + Low)/2, nó không cần phải diễn giải lại mã này cho từng thanh. Thay vào đó, nó lấy Mảng Cao và Mảng Thấp và thêm các phần tử mảng tương ứng trong một giai đoạn. Nói cách khác, toán tử + (và các toán tử khác nữa) hoạt động trên các mảng cùng một lúc và nó được thực thi ở tốc độ mã biên dịch đầy đủ, sau đó mảng kết quả (mỗi phần tử của nó) được chia cho 2 cũng trong một giai đoạn.

Hãy xem xét chi tiết - xem hình 2.. Khi công cụ AFL xem xét (High + Low)/2, trước tiên nó sẽ lấy các mảng Cao (1) và Thấp (2) và tạo ra (trong một bước biên dịch duy nhất) mảng tạm thời (3). Sau đó, nó tạo mảng cuối cùng (4) bằng cách chia mỗi phần tử của mảng tạm thời cho hai. Kết quả này được gán cho myVariable

1	High (mảng tích hợp)	1.24	1.27	1.25	1.29	1.25	1.29	1.35	1.35	1.37	1.29
2	Low (mảng tích hợp)	1.20	1.21	1.19	1.20	1.21	1.24	1.30	1.28	1.31	1.27
3	High+Low (mảng tạm thời được tạo trong quá trình đánh giá)	2.44	2.48	2.44	2.49	2.46	2.53	2.65	2.63	2.68	2.46
4	(High+Low) /2 (được gán cho MyVariable)	1.22	1.24	1.22	1.245	1.23	1.265	1.325	1.315	1.34	1.23

Hình 2. Các bước AFL khi xử lý (High + Low) /2

Trung bình động, câu lệnh điều kiện

Bây giờ chúng ta hãy xem xét đoạn mã sau:

Cond1 = Đóng > MA(Đóng, 3);

Cond2 = Khối lượng > Tham chiếu(Khối lượng, -1);

Mua = Cond1 VÀ Cond2;

Bán = Cao > 1,30;

Đoạn mã này tạo ra tín hiệu mua khi giá đóng cửa hôm nay cao hơn giá đóng cửa trung bình động 3 ngày VÀ khối lượng hôm nay cao hơn khối lượng hôm qua. Nó cũng tạo ra tín hiệu bán khi giá cao nhất hôm nay cao hơn 1,30.

Nếu trong mã AFL của bạn, bạn cần xem giá đóng cửa có lớn hơn giá trung bình động đơn giản 3 ngày hay không, AFL sẽ chạy qua mảng đóng cửa trước tiên, tạo một mảng mới có tên là MA(close,3) cho ký hiệu đang được phân tích. Mỗi ô trong mảng mới sau đó có thể được so sánh từng ô trong mảng đóng cửa. Trong ví dụ này, một mảng có tên là Cond1 được tạo theo cách này. Đối với mỗi ô có giá đóng cửa lớn hơn giá trị ô tương ứng trong MA(close,3), giá trị ô cho mảng mới 'Cond1' được đặt thành '1'. Nếu giá đóng cửa không lớn hơn giá tương ứng trong mảng đóng cửa, giá trị trong 'Cond1' được đặt thành '0'.

AFL cũng có thể xem trước hoặc sau một số ô trong một mảng bằng hàm Ref (xem hàng 6, nơi mảng tạm thời được tạo giữ khối lượng ngày trước đó)

Trong hàng 9, một mảng mới có tên là Cond2 đã được tạo bằng cách so sánh giá trị của từng ô trong mảng khối lượng với ô trước đó, đặt giá trị ô Cond2 thành '1' nếu đúng và '0' nếu sai.

Hàng 10 hiển thị một mảng có tên là 'Mua' được tạo bằng cách so sánh các giá trị ô trong Cond1 với các giá trị ô trong Cond2. Nếu ô trong Cond1 có '1' VÀ ô tương ứng trong Cond2 cũng vậy thì '1' được đặt trong ô mảng 'Mua'.

Hàng 11 hiển thị một mảng có tên là 'Bán' được tạo bất cứ khi nào giá trị ô trong mảng đóng lớn hơn $1,30.

	Bar	0	1	2	3	4	5	6	7	8	9
1	Mở	1.23	1.24	1.21	1.26	1.24	1.29	1.33	1.32	1.35	1.37
2	Cao	1.24	1.27	1.25	1.29	1.25	1.29	1.35	1.35	1.37	1.29
3	Thấp	1.20	1.21	1.19	1.20	1.21	1.24	1.30	1.28	1.31	1.27
4	Đóng	1.23	1.26	1.24	1.28	1.25	1.25	1.31	1.30	1.32	1.28
5	Mức độ	8310	3021	5325	2834	1432	5666	7847	555	6749	3456
6	Ref(Volume, -1) (mảng tạm thời được tạo trong quá trình eval)	Vô giá trị	8310	3021	5325	2834	1432	5666	7847	555	6749
7	MA(Close, 3) (mảng tạm thời được tạo trong quá trình đánh giá)	Vô giá trị	Vô giá trị	1.243	1.260	1.257	1.260	1.270	1.287	1.310	1.300
8	Cond1 = Đóng < MA(đóng,3) (cho 1 (hoặc đúng) nếu điều kiện được đáp ứng, nếu không thì cho 0)	Vô giá trị	Vô giá trị	1	0	1	1	0	0	0	1
9	Cond2 = Khối lượng > Tham chiếu(khối lượng,-1)	Vô giá trị	0	1	0	0	1	1	0	1	0
10	Mua = Cond1 VÀ Cond2	Vô giá trị	Vô giá trị	1	0	0	1	0	0	0	0
11	Bán = Cao > 1,30	0	0	0	0	0	0	1	1	1	0

Rõ ràng Buy và Sell là các mảng đặc biệt có thể hiển thị kết quả trong cửa sổ Analyser hoặc trên màn hình bằng giá trị màu đỏ hoặc xanh lá cây khi cần.

Phức tạp hơn một chút

Các ví dụ trên rất đơn giản. Bây giờ tôi sẽ chỉ giải thích 3 điều có vẻ gây nhầm lẫn cho người dùng:

• tham chiếu các giá trị đã chọn (SelectedValue, BeginValue, EndValue, LastValue)

• Hàm IIF

• Hàm AMA

Như đã viết trong Hướng dẫn: Hướng dẫn lập biểu đồ cơ bản, bạn có thể chọn bất kỳ báo giá nào từ biểu đồ và bạn có thể đánh dấu phạm vi From-To. Thanh được chọn theo đường thẳng đứng được gọi là thanh "selected" trong khi thanh bắt đầu và kết thúc của phạm vi được gọi là thanh "begin" và "end". AFL có các hàm đặc biệt cho phép tham chiếu giá trị của mảng tại thanh đã chọn, bắt đầu và kết thúc tương ứng. Các hàm này được gọi là SelectedValue, BeginValue và EndValue. Có một hàm nữa có tên là LastValue cho phép lấy giá trị của mảng tại thanh cuối cùng. Bốn hàm này lấy phần tử mảng tại thanh đã cho và trả về SỐ DUY NHẤT biểu thị giá trị của mảng tại điểm đã cho. Điều này cho phép tính toán một số thống kê liên quan đến các điểm đã chọn. Ví dụ:

EndValue(Close) - BeginValue(Close)

Sẽ cung cấp cho bạn sự thay đổi đô la giữa các mức giá đóng cửa trong phạm vi từ-đến đã chọn.

Khi số được lấy ra bởi bất kỳ hàm nào trong số các hàm này được so sánh với một mảng hoặc bất kỳ phép toán số học nào khác liên quan đến số và mảng được thực hiện, nó hoạt động giống như số được kéo dài tất cả các phần tử mảng. Điều này được minh họa trong bảng bên dưới (hàng 2, 6, 7). Màu xanh lá cây đánh dấu thanh "bắt đầu" và màu đỏ đánh dấu thanh "kết thúc".Thanh đã chọn được đánh dấu bằng màu xanh lam.

	Thanh	0	1	2	3	4	5	6	7	8	9
1	Mở	1.23	1.24	1.21	1.26	1.24	1.29	1.33	1.32	1.35	1.37
2	BeginValue(Mở)	1.24	1.24	1.24	1.24	1.24	1.24	1.24	1.24	1.24	1.24
3	EndValue(Mở)	1.32	1.32	1.32	1.32	1.32	1.32	1.32	1.32	1.32	1.32
4	SelectedValue(Mở)	1.21	1.21	1.21	1.21	1.21	1.21	1.21	1.21	1.21	1.21
5	LastValue(Mở)	1.37	1.37	1.37	1.37	1.37	1.37	1.37	1.37	1.37	1.37
6	Đóng	1.22	1.26	1.23	1.28	1.25	1.25	1.31	1.30	1.32	1.28
7	Đóng <= BeginValue(Mở)	1	0	1	0	0	0	0	0	0	0
8	kết quả = IIF(Đóng <= BeginValue(	1.22	1.24	1.23	1.26	1.24	1.29	1.33	1.32	1.35	1.37

	Mở), Đóng, Mở);										
9	Chu kỳ	2	3	4	2	3	5	2	3	4	2
10	Hệ số = 2/(Chu kỳ+1)	0.667	0.500	**0.400**	0.667	0.500	0.333	0.667	0.500	0.400	0.667
11	1 - Hệ số	0.333	0.500	0.600	0.333	0.500	0.667	0.333	0.500	0.600	0.333
12	AMA(Đóng, Hệ số)	0.8125	1.0363	**1.1138**	1.2234	1.2367	1.2399	1.2853	1.2927	1.3036	1.2866

Bây giờ là hàm IIF(condition, truepart, falsepart). Hàm này trả về giá trị của đối số thứ hai (truepart) hoặc thứ ba (falsepart) tùy thuộc vào điều kiện. Như bạn có thể thấy trong bảng trên ở hàng 8, các giá trị đến từ mảng Close (truepart) cho các thanh khi điều kiện là đúng (1) và đến từ mảng Open (falsepart) cho các thanh còn lại. Trong trường hợp đó, mảng được trả về bởi hàm IIF bao gồm một số giá trị từ Close và một số giá trị từ mảng Open. Lưu ý rằng cả truepart và falsepart đều là mảng và chúng được đánh giá bất kể điều kiện (do đó, đây không phải là câu lệnh IF-THEN-ELSE thông thường mà là hàm trả về mảng)

Hàm AMA(array, factor) dường như gây ra nhiều vấn đề nhất trong việc hiểu hàm này. Nhưng thực tế, hàm này rất đơn giản. Nó hoạt động theo cách đệ quy. Nghĩa là nó sử dụng giá trị trước đó để tính giá trị hiện tại. Nó xử lý từng thanh mảng, với mỗi bước, nó nhân ô cho trước của đối số đầu tiên (mảng) với ô cho trước của đối số thứ hai (hệ số) và thêm vào giá trị AMA trước đó nhân với (hệ số 1). Hãy xem xét thanh số 2 (được đánh dấu màu xanh lam). Giá trị AMA trên thanh đó được đưa ra bằng cách nhân giá đóng cửa từ thanh đó (1,23) với hệ số (0,4). Sau đó, chúng ta cộng giá trị AMA trước đó (1,0363) nhân với (hệ số 1 = 0,6). Kết quả (làm tròn đến 4 chữ số) là 1,23 * 0,4 + 1,0363 * 0,6 = 1,1138.

Nếu bạn nhìn vào các số liệu ở hàng 12, bạn có thể nhận thấy rằng các giá trị này trông giống như đường trung bình động của giá đóng cửa. Và điều đó là đúng. Trên thực tế, chúng tôi đã trình bày cách tính đường trung bình động hàm mũ chu kỳ biến đổi bằng hàm AMA.

Vòng lặp mới

Với phiên bản 4.40, AmiBroker mang đến khả năng lặp lại qua các dấu ngoặc kép bằng vòng lặp for và while và thêm câu lệnh điều khiển luồng if-else. Những cải tiến này giúp có thể hoạt động CẢ HAI cách: sử dụng xử lý MẢNG (được mô tả ở trên) để tăng tốc độ và đơn giản hoặc sử dụng VÒNG LẶP để thực hiện những việc phức tạp. Để biết ví dụ về cách triển khai trung bình mũ chu kỳ biến đổi (được mô tả ở trên) bằng vòng lặp, hãy xem đoạn mã sau:

```
Period = ... một số phép tính

vaexp[ 0 ] = Close[ 0 ]; // khởi tạo giá trị đầu tiên for( i = 1; i < BarCount; i++ )

{
```

```
// tính giá trị của hệ số làm mịn Factor = 2/(Period[ i ] + 1 );


// tính giá trị của phần tử thứ i của mảng

// sử dụng thanh này close ( close[ i ] ) và giá trị trung bình trước đó ( vaexp[ i - 1 ] )

vaexp[ i ] = Factor * Close[ i ] + ( 1 - Factor ) * vaexp[ i - 1 ];

}
```

Như bạn thấy, đoạn mã dài hơn nhưng mặt khác, nó rất giống với bất kỳ ngôn ngữ lập trình nào khác như C/Pascal/Basic. Vì vậy, những người có một số kinh nghiệm lập trình có thể thấy dễ nắm bắt hơn.

Nếu bạn là người mới bắt đầu, tôi khuyên bạn nên học xử lý mảng trước khi tìm hiểu sâu hơn về các vòng lặp phức tạp hơn.

Nếu bạn gặp sự cố khi mã hóa AFL, tôi khuyên bạn nên tự tạo các mảng trong ví dụ trong Excel. Nếu gặp sự cố, hãy nhờ bạn bè giúp đỡ - đặc biệt nếu người bạn đó là một kế toán.

Khi đã nắm bắt được, bạn có thể mã hóa bất kỳ hệ thống nào từ một cuốn sách về giao dịch - hoặc tự xây dựng một hệ thống.

--- Xin chân thành cảm ơn Geoff Mulhall vì bài viết gốc trong bản tin làm cơ sở cho hướng dẫn này ---

Tạo chỉ báo của riêng bạn

Có hai cách để tạo chỉ báo của riêng bạn:

1) sử dụng giao diện kéo và thả

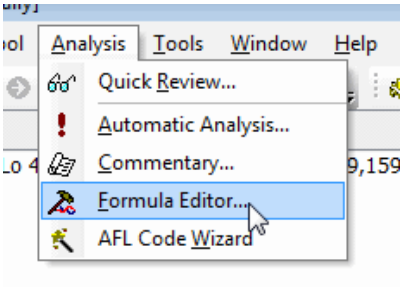
2) bằng cách viết công thức của riêng bạn

Phương pháp đầu tiên, sử dụng giao diện kéo và thả rất đơn giản và không yêu cầu viết một dòng mã nào. Để tìm hiểu thêm về cách tạo chỉ báo kéo và thả, vui lòng xem Hướng dẫn: Cách sử dụng giao diện biểu đồ kéo và thả

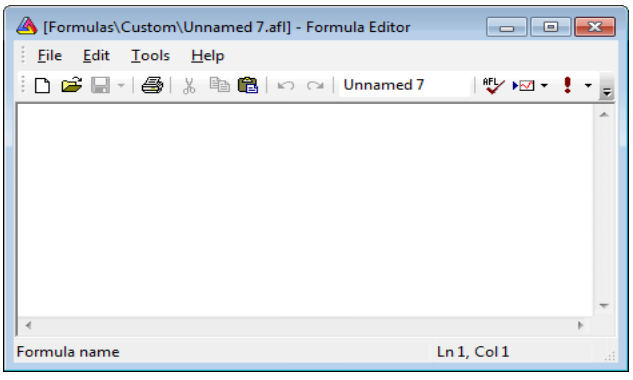
Phương pháp thứ hai liên quan đến việc viết công thức chỉ báo bằng AFL (Ngôn ngữ công thức AmiBroker) linh hoạt. Bạn có thể tìm thấy mô tả về ngôn ngữ này trong phần Hướng dẫn tham khảo AFL của hướng dẫn sử dụng. Ở đây, chúng tôi sẽ trình bày các bước cơ bản cần thiết để xác định và hiển thị chỉ báo tùy chỉnh của riêng bạn. Trong ví dụ này, chúng tôi sẽ xác định một "chỉ báo" sẽ hiển thị biểu đồ khối lượng đường (ngược lại với biểu đồ khối lượng thanh tích hợp).

Chỉ cần làm theo các bước sau

1. Chọn tùy chọn Analysis->Formula Editor từ menu như hiển thị bên dưới:

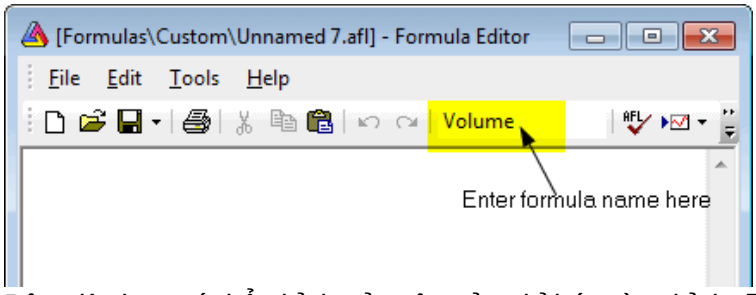


2. Bạn sẽ thấy hộp thoại sau hiển thị trên màn hình:

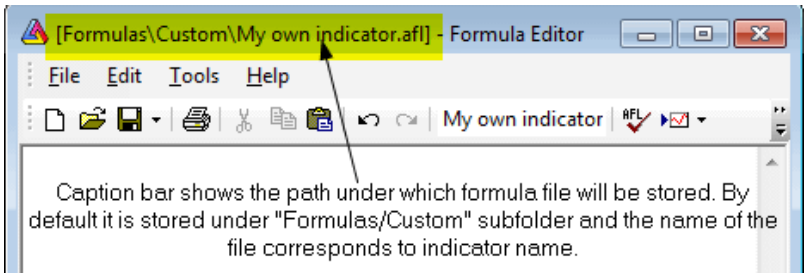


Nó hiển thị một cửa sổ Formula Editor trống.

3. Bây giờ, nhấp một lần vào trường chỉnh sửa nằm trong thanh công cụ của trình chỉnh sửa để thay đổi tên của chỉ báo:



Bây giờ, bạn có thể chỉnh sửa tên của chỉ báo tùy chỉnh. Đặt tên cho nó là "My own indicator". Sau khi bạn nhấn phím ENTER, chú thích sẽ được cập nhật bằng tên tệp mới như hiển thị bên dưới:



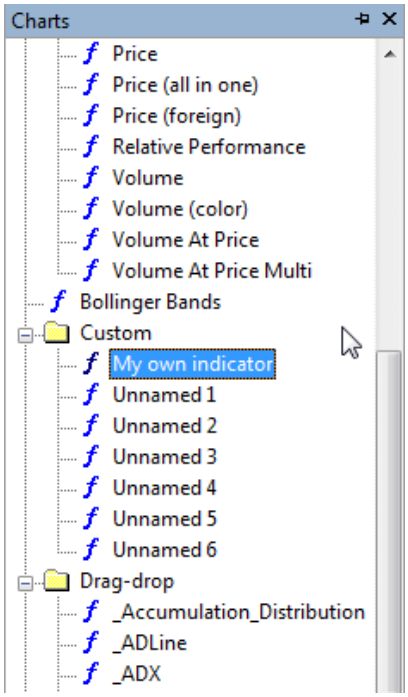
4. Bây giờ, hãy nhập công thức:

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

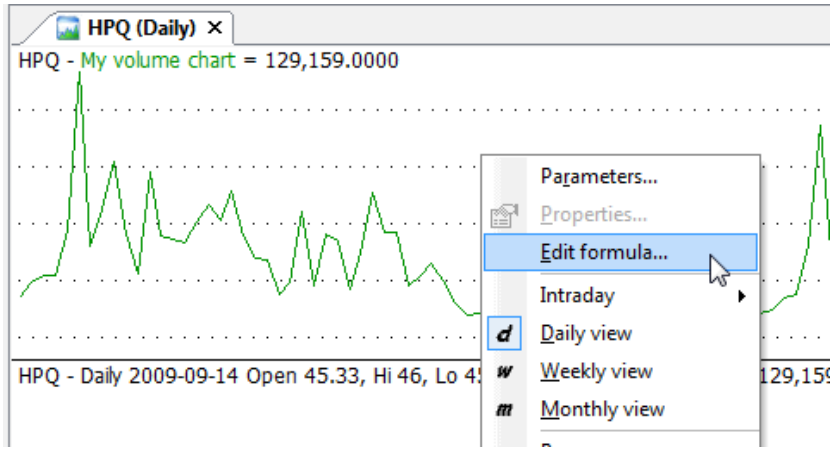
Công thức này hướng dẫn AmiBroker vẽ mảng Volume tích hợp. Tham số thứ hai chỉ định tiêu đề của biểu đồ và tham số thứ ba xác định màu. Hình ảnh bên dưới hiển thị trình chỉnh sửa công thức sau khi nhập mã:

5. Bây giờ, hãy nhấp vào nút thanh công cụ Apply indicator (hoặc chọn Tools->Apply indicator menu) như hiển thị trong hình và đóng trình chỉnh sửa bằng cách nhấn nút X ở góc trên bên phải của cửa sổ trình chỉnh sửa.

Bây giờ, chỉ báo bạn vừa viết được hiển thị dưới dạng biểu đồ. Bạn cũng có thể tìm thấy nó được lưu trữ dưới dạng công thức trong Chart tree:



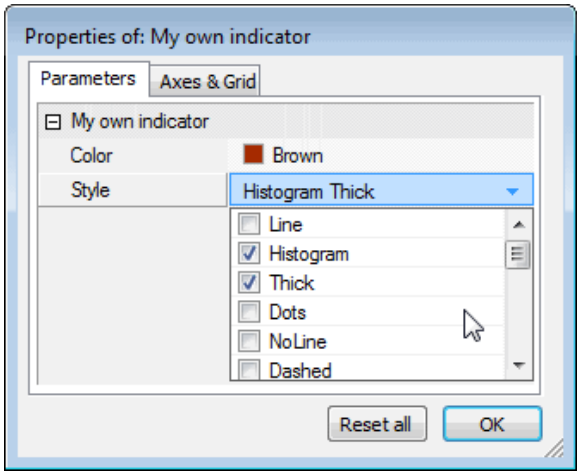
Bây giờ bạn có thể cải thiện chỉ báo của mình bằng cách thêm các hàm Param để cả màu sắc và kiểu dáng của biểu đồ có thể được sửa đổi bằng hộp thoại Parameters. Để thực hiện, hãy nhấp chuột PHẢI vào khung biểu đồ và chọn Edit Formula (hoặc nhấn Ctrl+E)



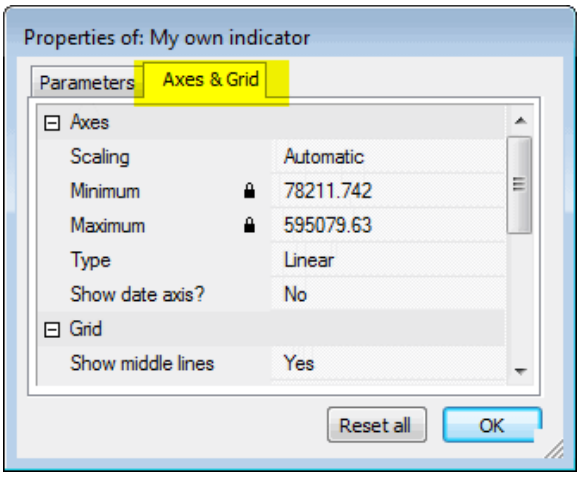
Và sửa đổi công thức thành:

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

Sau đó nhấn Apply indicator để áp dụng các thay đổi. Bây giờ, nhấp chuột PHẢI vào khung biểu đồ một lần nữa và chọn Parameters (hoặc nhấn Ctrl+R) và bạn sẽ thấy hộp thoại tham số cho phép sửa đổi màu sắc và kiểu dáng được sử dụng để vẽ biểu đồ:



Ngoài ra, trong tab "Axes & Grid", bạn sẽ có thể thay đổi cài đặt cho trục, lưới và các tùy chọn biểu đồ khác liên quan đến biểu đồ cụ thể này:



Để biết thêm thông tin về cách tạo chỉ báo của bạn, vui lòng kiểm tra phần hướng dẫn Sử dụng kiểu dáng và màu sắc biểu đồ

Để tham khảo thêm về cách sử dụng Formula Editor, vui lòng tham khảo phần Environment - Formula Editor và AmiBroker Formula Language - AFL Tools của hướng dẫn sử dụng AmiBroker và sử dụng AFL editor.

Sử dụng kiểu dáng, màu sắc, tiêu đề và tham số biểu đồ trong Indicators

AmiBroker cung cấp các kiểu dáng và màu sắc biểu đồ tùy chỉnh trong các chỉ báo tùy chỉnh. Các tính năng này cho phép linh hoạt hơn trong việc thiết kế các chỉ báo của bạn. Bài viết này sẽ giải thích cách sử dụng các kiểu và màu. Bài viết cũng sẽ giải thích cách xác định tiêu đề biểu đồ xuất hiện ở đầu biểu đồ.

Hàm Plot()

Plot là hàm được sử dụng để vẽ biểu đồ. Hàm này lấy 9 tham số, trong đó 3 tham số đầu tiên là bắt buộc.

Plot(array, name, color, style = styleLine, minvalue = Null, maxvalue = Null, XShift = 0, ZOrder = 0, width = 1)

· tham số mảng biểu diễn dữ liệu cần vẽ,

· tham số name xác định tên của biểu đồ (được sử dụng trong chuỗi tiêu đề để hiển thị các giá trị của chỉ báo),

· tham số color xác định màu của biểu đồ,

· style xác định "giao diện" của biểu đồ (tức là đường/biểu đồ histogram/nến/thanh, v.v.). Kiểu mặc định là đường.

· minvalue và maxvalue là các tham số hiếm khi được sử dụng để xác định các giá trị tối thiểu và tối đa được mã hóa cứng khi biểu đồ sử dụng tỷ lệ "độc lập", tức là styleOwnScale được chỉ định trong tham số style. Thông thường, bạn không cần phải chỉ định chúng.

· XShift cho phép dịch chuyển biểu đồ qua thanh cuối cùng (ví dụ: đường trung bình động dịch chuyển hoặc phép chiếu vào tương lai)

· ZOrder - xác định vị trí trục Z của biểu đồ đã cho. Mặc định là số không. Zorder = 0 cũng có nghĩa là vị trí của "lưới". Vì vậy, nếu bạn muốn vẽ biểu đồ PHÍA SAU lưới, bạn cần chỉ định tham số zorder âm. Biểu đồ được vẽ theo thứ tự sau:

Tham số zorder được ưu tiên hơn thứ tự gọi các hàm Plot(), vì vậy nếu z-order được đặt, nó sẽ xác định thứ tự vẽ. Xem hình ảnh này. Nếu nhiều biểu đồ sử dụng cùng một tham số z-order, chúng sẽ được vẽ theo thứ tự gọi ngược (những biểu đồ xuất hiện cuối cùng trong mã sẽ được vẽ đầu tiên). Quy tắc này có thể được thay đổi bằng công tắc graphzorder = 1 hiện có, khi được chỉ định, sẽ đảo ngược hành vi này (do đó, các biểu đồ được vẽ theo thứ tự gọi). Xin lưu ý rằng phần trên áp dụng riêng cho từng "lớp" zorder (do đó, trong cùng một "lớp" zorder, quy tắc gọi ngược được áp dụng)

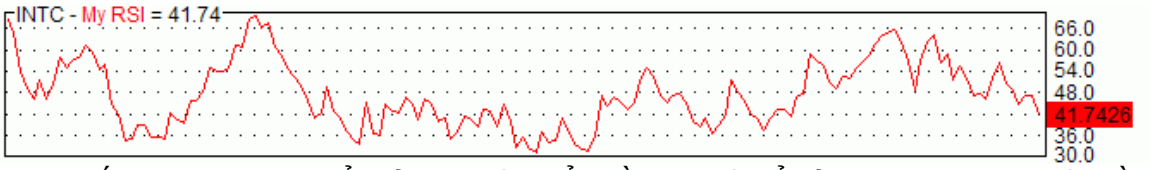
· width - (mới trong 5.60) xác định pixel hoặc phần trăm chiều rộng của biểu đồ đã cho. Mặc định là 1 pixel. Các giá trị dương

chỉ định chiều rộng PIXEL, các giá trị âm chỉ định chiều rộng theo phần trăm chiều rộng thanh hiện tại. Vì vậy, ví dụ -20 sẽ cung cấp cho bạn chiều rộng động bằng 20% chiều rộng thanh.

Ví dụ, lệnh gọi hàm đơn sau đây vẽ chỉ báo RSI bằng đường màu đỏ:

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

Như bạn thấy, chúng tôi chỉ cung cấp ba tham số đầu tiên (bắt buộc). Tham số đầu tiên là mảng chúng ta cần vẽ. Trong ví dụ của chúng tôi, đó là chỉ báo RSI(14). Tham số thứ hai chỉ là tên. Tên có thể là bất kỳ tên nào bạn muốn. Tên sẽ được hiển thị ở dòng tiêu đề cùng với giá trị chỉ báo như trong hình bên dưới:



Tham số thứ ba là màu. Để chỉ định màu biểu đồ, bạn có thể sử dụng một trong các hằng số được xác định trước sau:

Hằng số màu

Màu tùy chỉnh đề cập đến bảng màu do người dùng xác định có thể chỉnh sửa bằng Tools->Preferences->Colors, các giá trị số xuất hiện sau dấu = (phương trình) chỉ để tham khảo và bạn không cần sử dụng chúng. Chỉ cần sử dụng tên như colorDarkGreen.

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

Bạn có thể dễ dàng vẽ biểu đồ nhiều màu bằng cả hai hàm Plot. Tất cả những gì bạn cần làm là xác định mảng chỉ số màu.

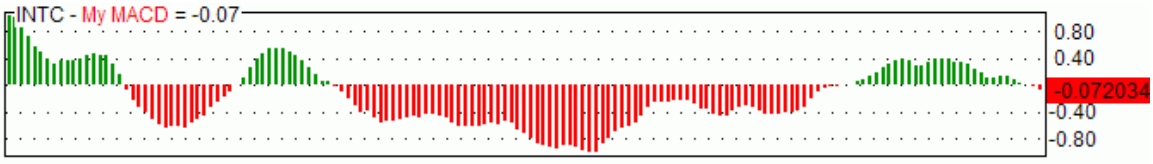
Trong ví dụ sau, MACD được vẽ bằng màu xanh lá cây khi nó lớn hơn 0 và bằng màu đỏ khi nó nhỏ hơn 0.

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

Ngoài việc xác định màu, chúng ta có thể cung cấp tham số thứ 4 xác định kiểu biểu đồ. Ví dụ, chúng ta có thể thay đổi biểu đồ MACD trước đó thành biểu đồ histogram dày thay vì đường:

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

Như bạn thấy, nhiều kiểu có thể được kết hợp với nhau bằng toán tử | (nhị phân hoặc). (Lưu ý: ký tự | có thể được nhập bằng cách nhấn phím gạch chéo ngược '\' trong khi giữ phím SHIFT). Biểu đồ kết quả trông như thế này:



Để vẽ biểu đồ nến, chúng ta sử dụng hằng số styleCandle, như trong ví dụ này:

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

Để vẽ các thanh truyền thống có màu (thanh màu xanh lá cây tăng và thanh màu đỏ giảm), chúng ta chỉ cần chỉ định màu tùy thuộc vào mối quan hệ giữa giá mở và giá đóng và styleBar trong đối số style:

```
Plot( Close, "Price", IIf( Close > Open, colorGreen, colorRed ),
styleBar | styleThick );
```

Tất cả các hằng số kiểu có sẵn được tóm tắt trong bảng bên dưới.

Hằng số kiểu

Kiểu được định nghĩa là sự kết hợp (sử dụng toán tử cộng (+) hoặc toán tử nhị phân hoặc (|)) của một hoặc nhiều cờ sau (bạn có thể sử dụng hằng số kiểu được xác định trước thay vì số)

styleLine = 1 - biểu đồ (đường) bình thường (mặc định) styleHistogram = 2 - biểu đồ histogram styleThick = 4 - fat (dày)

styleDots = 8 - bao gồm các chấm styleNoLine = 16 - không có đường styleDashed = 32 - kiểu đường đứt nét styleCandle = 64 - biểu đồ nến styleBar = 128 - biểu đồ thanh truyền thống

styleNoDraw = 256 - không vẽ (chỉ thực hiện tỷ lệ trục) styleStaircase = 512 - biểu đồ cầu thang (hình vuông) styleSwingDots = 1024 - các chấm ở giữa cho biểu đồ cầu thang styleNoRescale = 2048 - không thay đổi tỷ lệ

styleNoLabel = 4096 - không có nhãn giá trị stylePointAndFigure = 8192 - điểm và hình styleArea = 16384 - biểu đồ diện tích (biểu đồ histogram cực rộng)

styleOwnScale = 32768 - biểu đồ đang sử dụng tỷ lệ độc lập

styleLeftAxisScale = 65536 - biểu đồ đang sử dụng tỷ lệ trục trái (độc lập với trục phải)
styleNoTitle = 131072 - không bao gồm giá trị biểu đồ này trong chuỗi tiêu đề

styleCloud = 262144 - vẽ biểu đồ "đám mây" (khu vực được tô màu) (xem ví dụ bên dưới)

styleClipMinMax = 524288 - cắt khu vực giữa các mức Min và Max được xác định trong câu lệnh Plot. (Lưu ý: kiểu này không tương thích với hầu hết các máy in)

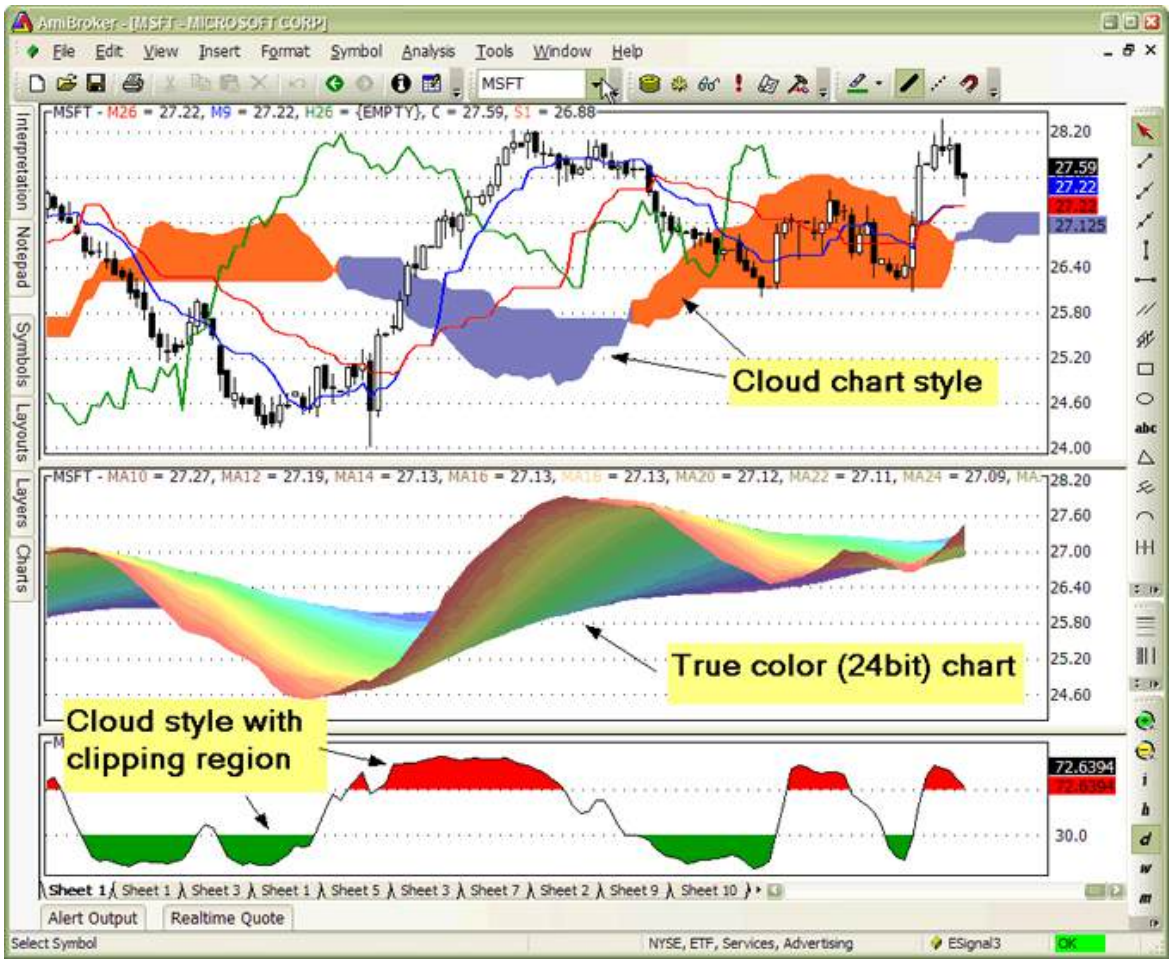
styleGradient - (mới trong 5.60) - biểu đồ khu vực gradient. Màu gradient trên được chỉ định bởi tham số color trong hàm Plot(), màu gradient dưới là màu nền hoặc có thể được xác định bằng hàm SetGradientFill. styleGradient có thể kết hợp với styleLine

Không phải tất cả các kết hợp cờ đều có ý nghĩa, ví dụ (64+1) (nến + đường) sẽ tạo ra biểu đồ nến (style=64)

Lưu ý về biểu đồ nến/thanh: nếu các kiểu này được áp dụng cho hàm Plot() thì chúng sẽ gián tiếp sử dụng các mảng O, H, L.

Nếu bạn muốn chỉ định các giá trị OHL của riêng mình, bạn cần sử dụng hàm PlotOHLC().

Các kiểu styleCloud và styleClipMinMax mới mang đến những khả năng thú vị mới được hiển thị trong hình ảnh mẫu bên dưới:



Công thức cho biểu đồ ở ngăn giữa (trung bình động 24 bit cầu vồng) trông như sau:

```
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("Hiệu ứng 3D?", "Không|Có", 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("Độ bão hòa", 128, 0,
255 ),

side * Param("Độ sáng", 255, 0, 255 ) ), styleCloud | styleNoLabel

);

}
```

Công thức cho biểu đồ trong ngăn bên dưới (%K ngẫu nhiên chậm với đỉnh và đáy có màu) trông như sau. Công thức này sử dụng styleClipMinMax để đạt được mục đích cắt vùng đám mây giữa các mức min và max được chỉ định trong câu lệnh plot. Nếu không có vùng kiểu này giữa min/max sẽ được điền. Xin lưu ý rằng do giới hạn của Windows GDI, vùng cắt (styleClipMinMax) chỉ được hỗ trợ trên các thiết bị raster (bitmap) nên không tương thích với máy in hoặc đầu ra WMF (tệp siêu dữ liệu windows).

```
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);
```

Tính năng dịch chuyển X

Tham số XShift cho phép dịch chuyển (dịch chuyển) biểu đồ theo hướng ngang theo số thanh được chỉ định. Điều này cho phép vẽ các đường trung bình động dịch chuyển và các phép chiếu vào tương lai. Xem mã mẫu sau đây của đường trung bình động dịch chuyển:

```
Periods = Param("Periods", 30, 2, 100 );

Displacement = Param("Displacement", 15, -50, 50 );
```

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

Hàm PlotForeign()

Giờ đây, bạn có thể dễ dàng chồng các biểu đồ giá của nhiều ký hiệu bằng hàm PlotForeign:

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

maxvalue = {empty}, xshift = 0)

Vẽ biểu đồ giá ký hiệu nước ngoài (ký hiệu được xác định bởi tham số tickersymbol). Đối số thứ hai name xác định tên biểu đồ được sử dụng để hiển thị các giá trị trong thanh tiêu đề. Màu biểu đồ có thể là tĩnh (nếu đối số thứ ba là số) hoặc động (khi đối số thứ ba là mảng). Chỉ số màu liên quan đến bảng màu hiện tại (xem Tùy chọn/Màu)

style xác định kiểu biểu đồ (xem hàm Plot() để biết các giá trị có thể)

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

Nhiều biểu đồ sử dụng tỷ lệ khác nhau

Có thể sử dụng hai kiểu mới để vẽ nhiều biểu đồ sử dụng tỷ lệ Y khác nhau: styleOwnScale và styleLeftAxisScale.

Nó cũng giúp dễ dàng vẽ 2 hoặc nhiều biểu đồ "tỷ lệ riêng" với cùng tỷ lệ:

minimum = LastValue(Lowest(Volume));

maximum = LastValue(Highest(Volume));

Plot(Close, "Price", colorBlue, styleCandle);

/* hai biểu đồ bên dưới sử dụng OwnScale nhưng tỷ lệ này là chung vì chúng ta đặt giá trị min và max của trục Y */

Plot(Volume, "Volume", colorGreen, styleHistogram | styleThick | styleOwnScale, minimum, maximum);

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

Kiểu mới: styleLeftAxisScale = 65536 - cho phép vẽ nhiều hơn một đồ thị bằng cách sử dụng tỷ lệ chung nhưng khác với tỷ lệ thông thường (trục phải).

Ví dụ: biểu đồ giá cộng với khối lượng và biểu đồ trung bình động:

```
// Vẽ biểu đồ giá và trung bình động của nó
Plot( Close, "Price", colorWhite, styleCandle );
Plot( MA( Close, 20 ), "MAC", colorRed );

// Bây giờ vẽ khối lượng và trung bình động của nó bằng cách sử dụng tỷ lệ trục bên trái
Plot( Volume , "Volume", colorBlue, styleLeftAxisScale |
styleHistogram | styleThick );
Plot( MA( Volume,15), "MAV", colorLightBlue, styleLeftAxisScale );
```

Các tham số mới cũng giúp vẽ ruy băng dễ dàng hơn, ví dụ:

```
Plot( Close, "Price", colorBlue, styleCandle );
Plot( 2, /* định nghĩa chiều cao của ruy-băng theo phần trăm chiều rộng của khung
*/ "Ribbon",
IIf( up, colorGreen, IIf( down, colorRed, 0 )), /* chọn màu */
styleOwnScale|styleArea|styleNoLabel, -0.5, 100 );
```

Sử dụng các tham số tùy chỉnh

AmiBroker cho phép tạo các tham số do người dùng định nghĩa. Các tham số như vậy sau đó có sẵn thông qua hộp thoại Tham số để điều chỉnh chỉ báo nhanh chóng và dễ dàng.

Các hàm tham số được sử dụng thường xuyên nhất là (nhấp vào liên kết để biết mô tả chi tiết hơn):

· Param("name", default, min, max, steps, incr = 0)

· ParamStr("name", "default");

· ParamColor("name", defaultcolor);

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

Chúng giúp bạn có thể định nghĩa các tham số của riêng mình trong các chỉ báo. Sau khi các hàm Param được đưa vào công thức, bạn có thể nhấp chuột phải vào khung biểu đồ và chọn "Tham số" hoặc nhấn Ctrl+R, rồi thay đổi chúng thông qua hộp thoại Tham số và nhận được phản hồi ngay lập tức.

Trường hợp đơn giản nhất trông như thế này:

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

Plot( RSI( period ), "RSI( " + period + ") ", colorRed );
```

Nhấp chuột phải vào biểu đồ và chọn "Parameters" rồi di chuyển thanh trượt, bạn sẽ thấy RSI được vẽ với các chu kỳ khác nhau ngay khi bạn di chuyển thanh trượt.

Mã mẫu bên dưới cho thấy cách sử dụng ParamStr để lấy ký hiệu ticker và ParamColor để lấy màu.

```
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 ) );
```

Công thức mẫu sau đây (từ danh sách gửi thư AmiBroker) cho phép căn chỉnh trực quan đỉnh/đáy giá với đường cong sin trên biểu đồ:

```
Chu kỳ = Param("Chu kỳ tháng", 12, 1, 12, 1 )*22;//264==12mth,22==1mth xfactor = Param("Stretch",1,0.1,2,0.1);//1==1yr,2==2yr

xshift = Param("slide",0,-22,22,2)/3.1416^2;//slide curve 1==5days

x = 2*3.1416/Chu kỳ/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);
```

Nhấp chuột phải vào biểu đồ và chọn "Parameters" rồi di chuyển thanh trượt, bạn sẽ thấy biểu đồ phản ánh ngay những thay đổi của mình.

Để biết thêm thông tin về các tham số do người dùng định nghĩa, vui lòng xem thêm Hướng dẫn: Sử dụng giao diện kéo và thả

Vẽ văn bản ở các vị trí tùy ý trên biểu đồ

AmiBroker hiện cho phép chú thích biểu đồ bằng văn bản được đặt ở bất kỳ vị trí x, y nào được chỉ định ở cấp công thức bằng hàm PlotText mới.

PlotText("text", x, y, color, bkcolor = colorDefault) trong đó

x - là tọa độ x theo thanh (như trong LineArray)

y - là tọa độ y theo đô la

color là màu văn bản, bkcolor là màu nền. Nếu bkcolor KHÔNG được chỉ định (hoặc bằng colorDefault), văn bản được viết bằng nền TRONG SUỐT, bất kỳ giá trị nào khác sẽ tạo ra nền liền với màu nền được chỉ định

Ví dụ:

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));

Tô màu nền

AmiBroker 4.90 cho phép tô màu nền chỉ báo bằng màu thay đổi dần dần. Để thực hiện điều này, bạn cần sử dụng hàm mới SetChartBkGradientFill(topcolor, bottomcolor, titlebkcolor = default)

Hàm này cho phép tô màu nền cho các chỉ báo.

Xin lưu ý rằng điều này không phụ thuộc vào màu nền biểu đồ (màu nền tô toàn bộ khung, tô màu nền chỉ dành cho phần bên trong biểu đồ thực tế, do đó vùng trục không bị ảnh hưởng bởi tô màu nền). Các tham số như sau:

topcolor - chỉ định màu trên cùng của tô màu nền gradient bottomcolor - chỉ định màu dưới cùng của tô màu nền gradient

titlebkcolor - (tùy chọn) màu nền của văn bản tiêu đề. Nếu không chỉ định thì màu trên cùng sẽ tự động được sử dụng cho nền tiêu đề.

Ví dụ:

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

Biểu đồ vùng tô màu gradient

Phiên bản 5.60 mang đến biểu đồ vùng tô màu gradient gốc. Để hiển thị biểu đồ gradient đơn giản, chỉ cần sử dụng styleGradient trong lệnh gọi hàm Plot(). Theo mặc định, màu gradient trên được chỉ định bởi tham số color trong hàm Plot(), màu gradient dưới là màu nền. styleGradient có thể được kết hợp với styleLine.

Có thể hiển thị biểu đồ vùng tô màu gradient đơn giản bằng cách sử dụng:

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

Để kiểm soát chi tiết màu gradient và đường cơ sở, có một hàm bổ sung SetGradientFill(topcolor, bottomcolor, baseline, baselinecolor) cần được gọi trước Plot().

Khi bạn sử dụng hàm SetGradientFill, màu gradient trên được chỉ định bởi đối số topcolor, màu gradient dưới được chỉ định bởi bottomttomcolor. Các tham số tùy chọn (baseline/baselinecolor) cho phép biểu đồ gradient ngược (như công bằng dưới nước) và 3 gradient màu top->baseline->bottom. Xem mã cho Underwater Equity để biết ví dụ về cách sử dụng biểu đồ gradient ngược (với baseline ở trên cùng). Tham số baseline chỉ định vị trí

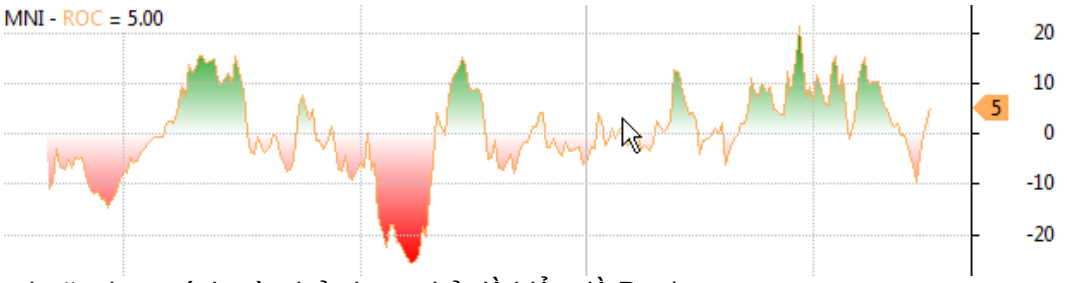
trục Y của đường cơ sở biểu đồ. Tham số baselinecolor chỉ định màu của gradient sẽ được sử dụng ở cấp độ đó. Nếu baselinecolor không được chỉ định, thì chỉ có gradient 2 màu được vẽ (topcolor->bottomcolor).

Ví dụ để hiển thị Tỷ lệ thay đổi gradient ba màu sẽ sử dụng màu xanh lá cây làm màu "trên cùng" cho các giá trị dương, màu nền làm màu "baseline" và màu đỏ làm màu "dưới cùng" cho các giá trị âm, chỉ cần viết:

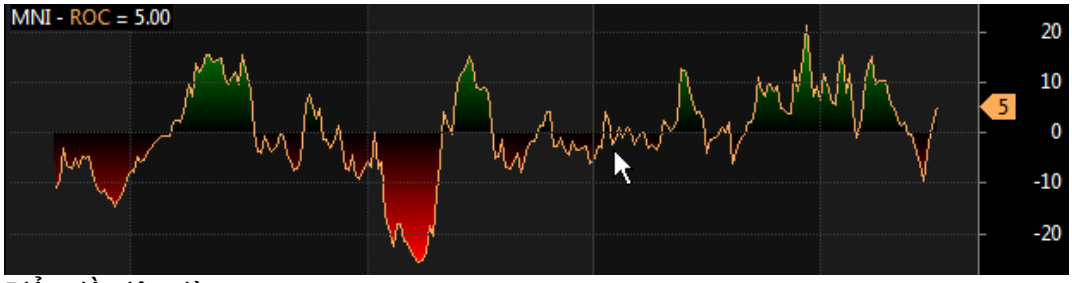
```
SetGradientFill( colorGreen /*top*/, colorRed /*bottom*/, 0 /*baseline level*/, GetChartBkColor() /*baseline color */);

Plot( ROC( C, 14), "ROC", colorLightOrange, styleLine | styleGradient, Null, Null, 0, -1);
```

Biểu đồ kết quả sẽ trông như sau (sử dụng chủ đề biểu đồ Cơ bản):



.. hoặc theo cách này (sử dụng chủ đề biểu đồ Đen):



Biểu đồ siêu dày

Phiên bản 5.60 cho phép xác định độ rộng đường vượt quá styleThick, trước đây là tùy chọn duy nhất.

Bây giờ, tham số thứ 9 của Plot() xác định chiều rộng pixel hoặc phần trăm của biểu đồ đã cho. Mặc định là 1 pixel. Các giá trị dương chỉ định chiều rộng pixel, các giá trị âm chỉ định chiều rộng theo phần trăm chiều rộng thanh hiện tại. Vì vậy, ví dụ -20 sẽ cung cấp cho bạn chiều rộng động bằng 20% chiều rộng thanh. Ví dụ:

Plot(C, "Close", colorDefault, styleBar, Null, Null, 0, 1, -20 /* độ rộng đường tính theo phần trăm của thanh */);

Khi bạn phóng to, các thanh sẽ ngày càng dày hơn.

Bây giờ, bạn có thể có các đường siêu dày như trong ví dụ bên dưới (biểu đồ đường dày 10 pixel):

Plot(C, "Close", colorRed, styleLine, Null, Null, 0, 1, 10 /* rộng 10 pixel */

);

Khác

Như bạn đã biết, mỗi biểu đồ có tên riêng được sử dụng để tạo chuỗi tiêu đề hiển thị tên và giá trị của các chỉ báo. Tuy nhiên, AmiBroker cho phép bạn ghi đè cơ chế tự động này và xác định chuỗi tiêu đề của riêng bạn từ đầu. Biến dành riêng Title được sử dụng cho mục đích đó. Bạn chỉ cần gán một chuỗi cho nó và nó sẽ được hiển thị trong biểu đồ thay vì chuỗi được tạo tự động.

Ngoài ra còn có hai biến dành riêng nữa (GraphXSpace và GraphZOrder) cho phép tinh chỉnh giao diện của chỉ báo.

Tất cả đều được mô tả trong bảng bên dưới.

Biến	Sử dụng	Áp dụng cho
Tiêu đề	Xác định văn bản tiêu đề Nếu bạn sử dụng biến Tiêu đề, bạn có thể chỉ định màu trong chuỗi. Nên sử dụng hàm AFL EncodeColor giúp việc này dễ dàng hơn so với mã hóa chuỗi thoát. EncodeColor(colornumber). Và bạn có thể viết ví dụ trên như sau: Title = "Điều này được viết bằng " + EncodeColor(colorViolet) + "violet color "	Chỉ báo

	+ EncodeColor(colorGreen) + "và điều này bằng màu xanh lá cây"; Có thể thêm chú thích nhiều dòng chỉ bằng cách nhúng ngắt dòng \n, ví dụ: Title = "Đây là dòng đầu tiên\nĐây là dòng thứ hai"; Để hoàn thiện: màu sắc cũng có thể được chỉ định bằng chuỗi espace nhưng KHÔNG được khuyến khích vì khó viết và khó đọc. \\cXX chuỗi trong đó XX là số 2 chữ số chỉ định chỉ số màu \\c38 - định nghĩa màu tím, có một chuỗi đặc biệt \\c-1 đặt lại thành màu trục mặc định. Ví dụ Title = "This is written in \\c38violet color \\c27and this in green";	
Tooltip	Obsoleted in 5.40. Sử dụng cửa sổ Data thay thế hoặc sử dụng Plot() với styleHidden nếu bạn muốn thêm các giá trị tùy chỉnh của mình vào tooltip dữ liệu. Ví dụ: Plot(my_value, "MyValueForTooltip", colorBlack, styleHidden);	Chỉ báo
GraphXSpace	xác định lượng không gian bổ sung cần được thêm vào phía trên và phía dưới đường biểu đồ (theo phần trăm). Ví dụ: GraphXSpace = 5; thêm 5% không gian bổ sung phía trên và phía dưới đường biểu đồ. Khi GraphXSpace không được xác định trong công thức thì mặc định sẽ sử dụng 2%.	Chỉ báo
GraphLabelDecimals	(mới trong 5.90) kiểm soát số lượng chữ số thập phân	Chỉ báo

	trong nhãn giá trị biểu đồ (ví dụ, thêm GraphLabelDecimals = 2; vào công thức sẽ cung cấp cho bạn nhãn giá trị có 2 chữ số thập phân)	
GraphZOrder	Biến GraphZOrder cho phép thay đổi thứ tự của các đường chỉ báo vẽ. Khi GraphZOrder không được xác định hoặc bằng 0 (sai) - thứ tự cũ (từ cuối đến đầu) được sử dụng, khi GraphZOrder bằng 1 (đúng) - thứ tự ngược lại được áp dụng.	Chỉ báo

Biến đồ thị lỗi thời

Bảng này hiển thị các biến được đặt trước đã lỗi thời. Chúng vẫn hoạt động để tương thích ngược nhưng mã mới chỉ nên sử dụng các hàm Plot(). Hơn nữa, khi sử dụng các hàm Plot() mới, bạn KHÔNG NÊN sử dụng các biến lỗi thời bên dưới.

Biến	Sử dụng	Áp dụng cho
maxgraph	chỉ định số lượng đồ thị tối đa được vẽ trong cửa sổ chỉ báo tùy chỉnh (mặc định=3)	Chỉ báo
graphN	xác định công thức cho số đồ thị N (trong đó N là số 0,1,2,..., maxgraph-1)	Chỉ báo
graphNopen, graphNhigh, graphNlow,	xác định các mảng giá O, H, L bổ sung cho biểu đồ nến và biểu đồ thanh truyền thống	Chỉ báo
graphNcolor	xác định chỉ số màu của đường đồ thị thứ N. Chỉ số màu liên quan đến bảng màu hiện tại - xem Tùy chọn/Màu.	Chỉ báo
graphNbarcolor	xác định mảng chứa chỉ số bảng màu cho mỗi thanh được vẽ	Chỉ báo
graphNstyle	xác định kiểu của đồ thị thứ N. Kiểu được xác định là sự kết hợp (tổng) của một hoặc nhiều cờ sau (bạn có thể sử dụng hằng số kiểu được xác định trước thay vì số)	Chỉ báo
		Chỉ báo

Cách tạo bản khám phá của riêng bạn

Một trong những tính năng hữu ích nhất của cửa sổ Phân tích được gọi là "Khám phá". Về cơ bản, một bản khám phá hoạt động theo cách tương tự như quét nhưng thay vì chỉ tìm kiếm và báo cáo các tín hiệu mua/bán, nó cho phép bạn tạo báo cáo sàng lọc (hoặc khám phá) có thể tùy chỉnh, có thể cung cấp cho bạn nhiều thông tin hơn so với quét đơn giản.

Ý tưởng đằng sau một bản khám phá rất đơn giản - một biến được gọi là bộ lọc kiểm soát các ký hiệu/báo giá nào được chấp nhận. Nếu "đúng" (hoặc 1) được gán cho biến đó cho ký hiệu/báo giá đã cho, nó sẽ được hiển thị trong báo cáo.

Vì vậy, ví dụ, công thức sau sẽ chấp nhận tất cả các ký hiệu có giá đóng cửa lớn hơn 50:

```
bộ lọc = đóng > 50;
```

(LƯU Ý: Để tạo công thức mới, vui lòng mở Trình chỉnh sửa công thức bằng menu Phân tích->Trình chỉnh sửa công thức, nhập công thức và chọn Công cụ->Gửi đến trình chỉnh sửa công thức trong trình chỉnh sửa công thức)

Lưu ý rằng khám phá sử dụng tất cả các cài đặt phạm vi và bộ lọc cũng được sử dụng bởi chế độ kiểm tra ngược và quét để bạn có thể nhận được nhiều tín hiệu (dòng báo cáo) nếu bạn chọn phạm vi "Tất cả báo giá". Để chỉ kiểm tra báo giá gần đây nhất, bạn nên chọn "1 thanh gần đây"

Bây giờ, còn báo cáo tùy chỉnh thì sao?

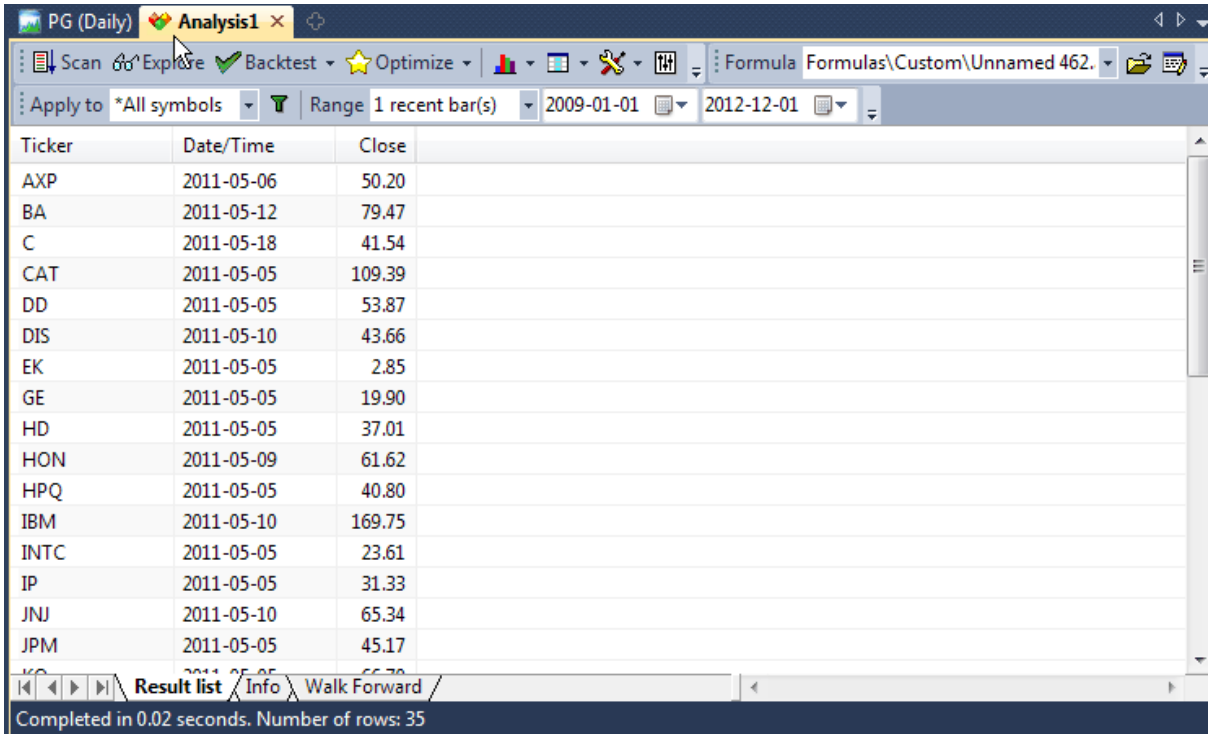
Có, chế độ khám phá cho phép bạn tạo và sau đó xuất báo cáo với các cột có thể tùy chỉnh hoàn toàn và khá đơn giản để thực hiện.

Tất cả những gì bạn phải làm là cho AmiBroker biết bạn muốn cột nào. Bạn có thể thực hiện việc này bằng cách gọi hàm AddColumn trong công thức khám phá của mình:

```
AddColumn( Close, "Close" );
```

Đối số đầu tiên của hàm AddColumn là MẢNG dữ liệu bạn muốn hiển thị, đối số thứ hai xác định tiêu đề cột

Nếu bây giờ bạn nhấn nút "Khám phá" trong cửa sổ Phân tích, bạn sẽ nhận được kết quả tương tự như sau:



Lưu ý rằng thực tế có 3 cột: Cột Ticker và Ngày/Giờ được xác định trước và một cột tùy chỉnh chứa giá đóng cửa. Lưu ý rằng chỉ những mã cổ phiếu có giá đóng cửa lớn hơn 50 mới được báo cáo.

Bây giờ bạn có thể nhấp vào "Xuất" và khám phá của bạn sẽ được lưu vào tệp CSV (giá trị phân tách bằng dấu phẩy) có thể dễ dàng tải vào bất kỳ chương trình nào khác bao gồm Excel để phân tích thêm.

Trên thực tế, hàm AddColumn chấp nhận nhiều đối số hơn để cho phép bạn tùy chỉnh đầu ra nhiều hơn nữa. Cú pháp đầy đủ là:

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

tham số format cho phép bạn xác định định dạng được áp dụng cho các số. Theo mặc định, tất cả các biến được hiển thị với 2 chữ số thập phân, nhưng bạn có thể thay đổi điều này bằng cách gán một giá trị khác cho biến này: 1.5 cho 5 chữ số thập phân, 1.0 không cho chữ số thập phân. Vì vậy, trong ví dụ của chúng tôi, nhập:

AddColumn(Close, "Close", 1.4);

sẽ hiển thị giá đóng cửa với 4 chữ số thập phân.

(Lưu ý dành cho người dùng nâng cao: phần nguyên của số này có thể được sử dụng để đệm số được định dạng bằng khoảng trắng - 6.0 sẽ không cung cấp chữ số thập phân nào nhưng sẽ có số được đệm khoảng trắng lên đến 6 ký tự.)

Ngoài ra còn có các hằng số định dạng đặc biệt được xác định trước cho phép hiển thị ngày/giờ và mã ký tự đơn:

· formatDateTime - tạo ngày giờ được định dạng theo cài đặt hệ thống của bạn

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

· formatChar - cho phép xuất mã ký tự ASCII đơn:

Ví dụ (tạo tệp tín hiệu được nhiều chương trình khác chấp nhận):

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 và các đối số bkgndColor cho phép bạn tạo báo cáo nhiều màu. Theo mặc định, danh sách kết quả được hiển thị bằng màu hệ thống nhưng bạn có thể ghi đè hành vi này bằng cách cung cấp màu của riêng bạn.

Ví dụ: mã hiển thị giá đóng cửa bằng màu xanh lá cây khi tỷ lệ thay đổi 1 ngày là dương và nếu không thì sử dụng màu đỏ:

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

Ví dụ

Chế độ khám phá cực kỳ linh hoạt: ví dụ, bạn có thể xuất toàn bộ cơ sở dữ liệu sang tệp CSV bằng công thức sau:

filter = 1; /* tất cả các ký hiệu và dấu ngoặc kép được chấp nhận */

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);

Công cụ này sẽ chỉ hiển thị cho bạn các chứng khoán được giao dịch nhiều:

filter = volume > 5000000; /* điều chỉnh ngưỡng này theo nhu cầu của riêng bạn */

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

hoặc...chỉ hiển thị các chứng khoán có khối lượng cao hơn 30% so với mức trung bình theo hàm mũ 40 ngày của nó

filter = volume > 1.3 * ema(volume, 40); AddColumn(Close,"Close",1.4); AddColumn(Volume,"Volume",1.0);

Với công cụ này, bạn có thể xuất nhiều giá trị chỉ báo để phân tích thêm:

filter = close > ma(close, 20); /* chỉ các cổ phiếu giao dịch trên MA 20 ngày của nó*/

AddColumn(macd(), "MACD", 1.4);

AddColumn(signal(), "Signal", 1.4);

AddColumn(adx(), "ADX", 1.4);

AddColumn(rsi(), "RSI", 1.4);

AddColumn(roc(đóng, 15), "ROC(15)", 1.4);

AddColumn(obv(), "OBV", 1.4);

AddColumn(cci(), "CCI", 1.4); AddColumn(ultimate(), "Ultimate", 1.4);

Một ví dụ nữa về đầu ra màu:

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));

Biểu đồ phân tán (X-Y) trong Exploration

Phiên bản 5.60 mang đến một tính năng mới cho khám phá - biểu đồ phân tán X/Y. Biểu đồ phân tán hữu ích để hiển thị mối quan hệ giữa nhiều ký hiệu như tương quan, rủi ro, v.v. Chúng có thể được xem như sự thay thế và nâng cấp cho bản đồ "Rủi ro/lợi nhuận" được mã hóa cứng thành chỉ một hàm. Bây giờ bạn có thể mã hóa biểu đồ X-Y của riêng mình không chỉ giới hạn ở bản đồ rủi ro/lợi nhuận.

Tất cả những gì bạn cần làm để hiển thị biểu đồ phân tán của riêng mình là thêm XYChartAddPoint vào công thức của bạn cho mỗi điểm X-Y mà bạn muốn có trên biểu đồ của mình.

Ví dụ, bạn có thể nhận được biểu đồ phân tán của mối quan hệ MFE/Lợi nhuận và MAE/Lợi nhuận như được hiển thị trong phần mô tả của hàm XYChartAddPoint AFL.

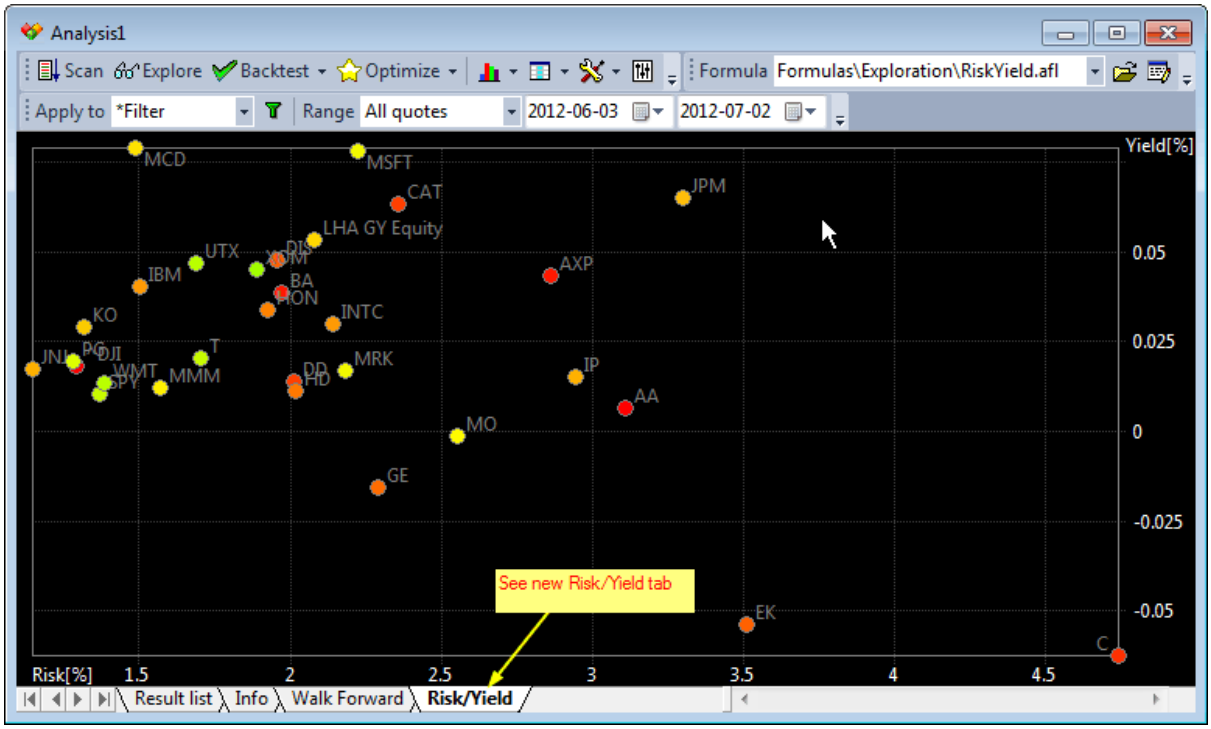
Để hiển thị biểu đồ phân tán rủi ro/lợi nhuận bằng các hàm mới, hãy làm theo các bước dưới đây.

1. Nhấp vào Tệp-> Mới-> Phân tích

2. Chọn tệp "Formulas\Exploration\RiskYield.afl" (được liệt kê bên dưới)

3. Nhấp vào nút Khám phá trong cửa sổ Phân tích mới

4. Ở hàng tab dưới cùng, bạn sẽ thấy tab "Rủi ro/Lợi nhuận" mới, nhấp vào tab đó và bạn sẽ thấy biểu đồ XY được tạo trong quá trình khám phá:



Bạn có thể di chuột qua biểu đồ X-Y đó để đọc các giá trị và bạn có thể nhấp, kéo để đánh dấu hình chữ nhật để phóng to. Nhấp mà không đánh dấu hình chữ nhật sẽ khôi phục chế độ xem đầy đủ.

// Ví dụ về biểu đồ phân tán XY

// Đây là AFL tương đương với bản đồ Rủi ro-Lợi nhuận

// Lưu ý rằng khám phá này phải được chạy trên

// dữ liệu HÀNG TUẦN

// dữ liệu này tính toán mức tăng trung bình hàng tuần (lợi nhuận)

// và độ lệch chuẩn của mức tăng (rủi ro)

```
Filter=Status("lastbarinrange"); Length = SelectedValue( BarIndex() ); Chg = ROC( C, 1 );
//one bar yield 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[%]");
```

Mẹo cuối cùng

Xin đừng quên rằng bạn có thể sắp xếp kết quả của quá trình khám phá theo bất kỳ cột nào chỉ bằng cách nhấp vào tiêu đề của cột đó.

Cách viết bình luận biểu đồ của riêng bạn

Một trong những khía cạnh thú vị khi sử dụng Ngôn ngữ công thức AmiBroker là viết bình luận biểu đồ tự động. Ý tưởng đằng sau kỹ thuật này như sau:

1. Bạn viết công thức bình luận bao gồm hai thành phần cơ bản: văn bản tĩnh và biểu thức AFL

2. AmiBroker đánh giá các biểu thức bằng cách sử dụng dữ liệu ký hiệu hiện đang được chọn và tạo nội dung động

3. Sự kết hợp giữa văn bản tĩnh và các công thức được đánh giá được hiển thị trong cửa sổ đầu ra bình luận

4. Ngoài ra, các mũi tên mua/bán được vẽ trên biểu đồ

Bình luận có sẵn trong menu Phân tích->Bình luận. Khi bạn mở cửa sổ bình luận, bạn sẽ thấy hai tab: Bình luận và Công thức. Trong tab Công thức, bạn có thể nhập các câu lệnh AFL sẽ được AmiBroker đánh giá, tạo ra bình luận động xuất hiện trong tab Bình luận. Các phần sau sẽ hướng dẫn bạn thực hiện các bước cần thiết để viết công thức bình luận của riêng bạn.

Viết văn bản tĩnh

Các thành phần văn bản tĩnh được viết trong công thức phải được đặt trong dấu ngoặc kép và kết thúc bằng dấu chấm phẩy như minh họa bên dưới:

```
"Đây là câu lệnh văn bản tĩnh mẫu";
```

Bạn có thể viết nhiều câu lệnh và mỗi câu lệnh sẽ được đặt trên một dòng mới trong cửa sổ đầu ra chú thích:

```
"Đây là dòng văn bản đầu tiên"; "Đây là dòng văn bản thứ hai";
```

Vui lòng nhập các ví dụ này vào trường chỉnh sửa trong tab Công thức và chuyển sang tab Bình luận. Bạn sẽ thấy các văn bản được hiển thị trong vùng đầu ra nhưng không có dấu ngoặc kép hoặc dấu chấm phẩy. Điều này là do AmiBroker đã đánh giá các câu lệnh văn bản đơn giản này thành chuỗi và hiển thị các chuỗi trong cửa sổ đầu ra.

Thay vì chỉ nhập văn bản, bạn nên sử dụng hàm printf cho bất kỳ mã mới nào.

```
printf( "Đây là câu lệnh văn bản tĩnh mẫu" );
```

Để viết nhiều dòng văn bản, bạn có thể sử dụng một vài câu lệnh như được hiển thị ở trên hoặc bạn có thể thực hiện việc này bằng cách sử dụng một câu lệnh và chuỗi ngắt dòng ('\n'):

printf("Đây là dòng văn bản đầu tiên\nĐây là dòng văn bản thứ hai\nĐây là dòng văn bản thứ ba");

Bạn cũng có thể nối các hằng chuỗi sẽ tạo ra văn bản một dòng:

printf("This" + " is" +

" single"+

" line" + " of text");

Màu sắc và kiểu

Kể từ phiên bản 5.90, cửa sổ chú thích và diễn giải hỗ trợ màu sắc và kiểu in đậm/in nghiêng. Để chỉ định phần đầu và phần cuối của phần in đậm, hãy sử dụng thẻ và . Để chỉ định phần đầu và phần cuối của phần in nghiêng, hãy sử dụng thẻ

 và . Để thay đổi màu văn bản, hãy sử dụng EncodeColor như trong ví dụ dưới đây:

printf("Văn bản in đậm\n"); printf("Văn bản in nghiêng\n");

printf("Now " + EncodeColor(colorRed) + "red text\n");

printf("and finally " + EncodeColor(colorGreen) + "green AND bold AND italic\n");

printf(EncodeColor(colorBlack) + "going back to black");

Nội dung động

Tôi đoán là bạn khá chán với những ví dụ đơn giản này, chúng ta hãy bắt đầu với một số nội dung động.

Để bật bình luận động, AFL có một vài hàm đặc biệt, nhưng hai trong số đó là quan trọng nhất: NumToStr() và WriteIF(). Hàm WriteIF() được sử dụng để hiển thị văn bản có điều kiện và sẽ được mô tả sau trong bài viết này, bây giờ chúng ta hãy xem chúng ta có thể làm gì khi sử dụng hàm NumToStr().

Sổ tay tham khảo AFL có ghi:

CÚ PHÁP NumToStr(NUMBER);

NumToStr(ARRAY);

TRẢ VỀ CHUỖI

HÀM Hàm này chỉ có thể được sử dụng trong bình luận của Guru. Hàm này được sử dụng để hiển thị giá trị số của NUMBER hoặc ARRAY.

Vì vậy, nếu bạn muốn hiển thị giá trị của một số hoặc thanh hiện đang được chọn của mảng, bạn nên sử dụng hàm NumToStr(). Nhưng... hãy đợi một chút - "thanh hiện đang được chọn của mảng" có nghĩa là gì? Để tôi giải thích điều này bằng công thức đơn giản (vui lòng nhập vào tab Công thức):

printf(NumToStr(close));

Khi bạn chuyển sang tab Bình luận, bạn sẽ thấy giá trị của giá đóng cửa (cùng giá được hiển thị ở đầu biểu đồ giá chính). Nhưng khi bạn nhấp vào biểu đồ ở một nơi khác, chọn ngày khác rồi nhấp vào nút "Làm mới", bạn sẽ thấy giá trị khác - giá đóng cửa vào ngày bạn đã chọn. Vì vậy, hàm NumToStr(close) hiển thị giá trị của thanh hiện đang được chọn của mảng đóng. Và nó hoạt động chính xác theo cùng một cách với các mảng khác. Nếu bạn viết

printf(NumToStr(macd()));

bạn sẽ thấy giá trị chính xác của chỉ báo MACD vào ngày bạn đã chọn trong biểu đồ chính. Với hiểu biết hiện tại của mình, chúng ta có thể viết một số số liệu thống kê:

printf("Giá đóng cửa = " + NumToStr(close) + "\n");

printf("Thay đổi kể từ hôm qua = " + NumToStr(đóng - ref(đóng,

-1)) + "\n");

printf("Phần trăm thay đổi kể từ hôm qua = " + NumToStr(roc(đóng, 1)

) + " %%\n");

printf("MACD =" + NumToStr(macd()) + " , Đường tín hiệu =" +

NumToStr(signal()) + "\n");

Khi bạn chuyển sang tab Commentary, bạn sẽ thấy đầu ra tương tự như thế này: Giá đóng cửa = 17.940

Thay đổi kể từ hôm qua = -0.180

Phần trăm thay đổi kể từ hôm qua = -0.993 % MACD = -0.001 , Đường tín hiệu = 0.063

Khá tuyệt phải không? Bạn cũng có thể viết ký hiệu ticker hiện tại và ngày đã chọn bằng cách sử dụng các hàm name() và date() như được hiển thị bên dưới:

printf("Thống kê của " + name() + " tính đến " + date());

Thay vì sử dụng NumToStr để chuyển đổi số thành chuỗi, chúng ta có thể định dạng số trực tiếp bằng cách sử dụng các chỉ định định dạng % linh hoạt printf. Ví dụ, sử dụng %.2f có nghĩa là viết một số có 2 chữ số thập phân, %.3f sẽ có nghĩa là viết một số có 3 chữ số thập phân, %g sẽ có nghĩa là viết một số có số chữ số tối thiểu bắt buộc

(tự động định dạng). Vì vậy, chúng ta có thể viết ví dụ trước như sau:

printf("Giá đóng cửa = %.3f\n", close);

printf("Thay đổi kể từ hôm qua = %.3f\n", close - ref(close, -1)); printf("Phần trăm thay đổi kể từ hôm qua = %.2f%%\n", roc(close, 1)); printf("MACD = %.4f, Đường tín hiệu = %.4f\n", macd(), signal());

Như chúng ta thấy, đoạn mã này ngắn hơn và rõ ràng hơn. Đối số đầu tiên của hàm printf là một chuỗi (nói một cách chính xác thì được gọi là chuỗi định dạng chứa văn bản và số giữ chỗ/chỉ định định dạng được đánh dấu bằng

%. Các đối số tiếp theo của hàm printf là các giá trị thực (số) mà chúng ta muốn viết (không cần phải chuyển đổi thành chuỗi nữa). Như bạn có thể đã lưu ý, nếu chúng ta chỉ muốn chỉ định dấu phần trăm, không phải chuỗi định dạng, chúng ta cần viết %% (hai dấu phần trăm).

Nhưng điều chúng ta thiếu ở đây là khả năng viết một cái gì đó nếu một số điều kiện được đáp ứng và viết một cái gì đó khác nếu không...

Đầu ra văn bản có điều kiện

AFL được trang bị một hàm rất hay có tên là WriteIF() có thể xuất ra các văn bản khác nhau tùy thuộc vào điều kiện. Chúng ta hãy xem tài liệu nói gì:

CÚ PHÁP writeif(EXPRESSION, "TRUE TEXT", "FALSE TEXT")

TRẢ VỀ CHUỖI

HÀM Hàm này chỉ có thể được sử dụng trong chú thích của Guru. Nếu EXPRESSION được đánh giá là "true", thì chuỗi TRUE TEXT sẽ được hiển thị trong chú thích. Nếu EXPRESSION được đánh giá là "false", thì chuỗi FALSE TEXT sẽ được hiển thị.

Vì vậy, chúng ta có thể dễ dàng xuất ra văn bản khác nhau tùy thuộc vào expession, ví dụ:

writeif(macd() > signal(), "MACD là tăng giá vì nó nằm trên đường tín hiệu của nó", "MACD là giảm giá vì nó nằm dưới đường tín hiệu của nó");

Bạn cũng có thể kết hợp một số lệnh gọi hàm WriteIf() để xử lý nhiều khả năng hơn:

"Tình hình thị trường hiện tại cho "+ name() + " là: ";

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

) + 0.1 * (close > ema(close , 30));

avgcond2 = -(c < ema(đóng, 200)) - 0,1 * (đóng < ema(đóng, 90)) - 0,1 * (đóng < ema(đóng, 30));

```
WriteIf( avgcond1 == 1.2,
"Very Bullish",
WriteIf( avgcond1 == 1.1,
"Bullish",
WriteIf( avgcond1 == 1.0,
```

"Hơi lạc quan", ""))) +

```
WriteIf( avgcond2 == -1.2,
"Very Bearish",
WriteIf( avgcond2 == -1.1,
"Bearish",
WriteIf( avgcond2 == -1.0,
```

"Biết điều", "")));

Công thức trên sẽ trả về văn bản "Tình hình thị trường hiện tại cho {mã chứng khoán của bạn tại đây} là: Rất tăng giá" nếu giá đóng cửa cao hơn mức trung bình 30 ngày và giá đóng cửa cao hơn mức trung bình 90 ngày và giá đóng cửa cao hơn mức trung bình 200 ngày. Trong các trường hợp khác, công thức sẽ cung cấp cho bạn xếp hạng Tăng giá, Tăng giá nhẹ, Giảm giá nhẹ, Giảm giá hoặc Rất giảm giá.

Để biết thêm ví dụ về bình luận AFL, vui lòng kiểm tra thư viện công thức AFL, đặc biệt là công thức bình luận MACD, công thức này minh họa tất cả các kỹ thuật được trình bày tại đây.

Bây giờ bạn đã sẵn sàng bắt đầu với bình luận của riêng mình... Chúc may mắn!

Sử dụng các nghiên cứu trong công thức AFL

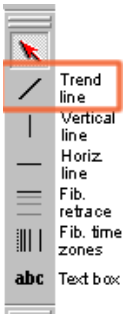
AmiBroker 3.52 giới thiệu khả năng tham chiếu các nghiên cứu vẽ tay từ công thức AFL. Tính năng này khá độc đáo trong số các phần mềm giao dịch và như bạn sẽ thấy, việc sử dụng tính năng này khá dễ dàng.

Tôi sẽ chỉ cho bạn một ví dụ về cách kiểm tra xem đường xu hướng có bị phá vỡ từ mã AFL hay không. Tất cả những gì chúng ta cần làm là ba bước đơn giản:

1. Vẽ đường xu hướng
2. Xác định ID nghiên cứu
3. Viết công thức kiểm tra điểm phá vỡ đường xu hướng

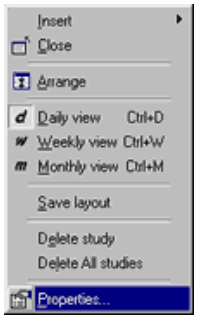
Vẽ đường xu hướng

Đường xu hướng là đường dốc được vẽ giữa hai điểm nổi bật trên biểu đồ.



Trong ví dụ này, chúng ta sẽ vẽ đường xu hướng tăng xác định xu hướng tăng. Loại đường xu hướng này thường được vẽ giữa hai (hoặc nhiều) đáy (điểm thấp) để minh họa cho mức hỗ trợ giá.

Chắc chắn bạn biết cách vẽ đường xu hướng trong AmiBroker - chỉ cần chọn công cụ "Đường xu hướng" từ thanh công cụ "Vẽ", tìm ít nhất hai đáy gần đây và chỉ cần vẽ đường.



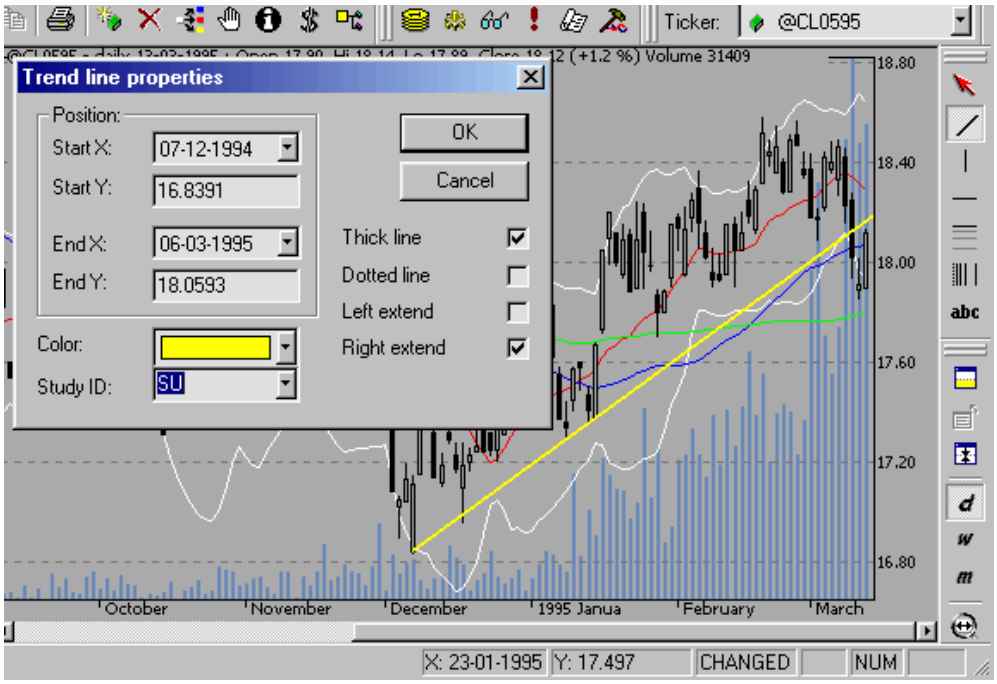
Xác định ID nghiên cứu

Như bạn có thể biết, bạn có thể sửa đổi các thuộc tính của từng đường được vẽ trong AmiBroker bằng cách nhấp chuột phải vào nghiên cứu và chọn "Thuộc tính" từ menu. Hộp thoại thuộc tính hiển thị cho phép bạn xác định điểm bắt đầu/kết thúc chính xác và chọn màu đường, kiểu và chế độ mở rộng sang trái và/hoặc sang phải.

Để phân tích sâu hơn, chúng tôi sẽ sử dụng đường xu hướng mở rộng bên phải (nhấp vào hộp kiểm thích hợp) để đảm bảo rằng đường xu hướng được tự động mở rộng khi dữ liệu mới được thêm vào.

Kể từ phiên bản 3.52, hộp thoại thuộc tính cũng cho phép xác định "ID nghiên cứu" (hộp màu kết hợp bên dưới). "ID nghiên cứu" là mã hai chữ cái của nghiên cứu có thể được gán cho bất kỳ nghiên cứu nào trong biểu đồ cho phép AmiBroker tham chiếu từ AFL. Các mã định danh được xác định trước là: "UP" - xu hướng tăng, "DN" - xu hướng giảm, "SU" - hỗ

trợ, "RE" - kháng cự, "ST" - dừng lỗ, tuy nhiên bạn có thể sử dụng BẤT KỲ mã định danh nào (không có giới hạn nào ngoại trừ việc AmiBroker chỉ chấp nhận mã 2 chữ cái). Theo cách này, nếu bạn vẽ các đường hỗ trợ bằng nhiều ký hiệu và cung cấp cho tất cả chúng mã định danh "SU" thì bạn sẽ có thể tham chiếu đường hỗ trợ từ mã AFL.



Vì vậy, chúng tôi sẽ gán ID nghiên cứu "SU" cho đường xu hướng hỗ trợ tăng mà chúng tôi vừa vẽ.

Viết công thức kiểm tra đường xu hướng bị phá vỡ

Trong ví dụ này, chúng ta sẽ phát hiện xem giá đóng cửa có GIẢM DƯỚI đường xu hướng hỗ trợ hay không. Thực ra, điều này rất đơn giản:

```
Sell = Cross( Study( "SU", GetChartID() ), Close );
```

Lưu ý rằng hàm study() chấp nhận hai đối số: đối số đầu tiên là mã hai chữ cái StudyID tương ứng với mã được cung cấp trong hộp thoại properites; đối số thứ hai là ID biểu đồ - đối số này phải được lấy thông qua hàm GetChartID() (sau đó tham chiếu đến chỉ báo hiện tại) hoặc đọc từ hộp thoại Parameter, Axes & Grid: Miscellaneous: Chart ID.

Kiểm tra ngược các ý tưởng giao dịch của bạn

Giới thiệu

Một trong những điều hữu ích nhất mà bạn có thể làm trong cửa sổ phân tích là kiểm tra ngược chiến lược giao dịch của mình trên dữ liệu lịch sử. Điều này có thể cung cấp cho bạn cái nhìn sâu sắc có giá trị về điểm mạnh và điểm yếu của hệ thống trước khi đầu tư tiền thật. Tính năng AmiBroker duy nhất này có thể giúp bạn tiết kiệm rất nhiều tiền.

Viết các quy tắc giao dịch của bạn

Trước tiên, bạn cần có các quy tắc khách quan (hoặc cơ học) để vào và thoát khỏi thị trường. Bước này là cơ sở cho chiến lược của bạn và bạn cần tự mình suy nghĩ về nó vì hệ thống phải phù hợp với khả năng chịu rủi ro, quy mô danh mục đầu tư, kỹ thuật quản lý tiền và nhiều yếu tố cá nhân khác của bạn.

Khi đã có các quy tắc giao dịch của riêng mình, bạn nên viết chúng thành các quy tắc mua và bán trong Ngôn ngữ công thức AmiBroker (cộng với lệnh bán khống và lệnh mua lại nếu bạn muốn thử nghiệm cả lệnh bán khống).

Trong chương này, chúng ta sẽ xem xét hệ thống giao cắt trung bình động cơ bản nhất. Hệ thống sẽ mua cổ phiếu/hợp đồng khi giá đóng cửa tăng trên đường trung bình động hàm mũ 45 ngày và sẽ bán cổ phiếu/hợp đồng khi giá đóng cửa giảm xuống dưới đường trung bình động hàm mũ 45 ngày.

Đường trung bình động hàm mũ có thể được tính toán trong AFL bằng hàm EMA tích hợp sẵn. Tất cả những gì bạn cần làm là chỉ định mảng đầu vào và chu kỳ trung bình, do đó, đường trung bình động hàm mũ 45 ngày của giá đóng cửa có thể được lấy bằng câu lệnh sau:

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

Mã định danh đóng là mảng tích hợp chứa giá đóng của ký hiệu hiện đang được phân tích.

Để kiểm tra xem giá đóng có vượt qua đường trung bình động hàm mũ hay không, chúng ta sẽ sử dụng hàm chéo tích hợp:

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

Câu lệnh trên định nghĩa một quy tắc giao dịch mua. Nó trả về "1" hoặc "true" khi giá đóng vượt qua ema(close, 45). Sau đó, chúng ta có thể viết quy tắc bán sẽ trả về "1" khi tình huống ngược lại xảy ra - giá đóng vượt qua dưới ema(close, 45):

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

Xin lưu ý rằng chúng ta đang sử dụng cùng một hàm chéo nhưng thứ tự đối số ngược lại. Vì vậy, công thức hoàn chỉnh cho các giao dịch dài hạn sẽ như sau:

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

LƯU Ý: Để tạo công thức mới, vui lòng mở Trình soạn thảo công thức bằng menu Phân tích->Trình soạn thảo công thức, nhập công thức và chọn menu Công cụ->Gửi đến Phân tích trong trình soạn thảo công thức

Kiểm tra ngược

Để kiểm tra ngược hệ thống của bạn, chỉ cần nhấp vào nút Kiểm tra ngược trong cửa sổ Phân tích tự động. Đảm bảo bạn đã nhập công thức có chứa ít nhất các quy tắc giao dịch mua và bán (như được hiển thị ở trên). Khi công thức chính xác, AmiBroker sẽ bắt đầu phân tích các ký hiệu của bạn theo các quy tắc giao dịch của bạn và tạo danh sách các giao dịch mô phỏng. Toàn bộ quá trình diễn ra rất nhanh - bạn có thể kiểm tra ngược hàng nghìn ký hiệu chỉ trong vài phút. Cửa sổ tiến trình sẽ hiển thị cho bạn thời gian hoàn thành ước tính. Nếu bạn muốn dừng quá trình, bạn chỉ cần nhấp vào nút Hủy trong cửa sổ tiến trình.

Phân tích kết quả

Khi quá trình hoàn tất, danh sách các giao dịch mô phỏng sẽ hiển thị ở phần dưới cùng của cửa sổ Phân tích tự động. (khung Kết quả). Bạn có thể kiểm tra thời điểm tín hiệu mua và bán xảy ra chỉ bằng cách nhấp đúp vào giao dịch trong khung Kết quả. Thao tác này sẽ cung cấp cho bạn các tín hiệu thô hoặc chưa lọc cho mọi thanh khi các điều kiện mua và bán được đáp ứng. Nếu bạn chỉ muốn xem các mũi tên giao dịch đơn lẻ (mở và đóng giao dịch hiện đang được chọn), bạn nên nhấp đúp vào dòng trong khi giữ phím SHIFT. Ngoài ra, bạn có thể chọn loại hiển thị bằng cách chọn mục thích hợp từ menu ngữ cảnh xuất hiện khi bạn nhấp vào ngăn kết quả bằng nút chuột phải.

Ngoài danh sách kết quả, bạn có thể nhận được số liệu thống kê rất chi tiết về hiệu suất của hệ thống bằng cách nhấp vào nút Báo cáo. Để tìm hiểu thêm về số liệu thống kê báo cáo, vui lòng xem mô tả cửa sổ báo cáo.

Thay đổi cài đặt kiểm tra ngược của bạn

Công cụ kiểm tra ngược trong AmiBroker sử dụng một số giá trị được xác định trước để thực hiện nhiệm vụ của nó bao gồm quy mô danh mục đầu tư, chu kỳ (hàng ngày/hàng tuần/hàng tháng), số tiền hoa hồng, lãi suất, mục tiêu dừng lỗ và lợi nhuận tối đa, loại giao dịch, trường giá, v.v. Người dùng có thể thay đổi tất cả các cài đặt này bằng cửa sổ cài đặt. Sau khi thay đổi cài đặt, hãy nhớ chạy lại kiểm tra ngược của bạn nếu bạn muốn kết quả đồng bộ với cài đặt.

Ví dụ, để kiểm tra ngược trên các thanh hàng tuần thay vì hàng ngày, chỉ cần nhấp vào nút Cài đặt, chọn Hàng tuần từ hộp kết hợp

Chu kỳ và nhấp vào OK, sau đó chạy phân tích của bạn bằng cách nhấp vào Kiểm tra ngược.

Tên biến được đặt trước

Bảng sau đây hiển thị tên của các biến được đặt trước do Automatic Analyser sử dụng. Ý nghĩa và ví dụ về cách sử dụng chúng được đưa ra sau trong chương này.

Biến	Sử dụng	Áp dụng cho
mua	xác định quy tắc giao dịch "mua" (vào vị thế mua)	Phân tích tự động, Bình luận
bán	xác định quy tắc giao dịch "bán" (đóng vị thế mua)	Phân tích tự động, Bình luận
ngắn	xác định quy tắc giao dịch "ngắn" (vào vị thế bán - bán khống)	Phân tích tự động
bao phủ	xác định quy tắc giao dịch "bao phủ" (đóng vị thế bán - mua để bao phủ)	Phân tích tự động
buyprice	xác định mảng giá mua (mảng này được điền các giá trị mặc định theo cài đặt của Trình phân tích tự động)	Phân tích tự động
sellprice	xác định mảng giá bán (mảng này được điền các giá trị mặc định theo cài đặt của Trình phân tích tự động)	Phân tích tự động
shortprice	xác định mảng giá bán khống (mảng này được điền các giá trị mặc định theo cài đặt của Trình phân tích tự động)	Phân tích tự động
coverprice	xác định mảng giá mua để bao phủ (mảng này được điền các giá trị mặc định theo cài đặt của Trình phân tích tự động)	Phân tích tự động
loại trừ	Nếu được xác định, giá trị true (hoặc 1) giá trị của biến này loại trừ ký hiệu hiện tại khỏi quét/khám phá/kiểm tra ngược. Chúng cũng không được xem xét trong các phép tính mua và giữ. Hữu ích khi bạn muốn thu hẹp phân tích của mình vào một tập hợp ký hiệu nhất định.	Phân tích tự động
roundlotsize	xác định kích thước lô tròn được backtester sử dụng (xem giải thích bên dưới)	Phân tích tự động (mới trong 4.10)

ticksize	xác định kích thước tick được sử dụng để căn chỉnh giá do các lệnh dừng tích hợp tạo ra (xem giải thích bên dưới) (lưu ý: không ảnh hưởng đến giá vào/ra được chỉ định bởi buyprice/sellprice/shortprice/coverprice)	Phân tích tự động (mới trong 4.10)
pointvalue	cho phép đọc và sửa đổi giá trị điểm hợp đồng tương lai (xem backtesting futures) CAVEAT: biến AFL này theo mặc định được đặt thành 1 (một) bất kể nội dung của cửa sổ Thông tin TRỪ KHI bạn BẬT chế độ tương lai (SetOption("FuturesMode", True))	Phân tích tự động (mới trong 4.10)
margindeposit	cho phép đọc và sửa đổi biên độ hợp đồng tương lai (xem backtesting futures)	Phân tích tự động (mới trong 4.10)
positionsize	Cho phép kiểm soát số tiền đô la hoặc tỷ lệ phần trăm danh mục đầu tư được đầu tư vào giao dịch (xem giải thích bên dưới)	Phân tích tự động (mới trong 3.9)

Các khái niệm nâng cao

Cho đến bây giờ, chúng ta đã thảo luận về cách sử dụng khá đơn giản của trình kiểm tra ngược. Tuy nhiên, AmiBroker hỗ trợ nhiều phương pháp và khái niệm phức tạp hơn nhiều sẽ được thảo luận sau trong chương này. Xin lưu ý rằng người dùng mới bắt đầu trước tiên nên chơi một chút với các chủ đề dễ hơn được mô tả ở trên trước khi tiếp tục.

Vì vậy, khi bạn đã sẵn sàng, vui lòng xem các tính năng mới được giới thiệu sau đây của trình kiểm tra ngược:

a) Máy chủ tập lệnh AFL dành cho người viết công thức nâng cao

b) hỗ trợ nâng cao cho các giao dịch bán khống

c) cách kiểm soát giá thực hiện lệnh từ tập lệnh

d) nhiều loại điểm dừng khác nhau trong trình kiểm tra ngược

e) định cỡ vị thế

f) kích thước lô tròn và kích thước tick

g) tài khoản ký quỹ

h) kiểm tra ngược hợp đồng tương lai

Máy chủ tập lệnh AFL là một chủ đề nâng cao được đề cập trong một tài liệu riêng có tại đây và tôi sẽ không thảo luận về chủ đề này trong tài liệu này. Các tính năng còn lại dễ hiểu hơn nhiều.

Hỗ trợ giao dịch bán khống

Trong các phiên bản trước của AmiBroker, nếu bạn muốn kiểm tra ngược hệ thống bằng cả giao dịch mua và bán khống, bạn chỉ có thể mô phỏng chiến lược dừng và đảo ngược. Khi vị thế mua được đóng, một vị thế bán khống mới được mở ngay lập tức. Đó là vì các biến dự trữ mua và bán được sử dụng cho cả hai loại giao dịch.

Bây giờ (với phiên bản 3.59 trở lên), có các biến dự trữ riêng biệt để mở và đóng giao dịch mua và bán khống:

mua - "đúng" hoặc 1 giá trị mở giao dịch mua bán - "đúng" hoặc 1 giá trị đóng giao dịch mua bán - "đúng" hoặc 1 giá trị mở giao dịch bán khống che - "đúng" hoặc 1 giá trị đóng giao dịch bán khống

Vì vậy, để kiểm tra ngược giao dịch bán khống, bạn cần chỉ định các biến bán khống và che.

Nếu bạn sử dụng hệ thống dừng và đảo ngược (luôn trên thị trường), chỉ cần chỉ định bán cho bán khống và mua cho che

bán khống = bán; che = mua;

Điều này mô phỏng cách hoạt động của các phiên bản trước 3.59.

Nhưng hiện tại AmiBroker cho phép bạn có các quy tắc giao dịch riêng biệt để mua và bán như thể hiện trong ví dụ đơn giản này:

```
// quy tắc vào và thoát lệnh mua:
buy = cross( cci(), 100 ); sell = cross( 100, cci() );

// quy tắc vào và thoát lệnh bán:
short = cross( -100, cci() ); cover = cross( cci(), -100 );
```

Lưu ý rằng trong ví dụ này, nếu CCI nằm trong khoảng từ -100 đến 100, bạn sẽ không vào thị trường.

Kiểm soát giá giao dịch

AmiBroker hiện cung cấp 4 biến dự trữ mới để chỉ định giá thực hiện các lệnh mua, bán, bán khống và che phủ. Các mảng này có tên sau: buyprice, sellprice, shortprice và coverprice.

Ứng dụng chính của các biến này là kiểm soát giá giao dịch:

BuyPrice = IIF(dayofweek() == 1, HIGH, CLOSE);

// vào thứ hai mua ở mức cao, nếu không mua khi đóng

Vì vậy, bạn có thể viết lệnh sau để mô phỏng lệnh dừng thực tế:

BuyStop = ... công thức cho mức dừng mua; SellStop = ... công thức cho mức dừng bán;

// nếu bất kỳ lúc nào trong ngày giá tăng trên mức dừng mua (cao> dừng mua)

// lệnh mua diễn ra (ở mức dừng mua hoặc thấp tùy theo mức nào cao hơn) Buy = Cross(Cao, BuyStop);

// nếu bất kỳ lúc nào trong ngày giá giảm xuống dưới mức giá bán (thấp

< dừng bán)

// lệnh bán diễn ra (ở mức dừng bán hoặc cao tùy theo mức nào thấp hơn)

Sell = Cross(Giá bán, Dừng bán);

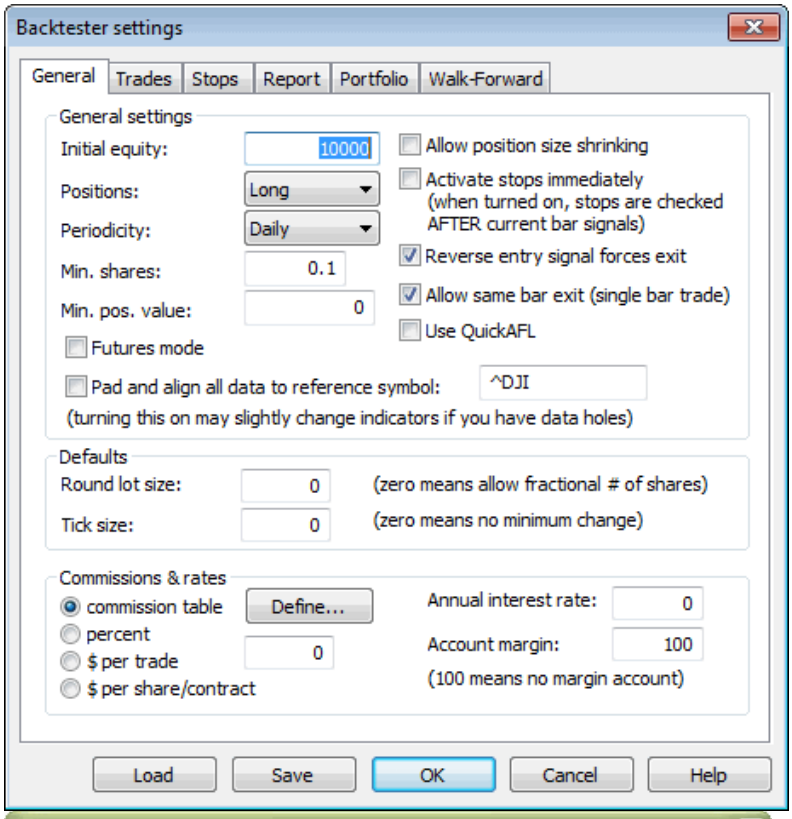
BuyPrice = max(Dừng mua, Thấp); // đảm bảo giá mua không thấp hơn Thấp

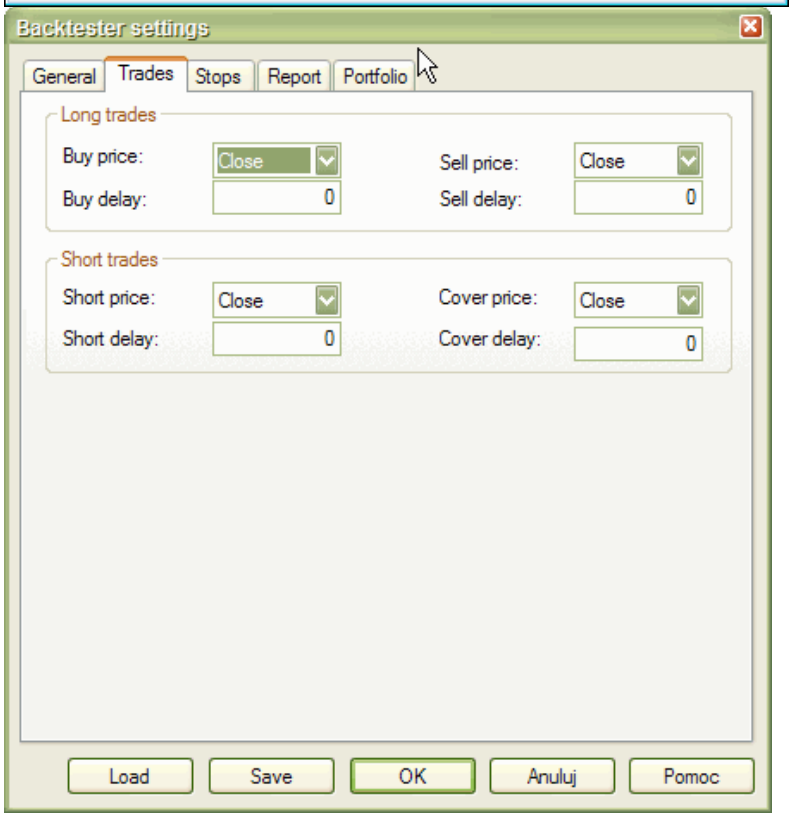
SellPrice = min(Dừng bán, Cao); // đảm bảo giá bán không lớn hơn Cao

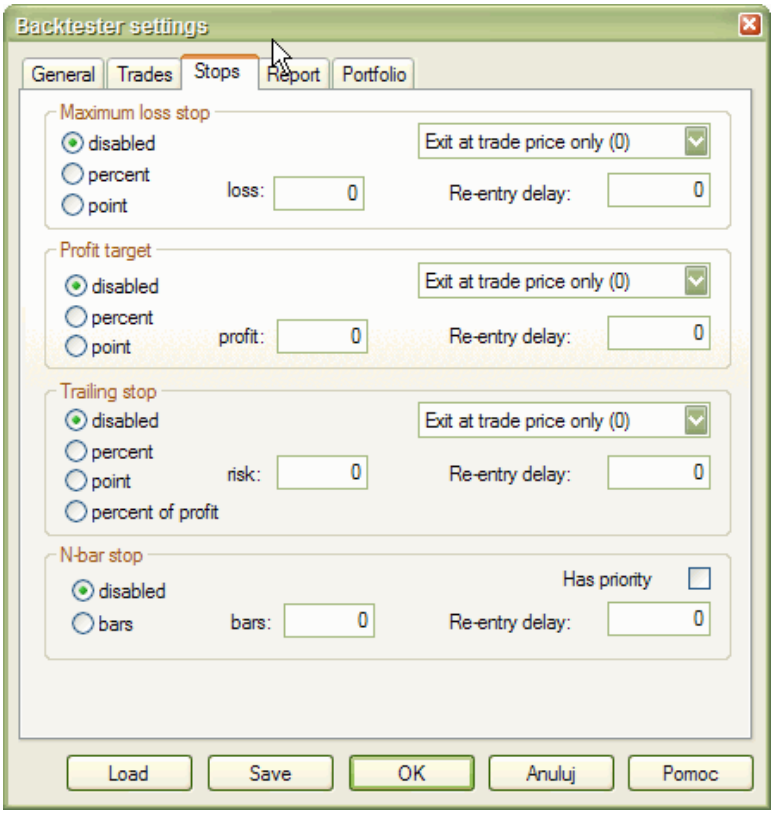
Xin lưu ý rằng AmiBroker cài đặt trước các biến mảng buyprice, sellprice, shortprice và coverprice với các giá trị được xác định trong cửa sổ cài đặt kiểm tra hệ thống (hiển thị bên dưới), do đó bạn có thể nhưng không cần phải xác định chúng trong công thức của mình. Nếu bạn không xác định chúng, AmiBroker hoạt động như trong các phiên bản cũ.

Trong quá trình kiểm tra ngược, AmiBroker sẽ kiểm tra xem các giá trị bạn gán cho buyprice, sellprice, shortprice, coverprice có phù hợp với phạm vi cao-thấp của thanh đã cho

hay không. Nếu không, AmiBroker sẽ điều chỉnh thành giá cao (nếu giá trị mảng giá cao hơn giá cao) hoặc thành giá thấp (nếu giá trị mảng giá thấp hơn giá thấp)







Mục tiêu lợi nhuận dừng

Như bạn có thể thấy trong hình trên, các thiết lập mới cho mục tiêu lợi nhuận dừng có sẵn trong cửa sổ thiết lập kiểm tra hệ thống. Mục tiêu lợi nhuận dừng được thực hiện khi giá cao trong một ngày nhất định vượt quá mức dừng có thể được đưa ra dưới dạng phần trăm hoặc điểm tăng từ giá mua. Theo mặc định, các lệnh dừng được thực hiện ở mức giá mà bạn xác định là mảng giá bán (đối với giao dịch mua) hoặc mảng giá bảo hiểm (đối với giao dịch bán khống). Hành vi này có thể được thay đổi bằng cách sử dụng tính năng "Thoát khi dừng".

Tính năng "Thoát khi dừng"

Nếu bạn đánh dấu vào ô "Thoát khi dừng" trong cài đặt, các lệnh dừng sẽ được thực hiện ở mức dừng chính xác, tức là nếu bạn xác định mục tiêu lợi nhuận dừng ở mức +10% lệnh dừng của bạn và giá mua là 50 lệnh dừng sẽ được thực hiện ở mức 55 ngay cả khi mảng giá bán của bạn chứa giá trị khác (ví dụ giá đóng cửa là 56).

Lệnh dừng lỗ tối đa hoạt động theo cách tương tự - chúng được thực hiện khi giá thấp trong một ngày nhất định giảm xuống dưới mức dừng có thể được đưa ra dưới dạng phần trăm hoặc điểm tăng từ giá mua

Dừng theo sau

Loại dừng này được sử dụng để bảo vệ lợi nhuận vì nó theo dõi giao dịch của bạn, do đó mỗi khi giá trị vị thế đạt đến mức cao mới, lệnh dừng theo sau được đặt ở mức cao hơn. Khi lợi nhuận giảm xuống dưới mức dừng theo sau, vị thế sẽ đóng lại. Cơ chế này được minh họa trong hình bên dưới (hiển thị lệnh dừng theo sau 10%)



Dừng theo sau, cũng như hai loại dừng khác có thể được bật từ giao diện người dùng (cửa sổ Cài đặt phân tích tự động) hoặc từ cấp công thức - sử dụng hàm ApplyStop:

Để tái tạo ví dụ trên, bạn sẽ cần thêm mã sau vào công thức phân tích tự động của mình:

ApplyStop(2, 1, 10, 1); // 10% trailing stop, chế độ phần trăm, thoát khi dừng BẬT

hoặc bạn có thể viết nó bằng các hằng số được xác định trước có tính mô tả hơn
ApplyStop(stopTypeTrail, stopModePercent, 10, True);

Trailing stop cũng có thể được định nghĩa theo điểm (đô la) và phần trăm lợi nhuận (rủi ro). Trong trường hợp sau, tham số số tiền xác định phần trăm lợi nhuận có thể bị mất mà không kích hoạt lệnh dừng. Vì vậy, lệnh dừng 20% phần trăm lợi nhuận (rủi ro) sẽ thoát khỏi giao dịch của bạn có lợi nhuận tối đa là 100 đô la khi lợi nhuận giảm xuống dưới 80 đô la.

Lệnh dừng động

Giờ đây, hàm ApplyStop() cho phép thay đổi mức dừng từ giao dịch này sang giao dịch khác. Điều này cho phép bạn triển khai lệnh dừng dựa trên biến động rất dễ dàng.

Ví dụ, để áp dụng lệnh dừng lỗ tối đa sẽ điều chỉnh mức lỗ tối đa có thể chấp nhận được dựa trên phạm vi thực trung bình 10 ngày, bạn sẽ cần viết:

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

hoặc bạn có thể viết nó bằng các hằng số được xác định trước có tính mô tả hơn

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

Hàm trên sẽ đặt lệnh dừng 2 lần ATR 10 ngày dưới giá vào lệnh.

Khi ATR thay đổi từ giao dịch này sang giao dịch khác - điều này sẽ dẫn đến mức dừng động, dựa trên biến động. Xin lưu ý rằng tham số thứ 3 của hàm ApplyStop (số tiền) được lấy mẫu tại mục nhập giao dịch và được giữ trong suốt giao dịch. Vì vậy, trong ví dụ trên, nó sử dụng giá trị ATR(10) từ ngày vào lệnh. Những thay đổi tiếp theo của ATR không ảnh hưởng đến mức dừng.

Xem tài liệu đầy đủ về hàm APPLYSTOP để biết thêm chi tiết.

Mã hóa các loại dừng tùy chỉnh của riêng bạn

Hàm ApplyStop có mục đích bao gồm hầu hết các loại dừng "phổ biến". Tuy nhiên, bạn có thể mã hóa loại dừng và thoát của riêng mình bằng cách sử dụng mã lặp. Ví dụ, sau đây triển khai lại lệnh dừng mục tiêu lợi nhuận và cho biết cách tham chiếu đến giá vào lệnh giao dịch trong công thức của bạn:

```
/* một ví dụ triển khai cấp thấp của lệnh dừng mục tiêu lợi nhuận trong AFL: */

Mua = 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 )

{
```

Bán[i] = 1;

Bán[i] = 1.1 * priceatbuy;

priceatbuy = 0;

}

else

Bán[i] = 0;

}

Kích thước vị thế

Đây là tính năng mới trong phiên bản 3.9. Kích thước vị thế trong backtester được triển khai bằng biến dự trữ mới

PositionSize = <mảng kích thước>

Bây giờ bạn có thể kiểm soát số tiền đô la hoặc tỷ lệ phần trăm danh mục đầu tư được đầu tư vào giao dịch

· số dương xác định số tiền (đô la) được đầu tư vào giao dịch, ví dụ: PositionSize = 1000; // đầu tư 1000 đô la vào mọi giao dịch

· số âm -100..-1 xác định tỷ lệ phần trăm:

-100 cho 100% kích thước danh mục đầu tư hiện tại,

-33 cho 33% vốn chủ sở hữu khả dụng, ví dụ:

PositionSize = -50; /* luôn chỉ đầu tư một nửa vốn chủ sở hữu hiện tại */

· ví dụ về kích thước động:

PositionSize = - 100 + RSI();

vì RSI thay đổi từ 0..100 nên vị thế sẽ phụ thuộc vào giá trị RSI -> giá trị RSI thấp sẽ dẫn đến tỷ lệ phần trăm đầu tư cao hơn

Nếu đầu tư ít hơn 100% tiền mặt khả dụng thì số tiền còn lại sẽ được hưởng lãi suất theo định nghĩa trong cài đặt.

Ngoài ra còn có một hộp kiểm mới trong cửa sổ cài đặt AA: "Cho phép thu hẹp quy mô vị thế" - hộp kiểm này kiểm soát cách backtester xử lý tình huống khi quy mô vị thế được yêu cầu (thông qua biến PositionSize) vượt quá tiền mặt khả dụng: khi cờ này được chọn, vị thế sẽ được nhập với quy mô thu hẹp thành tiền mặt khả dụng nếu không được chọn thì vị thế sẽ không được nhập.

Để xem quy mô vị thế thực tế, vui lòng sử dụng chế độ báo cáo mới trong cửa sổ cài đặt AA: "Danh sách giao dịch với giá và quy mô vị thế"

Cuối cùng, đây là một ví dụ về kỹ thuật định cỡ vị thế dựa trên ATR của Tharp được mã hóa trong AFL:

Mua = <công thức mua của bạn tại đây> Bán = 0; // chỉ bán bằng lệnh dừng

TrailStopAmount = 2 * ATR(20);

Vốn = 100000; /* QUAN TRỌNG: Đặt nó trong Cài đặt: Vốn chủ sở hữu ban đầu */

Rủi ro = 0,01*Vốn chủ sở hữu;

PositionSize = (Rủi ro/TrailStopAmount)*BuyPrice; ApplyStop(2, 2, TrailStopAmount, 1);

Kỹ thuật này có thể được tóm tắt như sau:

Tổng vốn chủ sở hữu trên mỗi ký hiệu là 100.000 đô la, chúng tôi đặt mức rủi ro ở mức 1% tổng vốn chủ sở hữu. Mức rủi ro được định nghĩa như sau: nếu lệnh dừng lỗ trên một cổ phiếu 50 đô la ở mức, chẳng hạn, 45 đô la (giá trị của hai ATR so với vị thế), khoản lỗ 5 đô la được chia thành rủi ro 1.000 đô la để có 200 cổ phiếu để mua. Vì vậy, rủi ro lỗ là 1.000 đô la nhưng rủi ro phân bổ là 200 cổ phiếu x 50 đô la/cổ phiếu hoặc 10.000 đô la. Vì vậy, chúng tôi

phân bổ 10% vốn chủ sở hữu cho giao dịch mua nhưng chỉ mạo hiểm 1000 đô la. (Trích đoạn đã chỉnh sửa từ danh sách gửi thư AmiBroker)

Kích thước lô tròn và kích thước tick

Kích thước lô tròn

Nhiều công cụ khác nhau được giao dịch với nhiều "đơn vị giao dịch" hoặc "khối" khác nhau. Ví dụ, bạn có thể mua một số lượng nhỏ đơn vị quỹ tương hỗ, nhưng bạn không thể mua một số lượng nhỏ cổ phiếu. Đôi khi bạn phải mua theo lô 10 hoặc 100. AmiBroker hiện cho phép bạn chỉ định kích thước khối ở cấp độ toàn cầu và

mỗi ký hiệu.

Bạn có thể xác định kích thước lô tròn theo ký hiệu trong trang Biểu tượng->Thông tin (hình 3). Giá trị bằng không có nghĩa là ký hiệu không có kích thước lô tròn đặc biệt và sẽ sử dụng "Kích thước lô tròn mặc định" (cài đặt toàn cầu) từ trang cài đặt Phân tích tự động (hình 1). Nếu kích thước mặc định cũng được đặt thành không, điều đó có nghĩa là số lượng nhỏ cổ phiếu/hợp đồng được phép.

Bạn cũng có thể kiểm soát kích thước lô tròn trực tiếp từ công thức AFL của mình bằng cách sử dụng biến được đặt trước RoundLotSize, ví dụ:

RoundLotSize = 10;

Kích thước tích tắc

Thiết lập này kiểm soát mức giá tối thiểu di chuyển của ký hiệu đã cho. Bạn có thể xác định nó ở cấp độ toàn cầu và theo từng ký hiệu. Cũng như kích thước lô tròn, bạn có thể xác định kích thước tích tắc theo từng ký hiệu trong trang Ký hiệu->Thông tin (hình 3). Giá trị bằng không hướng dẫn AmiBroker sử dụng "kích thước tích tắc mặc định" được xác định trong trang Cài đặt (hình 1) của cửa sổ Phân tích tự động. Nếu kích thước tích tắc mặc định cũng được đặt thành bằng không, điều đó có nghĩa là không có mức giá tối thiểu di chuyển.

Bạn cũng có thể đặt và truy xuất kích thước tích tắc từ công thức AFL bằng cách sử dụng biến được đặt trước TickSize, ví dụ:

TickSize = 0,01;

Lưu ý rằng thiết lập kích thước tích tắc CHỈ ảnh hưởng đến các giao dịch thoát khỏi lệnh dừng tích hợp và/hoặc ApplyStop(). Trình kiểm tra ngược giả định rằng dữ liệu giá tuân theo các yêu cầu về kích thước tích tắc và nó không thay đổi các mảng giá do người dùng cung cấp.

Vì vậy, việc chỉ định kích thước tick chỉ có ý nghĩa nếu bạn đang sử dụng các điểm dừng tích hợp để các điểm thoát được tạo ra ở mức giá "được phép" thay vì mức giá được tính toán. Ví dụ ở Nhật Bản - bạn không thể có các phần phân số của yên nên bạn nên xác định kích thước tick toàn cầu là 1, do đó các điểm dừng tích hợp thoát giao dịch ở mức số nguyên.

Tài khoản ký quỹ

Cài đặt ký quỹ tài khoản xác định yêu cầu ký quỹ theo phần trăm cho toàn bộ tài khoản. Giá trị mặc định của Ký quỹ tài khoản là 100. Điều này có nghĩa là bạn phải cung cấp 100% tiền để tham gia giao dịch và đây là cách backtester hoạt động trong các phiên bản trước. Nhưng giờ đây bạn có thể mô phỏng một tài khoản ký quỹ. Khi bạn mua ký quỹ, bạn chỉ cần vay tiền từ công ty môi giới của mình để mua cổ phiếu. Với các quy định hiện hành, bạn có thể đặt 50% giá mua cổ phiếu mà bạn muốn mua và vay một nửa còn lại từ công ty môi giới của mình. Để mô phỏng điều này, chỉ cần nhập 50 vào trường Ký quỹ tài khoản (xem hình 1). Nếu vốn chủ sở hữu ban đầu của bạn được đặt thành 10000, sức mua của bạn sẽ là 20000 và bạn sẽ có thể tham gia các vị thế lớn hơn. Xin lưu ý rằng cài đặt này đặt ký quỹ cho toàn bộ tài khoản và KHÔNG liên quan đến giao dịch tương lai. Nói cách khác, bạn có thể giao dịch cổ phiếu trên tài khoản ký quỹ.

Cài đặt bổ sung

· Hộp kiểm "Tín hiệu nhập ngược buộc phải thoát" trong cài đặt Backtester.

Khi BẬT (cài đặt mặc định) - backtester hoạt động như trong các phiên bản trước và đóng vị trí đã mở nếu gặp tín hiệu vào lệnh mới theo hướng ngược lại. Nếu công tắc này TẮT - ngay cả khi tín hiệu ngược xảy ra, backtester vẫn duy trì giao dịch hiện đang mở và không đóng vị trí cho đến khi tín hiệu thoát lệnh thường xuyên (bán hoặc bảo hiểm) được tạo ra.

Nói cách khác, khi công tắc này TẮT, backtester sẽ bỏ qua các tín hiệu Bán trong các giao dịch dài hạn và bỏ qua các tín hiệu Mua trong các giao dịch ngắn hạn.

· Tùy chọn "Cho phép thoát lệnh cùng một thanh (giao dịch một thanh)" trong Cài đặt

Khi BẬT (cài đặt mặc định) - cho phép vào lệnh và thoát lệnh tại cùng một thanh (như trong các phiên bản trước)

nếu TẮT - lệnh thoát lệnh có thể xảy ra chỉ bắt đầu từ thanh tiếp theo (điều này áp dụng cho các tín hiệu thường xuyên, có một cài đặt riêng cho lệnh thoát lệnh do ApplyStop tạo ra). Chuyển sang TẮT cho phép tái tạo hành vi của backtester MS không thể xử lý lệnh thoát lệnh trong cùng ngày.

· "Kích hoạt lệnh dừng ngay lập tức"

Cài đặt này giải quyết vấn đề kiểm tra các hệ thống vào lệnh giao dịch khi thị trường mở. Trong các phiên bản trước 4.09, backtester cho rằng bạn đang tham gia giao dịch khi thị trường đóng cửa nên các lệnh dừng tích hợp được kích hoạt từ ngày hôm sau. Vấn đề là khi bạn thực sự định nghĩa giá mở là giá vào lệnh giao dịch - thì các biến động giá trong ngày không kích hoạt lệnh dừng. Có một số giải pháp thay thế đã được công bố dựa trên

mã AFL nhưng giờ bạn không cần sử dụng chúng nữa. Đơn giản là nếu bạn giao dịch khi mở cửa, bạn nên đánh dấu "Kích hoạt lệnh dừng ngay lập tức" (hình 1).

Bạn có thể hỏi tại sao không chỉ cần kiểm tra mảng giá mua hoặc giá bán khống nếu nó bằng giá mở cửa. Thật không may, điều này sẽ không hiệu quả. Tại sao? Đơn giản là vì có những ngày doji khi giá mở cửa bằng giá đóng cửa và sau đó backtester sẽ không bao giờ biết được giao dịch được thực hiện khi thị trường mở cửa hay đóng cửa. Vì vậy, chúng ta thực sự cần một thiết lập riêng.

· "Sử dụng QuickAFL"

QuickAFL(tm) là một tính năng cho phép tính toán AFL nhanh hơn trong một số điều kiện nhất định. Ban đầu (từ năm 2003), tính năng này chỉ khả dụng cho các chỉ báo, kể từ phiên bản 5.14 trở lên, tính năng này cũng khả dụng trong Automatic Analysis.

Ban đầu, ý tưởng là cho phép vẽ lại biểu đồ nhanh hơn thông qua việc tính toán công thức AFL chỉ cho phần hiển thị trên biểu đồ. Tương tự như vậy, cửa sổ phân tích tự động có thể sử dụng tập hợp con các báo giá khả dụng để tính AFL, nếu tham số "phạm vi" được chọn nhỏ hơn "Tất cả các báo giá".

Giải thích chi tiết về cách QuickAFL hoạt động và cách kiểm soát nó được cung cấp trong bài viết Cơ sở kiến thức này: http://www.amibroker.com/kb/2008/07/03/quickafl/

Lưu ý rằng tùy chọn này không chỉ hoạt động trong backtester mà còn trong tối ưu hóa, khám phá và quét.

Xem thêm:

Bài viết backtesting cấp danh mục đầu tư.

Bài viết về hệ thống backtesting cho hợp đồng tương lai.

Mô tả hàm APPLYSTOP

Sử dụng phần trình soạn thảo AFL của hướng dẫn.

Hướng dẫn nội bộ về backtester (bản tin 1/2002)

Backtester cấp danh mục đầu tư

QUAN TRỌNG: Vui lòng đọc Hướng dẫn đầu tiên: Backtester ý tưởng giao dịch của bạn

Backtester mới hoạt động ở CẤP DANH MỤC ĐẦU TƯ, nghĩa là có một danh mục đầu tư vốn chủ sở hữu và quy mô vị thế tham chiếu đến vốn chủ sở hữu danh mục đầu tư. Vốn chủ sở hữu danh mục đầu tư bằng tiền mặt khả dụng cộng với tổng của tất cả các vị thế mở đồng thời tại thời điểm nhất định.

Công cụ kiểm tra ngược danh mục đầu tư của AmiBroker cho phép bạn kết hợp các tín hiệu giao dịch và chiến lược định cỡ giao dịch thành các mô phỏng mô phỏng chính xác cách bạn giao dịch theo thời gian thực. Một tính năng cốt lõi là khả năng thực hiện quản lý tiền năng động và kiểm soát rủi ro ở cấp danh mục đầu tư. Quy mô vị thế được xác định với đầy đủ kiến thức về những gì đang diễn ra ở cấp danh mục đầu tư tại thời điểm quyết định định cỡ được đưa ra. Giống như bạn làm trong thực tế.

CÁCH THIẾT LẬP?

Chỉ cần thực hiện hai việc để thực hiện kiểm tra ngược danh mục đầu tư

1. Trước tiên, bạn cần có công thức tạo ra tín hiệu mua / bán / bán khống / che phủ như được mô tả trong bài viết "Kiểm tra ngược ý tưởng giao dịch của bạn"

2. Bạn nên xác định số lượng giao dịch đồng thời mà bạn muốn kiểm tra và thuật toán định cỡ vị thế nào

THIẾT LẬP SỐ LƯỢNG GIAO DỊCH MỞ ĐỒNG THỜI TỐI ĐA

Có hai cách để thiết lập số lượng giao dịch mở đồng thời tối đa:

1. Vào hộp thoại Cài đặt, chuyển sang tab Danh mục đầu tư và nhập số vào trường Vị thế mở tối đa

2. Xác định giá trị tối đa trong chính công thức (điều này ghi đè bất kỳ thiết lập nào trong cửa sổ Cài đặt) bằng hàm SetOption:

SetOption("MaxOpenPositions", 5); // Điều này thiết lập số lượng vị thế mở tối đa là 5

THIẾT LẬP KÍCH THƯỚC VỊ THẾ

QUAN TRỌNG: để cho phép giao dịch nhiều hơn một ký hiệu, bạn phải thêm biến PositionSize vào công thức của mình, do đó, ít hơn 100% tiền được đầu tư vào một chứng khoán duy nhất:

PositionSize = -25; // đầu tư 25% vốn chủ sở hữu danh mục đầu tư vào một giao dịch duy nhất

hoặc

PositionSize = 5000; // đầu tư 5000 đô la vào một giao dịch duy nhất

Có một cách khá phổ biến để thiết lập cả quy mô vị thế và số lượng vị thế mở tối đa để vốn chủ sở hữu được phân bổ đều giữa các giao dịch:

PosQty = 5; // Bạn có thể xác định số lượng vị thế mở mà bạn muốn
SetOption("MaxOpenPositions", PosQty);

PositionSize = -100/PosQty; // đầu tư 100% vốn chủ sở hữu danh mục đầu tư chia cho số lượng vị thế tối đa

Bạn cũng có thể sử dụng các phương pháp định cỡ vị thế phức tạp hơn. Ví dụ, định cỡ vị thế dựa trên biến động (kiểu Van Tharp):

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

Theo cách đó, bạn đang đầu tư 2% vốn chủ sở hữu DANH MỤC vào giao dịch được điều chỉnh theo hệ số BuyPrice/2*ATR.

SỬ DỤNG ĐIỂM SỐ VỊ TRÍ

Bạn có thể sử dụng biến PositionScore mới để quyết định giao dịch nào nên được nhập nếu có nhiều tín hiệu nhập trên các chứng khoán khác nhau hơn số lượng tối đa cho phép của các vị thế mở hoặc quỹ khả dụng. Trong trường hợp đó, AmiBroker sẽ sử dụng giá trị tuyệt đối của biến PositionScore để quyết định giao dịch nào được ưu tiên. Xem mã bên dưới. Nó triển khai hệ thống giao cắt MA đơn giản, nhưng có thêm hương vị ưu tiên nhập giao dịch trên các ký hiệu có giá trị RSI thấp. Nếu có nhiều tín hiệu mua hơn tiền mặt khả dụng/vị thế tối đa thì cổ phiếu có RSI thấp hơn sẽ được ưu tiên. Bạn có thể theo dõi quá trình lựa chọn nếu bạn kiểm tra ngược với chế độ báo cáo "Nhật ký chi tiết" được bật.

Mã bên dưới cũng bao gồm ví dụ về cách tìm số lượng tối ưu các vị thế mở đồng thời bằng Tối ưu hóa mới trong chế độ Danh mục đầu tư.

/*****

** Chế độ DANH MỤC ĐẦU TƯ THƯỜNG XUYÊN

** Tối ưu hóa mẫu này

** tìm số lượng tối ưu các vị thế mở đồng thời

**

```
****/

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

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

// kích thước vị thế mong muốn là 100% vốn chủ sở hữu danh mục đầu tư
// chia cho các vị thế PosQty PositionSize = -100/posqty;
// Hệ thống rất đơn giản...
// Các tham số MA cũng có thể được tối ưu hóa... p1 = 10;

p2 = 22;
// MA giao thoa đơn giản
Short=Cross( MA(C,p1) , MA(C,p2) );
Buy=Cross( MA(C,p2) , MA(C,p1) );
// luôn ở trong thị trường Sell=Short;

Cover=Buy;

// giờ là điểm số bổ sung
// được sử dụng để xếp hạng cổ phiếu
// khi có nhiều tín hiệu ENTRY hơn
// vị thế/tiền mặt
PositionScore = 100-RSI(); // ưu tiên các cổ phiếu có RSI thấp;
```

CHẾ ĐỘ KIỂM TRA NGƯỢC

AmiBroker 5.0 cung cấp 6 chế độ kiểm tra ngược khác nhau:

· chế độ thông thường (backtestRegular)

· chế độ thô thông thường (backtestRegularRaw)

· chế độ thô thông thường + nhiều vị thế (backtestRegularRawMulti)

· chế độ thô thông thường 2 (backtestRegularRaw2)

· chế độ thô thông thường + nhiều vị thế (backtestRegularRaw2Multi)

· chế độ giao dịch luân phiên (backtestRotational)

Tất cả các chế độ "thông thường" đều sử dụng tín hiệu mua/bán/bán khống/bảo hiểm để vào/thoát giao dịch, trong khi chế độ "luân phiên" (hay còn gọi là hệ thống "xếp hạng/chuyển đổi") chỉ sử dụng điểm số vị thế và được mô tả sau.

Chế độ kiểm tra ngược có thể chuyển đổi bằng hàm SetBacktestMode() AFL.

Sự khác biệt giữa các chế độ "thông thường" là cách xử lý các tín hiệu vào lệnh lặp lại (còn được gọi là "dư thừa" hoặc "thêm"). Tín hiệu vào lệnh "thêm" là tín hiệu đến SAU lệnh vào lệnh ban đầu nhưng trước tín hiệu thoát lệnh khớp lệnh đầu tiên.

Ở chế độ thông thường - chế độ mặc định, các tín hiệu vào lệnh dư thừa sẽ bị xóa như trong hình bên dưới.

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	

Như bạn thấy, các cặp tín hiệu Mua-Bán được khớp lệnh và được coi là GIAO DỊCH. Nếu giao dịch KHÔNG được nhập lệnh vào lệnh đầu tiên do thứ hạng yếu, không đủ tiền mặt hoặc đạt đến số lượng vị thế mở tối đa, các tín hiệu vào lệnh tiếp theo sẽ bị bỏ qua cho đến khi tín hiệu thoát lệnh khớp lệnh. Sau tín hiệu thoát lệnh, tín hiệu vào lệnh tiếp theo sẽ có

thể là ứng cử viên để vào lệnh giao dịch. Quy trình xóa các tín hiệu dư thừa xảy ra sau lệnh mua và lệnh bán khớp lệnh đầu tiên (và cặp lệnh bán khống tương ứng) giống như hàm ExRem() AFL cung cấp. Để sử dụng chế độ thông thường, bạn không cần phải gọi hàm SetBacktestMode vì đây là chế độ mặc định.

Bạn có thể cân nhắc hoặc không cân nhắc việc xóa các tín hiệu bổ sung là mong muốn. Nếu bạn muốn hành động trên BẤT KỲ tín hiệu nhập nào, bạn cần sử dụng chế độ thứ hai - backtestRegularRaw. Để bật chế độ này, bạn cần đưa dòng này vào mã:

// kiểm tra ngược dựa trên tín hiệu, các tín hiệu dư thừa (thô) KHÔNG bị xóa, chỉ được phép một vị trí cho mỗi ký hiệu

SetBacktestMode(backtestRegularRaw);

Chế độ này KHÔNG xóa các tín hiệu nhập dư thừa và sẽ hành động trên BẤT KỲ mục nhập nào miễn là mục nhập đó được chấm điểm đủ cao và có tiền mặt khả dụng và số lượng vị thế mở tối đa chưa đạt đến. Tuy nhiên, chế độ này chỉ cho phép MỘT VỊ THẾ MỞ cho mỗi ký hiệu tại bất kỳ thời điểm nào. Điều này có nghĩa là nếu giao dịch nhật ký đã mở và sau đó trong chuỗi xuất hiện thêm tín hiệu mua, tín hiệu này sẽ bị bỏ qua cho đến khi có tín hiệu "bán" (tín hiệu che phủ ngắn hạn cũng hoạt động tương tự). Lưu ý rằng bạn vẫn có thể sử dụng sigScaleIn/sigScaleOut để tăng hoặc giảm kích thước của vị thế hiện tại này, nhưng nó sẽ xuất hiện dưới dạng một dòng trong danh sách kết quả kiểm tra ngược.

Nếu bạn muốn TẤT CẢ các tín hiệu nhập lặp lại được thực hiện và cho phép mở nhiều vị thế riêng biệt trên cùng một

ký hiệu mà không có hiệu ứng mở rộng/mở rộng (do đó nhiều vị thế trên cùng một ký hiệu mở đồng thời xuất hiện dưới dạng các dòng riêng biệt trong báo cáo kiểm tra ngược), bạn cần sử dụng chế độ backtestRegularRawMulti bằng cách thêm dòng sau vào mã:

SetBacktestMode(backtestRegularRawMulti);

Trong chế độ này, NHIỀU vị thế trên mỗi ký hiệu sẽ được mở nếu tín hiệu MUA/BÁN là "đúng" đối với nhiều hơn một thanh và có tiền miễn phí. Bán/Bọc thoát tất cả các vị thế mở trên ký hiệu đã cho, Scale-In/Out hoạt động trên tất cả các vị thế mở của ký hiệu đã cho cùng một lúc.

Lưu ý: Các chế độ còn lại chỉ dành cho người dùng nâng cao

Chế độ Raw2 là "đặc biệt" dành cho người dùng nâng cao của backtester tùy chỉnh. Chúng chỉ hữu ích nếu bạn thực hiện xử lý tùy chỉnh các tín hiệu thoát trong quy trình backtester

tùy chỉnh. Chúng KHÔNG NÊN được sử dụng nếu không, vì chế độ Raw2 gây ảnh hưởng đến hiệu suất và tiêu thụ bộ nhớ.

Điểm chung giữa chế độ Raw và Raw2 là cả hai đều KHÔNG xóa các tín hiệu ENTRY dư thừa. Sự khác biệt là chế độ Raw xóa các tín hiệu EXIT dư thừa, trong khi Raw2 KHÔNG làm vậy.

Trong chế độ Raw2, tất cả các tín hiệu thoát (kể cả tín hiệu dự phòng) đều được chuyển sang giai đoạn thứ hai của backtest trong trường hợp bạn muốn triển khai chiến lược bỏ qua lần thoát đầu tiên. Giả sử bạn muốn thoát theo một số điều kiện nào đó từ giai đoạn đầu tiên nhưng chỉ trong một số giờ nhất định hoặc sau một số thanh nhất định trong giao dịch hoặc chỉ khi điều kiện vốn chủ sở hữu danh mục đầu tư được đáp ứng. Bây giờ bạn có thể làm điều đó trong chế độ Raw2.

Lưu ý rằng chế độ Raw2 có thể chậm hơn đáng kể khi bạn sử dụng mã backtester tùy chỉnh lặp lại qua các tín hiệu vì có thể có hàng tỷ tín hiệu thoát trong danh sách ngay cả đối với các ký hiệu không bao giờ tạo ra bất kỳ tín hiệu nhập nào, do đó, bạn chỉ nên sử dụng khi thực sự cần thiết. Chế độ Raw2 cũng là chế độ tốn nhiều bộ nhớ nhất. Lưu ý rằng nếu bạn chạy hệ thống KHÔNG có quy trình kiểm tra ngược tùy chỉnh thì sẽ không có sự khác biệt nào giữa chế độ Raw và Raw2 (ngoại trừ tốc độ và mức sử dụng bộ nhớ) vì tín hiệu thoát khớp đầu tiên là tín hiệu được sử dụng theo mặc định.

GIAO DỊCH LUÂN CHUYỂN

Giao dịch luân chuyển (còn được gọi là chuyển đổi quỹ hoặc tính điểm và xếp hạng) cũng khả thi. Để biết thêm thông tin, hãy xem mô tả về hàm EnableRotationalTrading.

HOLDMINBARS và PHÍ THOÁT SỚM

(Lưu ý rằng các tính năng này chỉ khả dụng trong portfolio-backtester và không tương thích với backtester cũ hoặc hàm Equity())

HoldMinBars là tính năng vô hiệu hóa thoát trong số lượng thanh do người dùng chỉ định ngay cả khi tín hiệu/lệnh dừng được tạo trong khoảng thời gian đó

Xin lưu ý rằng NẾU trong khoảng thời gian HoldMinBars, BẤT KỲ lệnh dừng nào được tạo ra thì lệnh đó sẽ bị bỏ qua. Ngoài ra, khoảng thời gian này bị bỏ qua khi tính toán các điểm dừng theo sau (các mức cao nhất mới và mức giảm dưới các điểm dừng theo sau được tạo ra trong HoldMinBars sẽ bị bỏ qua). Cài đặt này, tương tự như EarlyExitFee/EarlyExitBars, có sẵn trên cơ sở mỗi ký hiệu (tức là có thể được đặt thành giá trị khác nhau cho mỗi ký hiệu)

Ví dụ:

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

// ngay cả khi các tín hiệu bán được tạo ra mỗi ngày,
//chúng vẫn bị bỏ qua cho đến thanh 128
```

Phí thoát sớm (hoàn trả) được tính khi giao dịch thoát trong N thanh đầu tiên kể từ khi vào lệnh.

Phí này được thêm vào hoa hồng thoát và bạn sẽ thấy trong hoa hồng được báo cáo, ví dụ như trong nhật ký chi tiết. Tuy nhiên, nó KHÔNG được phản ánh trong vốn chủ sở hữu danh mục đầu tư trừ khi giao dịch thực sự thoát trong N thanh đầu tiên - điều này nhằm ngăn chặn ảnh hưởng đến việc rút tiền nếu giao dịch KHÔNG thoát sớm.

```
// hai tùy chọn mới này có thể được thiết lập theo từng ký hiệu
// bao nhiêu thanh (ngày giao dịch)
// phí thoát sớm (hoàn trả) được áp dụng SetOption("EarlyExitBars", 128 );
// phí hoàn trả sớm (theo phần trăm) SetOption("EarlyExitFee", 2 );
```

(lưu ý 180 ngày dương lịch là 128 hoặc 129 ngày giao dịch)

```
// làm thế nào để thiết lập theo từng ký hiệu?
// rất đơn giản - sử dụng câu lệnh 'if'
if( Name() == "SYMBOL1" )
{
SetOption("EarlyExitBars", 128 );
SetOption("EarlyExitFee", 2 );
}

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

```
}
```

Ngoài HoldMinBars, EarlyExitBars còn có các tính năng liên quan (được thêm vào trong 4.90) có tên là HoldMinDays và EarlyExitDays hoạt động với các ngày trong lịch thay vì các thanh dữ liệu. Vì vậy, chúng ta có thể viết lại các ví dụ trước đó để sử dụng các ngày trong lịch một cách chính xác:

```
// ngay cả khi các tín hiệu bán được tạo ra mỗi ngày,
// chúng sẽ bị bỏ qua cho đến 180 ngày trong lịch kể từ khi nhập SetOption("HoldMinBars", 180 );

Mua=BarIndex()==0; Bán=1;

// hai tùy chọn mới này có thể được đặt trên cơ sở mỗi ký hiệu
// có bao nhiêu NGÀY TRONG LỊCH
// phí thoát sớm (hoàn lại) được áp dụng SetOption("EarlyExitDays", 180 );
// phí hoàn lại sớm (theo phần trăm) SetOption("EarlyExitFee", 2 );

(lưu ý 180 ngày trong lịch là 128 hoặc 129 ngày giao dịch)

// làm thế nào để thiết lập trên cơ sở mỗi ký hiệu?

// rất đơn giản - sử dụng câu lệnh 'if'
if( Name() == "SYMBOL1" )
{
SetOption("EarlyExitDays", 180 );
SetOption("EarlyExitFee", 2 );
}

if( Name() == "SYMBOL2" )
{
SetOption("EarlyExitDays", 30 );
SetOption("EarlyExitFee", 1 );
```

}

GIẢI QUYẾT CÁC XUNG ĐỘT TÍN HIỆU CÙNG MỘT THANH, CÙNG MỘT KÝ HIỆU

Hệ thống có thể tạo ra trên cùng một ký hiệu cả tín hiệu vào và tín hiệu ra tại cùng một thanh. Ví dụ, hãy xem xét hệ thống rất đơn giản này tạo ra tín hiệu mua và bán trên mọi thanh:

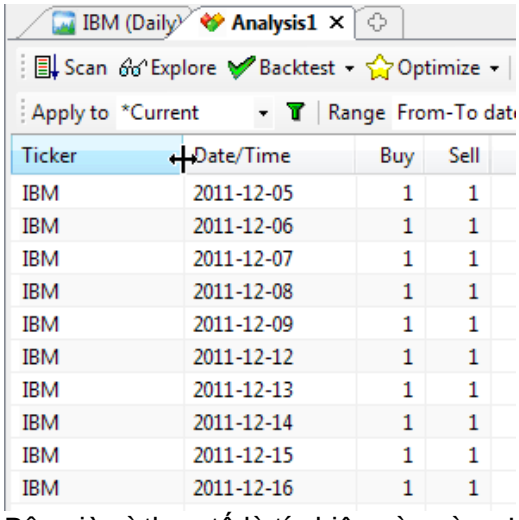
Mua = 1;

Bán = 1;

Nếu bạn thêm mã khám phá vào đó để hiển thị các tín hiệu:

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

bạn sẽ nhận được kết quả sau (khi bạn nhấn Explore);



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

Bây giờ vì thực tế là tín hiệu vào và ra KHÔNG mang theo bất kỳ thông tin thời gian nào, vì vậy bạn không biết tín hiệu nào đến trước, có ba cách để giải thích cùng một thanh, tín hiệu vào và ra xung đột như vậy:

1. chỉ có một tín hiệu được thực hiện tại bất kỳ thanh nào, vì vậy giao dịch bắt đầu tại thanh 1 kết thúc tại thanh 2 và giao dịch tiếp theo chỉ có thể mở tại thanh 3 và đóng tại thanh 4

2. cả hai tín hiệu đều được sử dụng và tín hiệu vào đi trước tín hiệu thoát, vì vậy giao dịch bắt đầu tại thanh 1 kết thúc tại thanh 1, sau đó giao dịch tiếp theo mở tại thanh 2 và kết thúc

tại thanh 2, v.v. (chúng ta có giao dịch thanh đơn và chúng ta không vào thị trường giữa các thanh)

3. cả hai tín hiệu đều được sử dụng và tín hiệu vào đi sau tín hiệu thoát. Trong tình huống này, tín hiệu đầu tiên (thoát) bị bỏ qua vì chúng ta đang đi ngang và giao dịch mở tại cùng một tín hiệu vào thanh. Sau đó, chúng ta không có thêm tín hiệu nào cho thanh đã cho và giao dịch được đóng tại tín hiệu thoát thanh tiếp theo, sau đó chúng ta có một mục nhập khác (cùng một thanh). Vì vậy, giao dịch bắt đầu ở thanh 1 kết thúc ở thanh 2, sau đó giao dịch tiếp theo mở ở thanh 2 và kết thúc ở thanh 3, v.v. (chúng ta có các giao dịch kéo dài giữa các thanh, nhưng cả tín hiệu thoát và nhập xảy ra trên cùng một thanh đều được tác động)

Vì, như chúng tôi đã đề cập, các mảng mua/bán/bán khống/bảo hiểm không mang thông tin về thời gian, chúng ta phải nói cho AmiBroker biết cách diễn giải xung đột như vậy. Người ta nghĩ rằng chỉ cần đặt giá mua mở cửa và giá bán đóng cửa để cung cấp thông tin về thời gian là đủ, nhưng KHÔNG phải vậy. Bản thân các mảng giá cũng KHÔNG cung cấp thông tin về thời gian. Bạn có thể hỏi tại sao. Điều này khá đơn giản, trước hết, giá giao dịch không cần phải được đặt thành giá mở/đóng chính xác. Trong một số trường hợp, bạn có thể muốn định nghĩa giá mua là giá mở cửa + trượt giá và giá bán là giá đóng cửa - trượt giá. Ngay cả khi bạn sử dụng giá mở cửa và giá đóng cửa chính xác, thì vẫn thường xảy ra tình trạng giá mở cửa bằng giá đóng cửa (như trong nến doji) và sau đó không có cách nào để tìm ra chỉ từ giá, bất cứ khi nào nó có nghĩa là giá đóng cửa hay giá mở cửa. Vì vậy, một lần nữa, các biến buyprice/sellprice/shortprice/coverprice KHÔNG cung cấp bất kỳ thông tin thời gian nào.

Cách duy nhất để kiểm soát cách giải quyết xung đột giữa cùng một thanh, cùng một ký hiệu vào/ra là thông qua tùy chọn

AllowSameBarExit và tùy chọn HoldMinBars.

Kịch bản 1. Chỉ có một tín hiệu cho mỗi ký hiệu được lấy tại bất kỳ thanh nào

Kịch bản này được sử dụng khi tùy chọn AllowSameBarExit được đặt thành False (tắt).

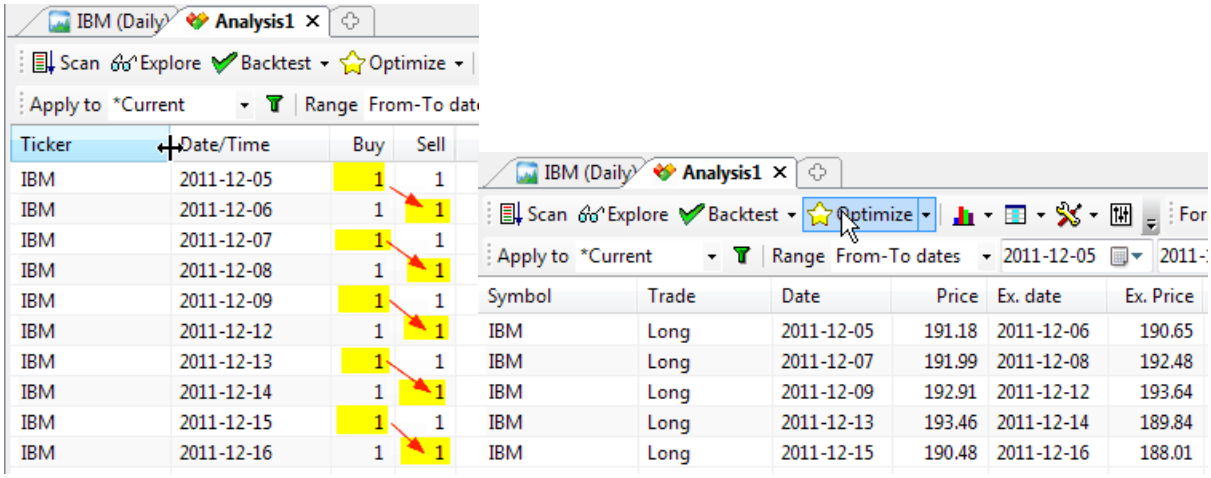
Trong trường hợp này, thực sự không quan trọng lệnh thoát hay lệnh nhập là lệnh đầu tiên trong một thanh. Điều này khá dễ hiểu: trên bất kỳ thanh nào, chỉ có một tín hiệu được tác động. Vì vậy, nếu chúng ta bằng phẳng trên ký hiệu đã cho, thì tín hiệu vào được lấy (với tín hiệu mua được ưu tiên hơn lệnh bán), các tín hiệu khác bị bỏ qua và chúng ta chuyển sang thanh tiếp theo. Nếu chúng ta mua vào ký hiệu đã cho, thì tín hiệu bán được lấy, giao dịch được thoát và chúng ta chuyển sang thanh tiếp theo mà bỏ qua các tín hiệu khác. Nếu chúng ta bán khống trên ký hiệu đã cho, thì tín hiệu bảo hiểm được lấy, giao dịch được thoát và chúng ta lại chuyển sang thanh tiếp theo mà bỏ qua các tín hiệu khác. Nếu chúng ta đang ở trên thị trường nhưng không có tín hiệu thoát lệnh phù hợp - vị thế được giữ nguyên và chúng ta chuyển sang thanh tiếp theo.

SetOption("AllowSameBarExit", False);

Buy = 1;

Sell = 1;

Các hình ảnh sau đây cho thấy tín hiệu nào được thực hiện và danh sách giao dịch kết quả. Tất cả các giao dịch bắt đầu một ngày và kết thúc vào ngày hôm sau. Giao dịch mới được mở vào ngày hôm sau.



Kịch bản 2. Cả tín hiệu vào lệnh và tín hiệu thoát lệnh đều được sử dụng và tín hiệu vào lệnh đi trước tín hiệu thoát lệnh

Kịch bản này được sử dụng khi tùy chọn AllowSameBarExit được đặt thành True (bật) và HoldMinBars được đặt thành 0 (là cài đặt mặc định).

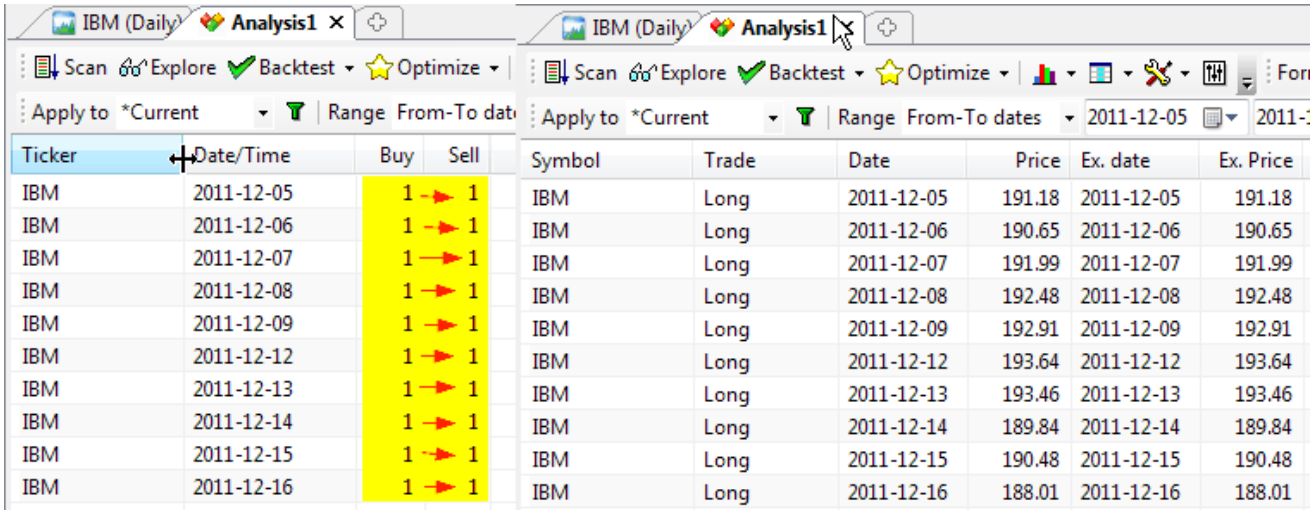
Trong trường hợp này, chúng ta chỉ cần hành động ngay lập tức theo cả hai tín hiệu (cùng một thanh). Vì vậy, nếu chúng ta bằng phẳng trên ký hiệu đã cho, thì tín hiệu vào lệnh được thực hiện (với tín hiệu mua được ưu tiên hơn tín hiệu bán), nhưng chúng ta không chuyển sang thanh tiếp theo ngay lập tức. Thay vào đó, chúng ta kiểm tra xem tín hiệu thoát lệnh có tồn tại không. Nếu chúng ta mua vào ký hiệu đã cho, thì tín hiệu bán được thực hiện. Nếu chúng ta bán khống trên ký hiệu đã cho, thì tín hiệu che lệnh được thực hiện. Chỉ sau khi xử lý tất cả các tín hiệu, chúng ta mới chuyển sang thanh tiếp theo.

SetOption("AllowSameBarExit", True);

Mua = 1;

Bán = 1;

Các hình ảnh sau đây cho thấy tín hiệu nào được thực hiện và danh sách giao dịch kết quả. Như chúng ta có thể thấy, lần này tất cả các tín hiệu đều được tác động và chúng ta có chuỗi các giao dịch thanh đơn.



Kịch bản 3. Cả hai tín hiệu đều được sử dụng và tín hiệu vào lệnh xuất hiện sau tín hiệu thoát lệnh.

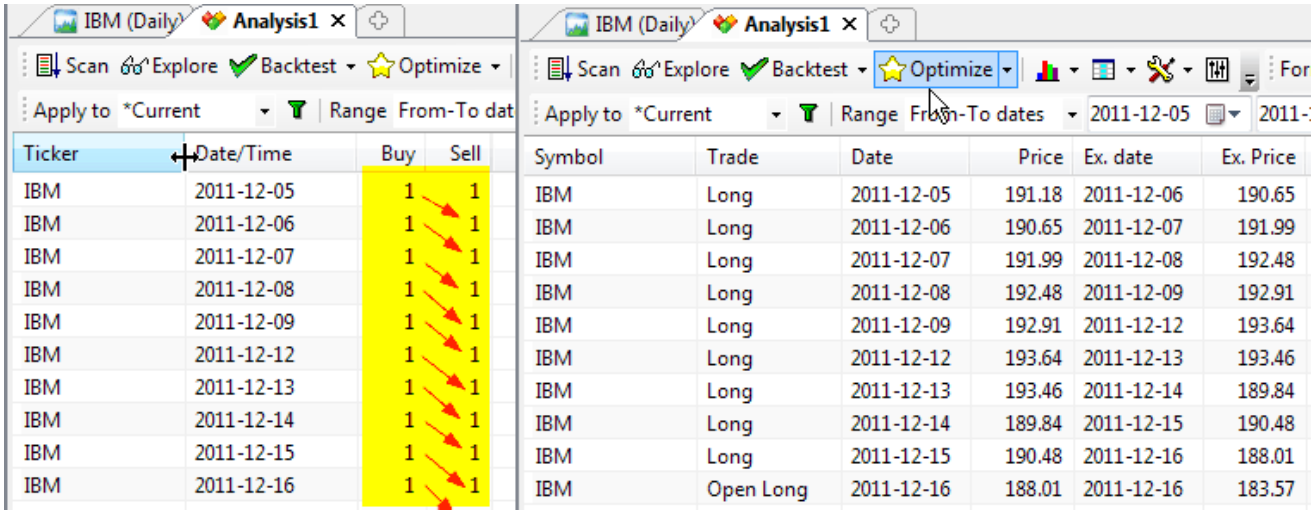
Kịch bản này được sử dụng khi tùy chọn AllowSameBarExit được đặt thành True (bật) và HoldMinBars được đặt thành 1 (hoặc nhiều hơn).

Trong trường hợp này, chúng ta chỉ cần tác động đến cả hai tín hiệu trong thanh đơn, nhưng chúng ta tôn trọng giới hạn HoldMinBars = 1, do đó giao dịch vừa mở không thể đóng cùng thanh. Vì vậy, nếu chúng ta mua vào ký hiệu đã cho, thì tín hiệu bán sẽ được thực hiện. Nếu chúng ta bán ra ký hiệu đã cho, thì tín hiệu che lệnh sẽ được thực hiện. Chúng ta chưa chuyển sang thanh tiếp theo. Bây giờ nếu chúng ta đang ở trạng thái bằng phẳng trên biểu tượng đã cho (có thể chỉ thoát khỏi vị thế trên tín hiệu thoát thanh này), thì tín hiệu vào lệnh được thực hiện nếu có (với tín hiệu mua được ưu tiên hơn tín hiệu bán khống) và sau đó chúng ta chuyển sang thanh tiếp theo.

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

Buy=1; Sell=1;
```

Các hình ảnh sau đây cho thấy tín hiệu nào được thực hiện và danh sách giao dịch kết quả. Như chúng ta có thể thấy, một lần nữa tất cả các tín hiệu đều được tác động NHƯNG... thời gian giao dịch dài hơn - chúng không phải là giao dịch cùng một thanh - tất cả đều kéo dài qua đêm.



Nó hoạt động như thế nào trong trường hợp danh mục đầu tư?

Cơ chế này giống nhau bất kể bạn thử nghiệm trên một biểu tượng hay nhiều biểu tượng. Đầu tiên, xung đột cùng một thanh được giải quyết trên từng biểu tượng riêng biệt theo cách được mô tả ở trên. Sau đó, khi bạn thử nghiệm trên nhiều biểu tượng, các ứng viên giao dịch kết quả sẽ phải chịu điểm theo PositionScore được mô tả trong phần trước của tài liệu này.

Hỗ trợ cho các chiến lược cân bằng dài-ngắn trung lập với thị trường

Một chiến lược đầu tư được coi là trung lập với thị trường nếu nó tìm cách tránh hoàn toàn một số hình thức rủi ro thị trường, thường là bằng cách phòng ngừa. Chiến lược này nắm giữ các vị thế cổ phiếu dài/ngắn, với các vị thế dài được phòng ngừa bằng các vị thế ngắn trong cùng một lĩnh vực và các lĩnh vực liên quan, do đó, nhà đầu tư trung lập với thị trường cổ phiếu sẽ ít bị ảnh hưởng bởi các sự kiện trong toàn ngành hoặc trên toàn thị trường. Về bản chất, điều này đặt cược rằng các vị thế dài sẽ vượt trội hơn các lĩnh vực của chúng (hoặc các vị thế ngắn sẽ hoạt động kém hơn) bất kể sức mạnh của các lĩnh vực.

Trong phiên bản 5.20, các tùy chọn kiểm tra ngược sau đây đã được thêm vào để đơn giản hóa việc triển khai các hệ thống trung lập với thị trường: SeparateLongShortRank, MaxOpenLong, MaxOpenShort.

Tùy chọn kiểm tra ngược SeparateLongShortRank

Để bật xếp hạng dài/ngắn riêng biệt, hãy sử dụng: SetOption("SeparateLongShortRank", True);

Khi bật xếp hạng dài/ngắn riêng biệt, kiểm tra ngược sẽ duy trì HAI danh sách tín hiệu "xếp hạng cao nhất" riêng biệt, một cho tín hiệu dài và một cho tín hiệu ngắn. Điều này đảm bảo rằng các ứng viên dài và ngắn là độc lập ngay cả khi điểm vị trí không đối xứng (ví dụ khi các ứng viên dài có điểm dương rất cao trong khi các ứng viên ngắn chỉ có điểm âm một phần). Điều đó trái ngược với chế độ mặc định trong đó chỉ có giá trị tuyệt đối của điểm vị trí là quan trọng, do đó một bên (dài/ngắn) có thể hoàn toàn thống trị thứ hạng nếu giá trị điểm không đối xứng.

Khi SeparateLongShortRank được bật, trong giai đoạn thứ hai của quá trình kiểm tra ngược, hai danh sách xếp hạng riêng biệt được xen kẽ để tạo thành danh sách tín hiệu cuối cùng bằng cách đầu tiên lấy lệnh mua dài được xếp hạng cao nhất, sau đó là lệnh bán ngắn được xếp hạng cao nhất, sau đó là lệnh mua dài được xếp hạng cao thứ 2, sau đó là lệnh bán ngắn được xếp hạng cao thứ 2, sau đó là lệnh mua dài được xếp hạng cao thứ 3 và lệnh bán ngắn được xếp hạng cao thứ 3, v.v... (miễn là các tín hiệu tồn tại trong CẢ HAI danh sách lệnh mua dài/ngắn, nếu không còn tín hiệu nào thuộc loại đã cho nữa, thì các tín hiệu còn lại từ danh sách lệnh mua dài hoặc lệnh bán ngắn sẽ được thêm vào)

Ví dụ:

Tín hiệu vào (điểm):ESRX=Mua(60,93), GILD=Ngắn(-47,56), CELG=Mua(57,68), MRVL=Ngắn(-10,75), ADBE=Mua(34,75), VRTX=Mua(15,55), SIRI=Mua(2,79),

Như bạn thấy, các tín hiệu lệnh bán ngắn được xen kẽ giữa các tín hiệu lệnh bán dài mặc dù giá trị tuyệt đối của điểm số của chúng nhỏ hơn điểm số tương ứng của các tín hiệu lệnh bán dài. Ngoài ra, chỉ có 2 tín hiệu ngắn cho thanh cụ thể đó, vì vậy, phần còn lại của danh sách hiển thị các tín hiệu dài theo thứ tự điểm vị trí. Mặc dù tính năng này có thể được sử dụng độc lập, nhưng nó được thiết kế để sử dụng kết hợp với các tùy chọn MaxOpenLong và MaxOpenShort.

Tùy chọn kiểm tra ngược MaxOpenLong / MaxOpenShort

MaxOpenLong - giới hạn số lượng vị thế DÀI có thể mở đồng thời MaxOpenShort - giới hạn số lượng vị thế NGẮN có thể mở đồng thời

Ví dụ:

```
SetOption("MaxOpenPositions", 15 );

SetOption("MaxOpenLong", 11 );

SetOption("MaxOpenShort", 7 );
```

Giá trị ZERO (mặc định) có nghĩa là KHÔNG GIỚI HẠN. Nếu cả MaxOpenLong và MaxOpenShort đều được đặt thành 0 (hoặc không được xác định) thì kiểm tra ngược hoạt động theo cách cũ - chỉ có giới hạn toàn cục đang hoạt động (MaxOpenPositions) bất kể loại giao dịch nào.

Lưu ý rằng các giới hạn này độc lập với giới hạn toàn cục (MaxOpenPositions). Điều này có nghĩa là MaxOpenLong

+ MaxOpenShort có thể bằng hoặc không bằng MaxOpenPositions.

Nếu MaxOpenLong + MaxOpenShort lớn hơn MaxOpenPositions thì tổng số vị thế được phép sẽ không vượt quá MaxOpenPositions và các giới hạn dài/ngắn riêng lẻ cũng sẽ được áp dụng. Ví dụ, nếu hệ thống MaxOpenLong của bạn được đặt thành 7 và maxOpenShort được đặt thành 7 và MaxOpenPositions được đặt thành 10 và hệ thống của bạn tạo ra 20 tín hiệu: 9 dài (xếp hạng cao nhất) và 11 ngắn, thì nó sẽ mở 7 dài và 3 ngắn.

Nếu MaxOpenLong + MaxOpenShort nhỏ hơn MaxOpenPositions (nhưng lớn hơn không), thì hệ thống sẽ không thể mở nhiều hơn (MaxOpenLong+MaxOpenShort).

Xin lưu ý rằng MaxOpenLong và MaxOpenShort chỉ giới hạn số lượng vị thế mở của loại nhất định (dài/ngắn). Chúng KHÔNG ảnh hưởng đến cách xếp hạng được thực hiện. Tức là theo mặc định, xếp hạng được thực hiện bằng cách sử dụng giá trị TUYỆT ĐỐI của positioncore.

Nếu điểm vị thế của bạn KHÔNG đối xứng, điều này có nghĩa là bạn không nhận được tín hiệu xếp hạng cao mong muốn từ một bên. Do đó, để tận dụng tối đa MaxOpenLong và MaxOpenShort trong các hệ thống cân bằng luân phiên ("trung lập thị trường") dài/ngắn, bạn nên thực hiện xếp hạng RIÊNG BIỆT cho tín hiệu dài và tín hiệu ngắn. Để bật xếp hạng dài/ngắn riêng biệt, hãy sử dụng:

SetOption("SeparateLongShortRank", True);

Xem thêm:

Bài viết về ý tưởng giao dịch của bạn.

Bài viết về hệ thống kiểm tra ngược cho hợp đồng tương lai.

Phần sử dụng trình chỉnh sửa AFL của hướng dẫn.

Hướng dẫn nội bộ về người kiểm tra ngược (bản tin 1/2002)

Đọc báo cáo kiểm tra lại

Để xem báo cáo của lần kiểm tra lại gần nhất, chỉ cần nhấp vào nút Báo cáo trong cửa sổ phân tích tự động. Để xem kết quả của TẤT CẢ các lần kiểm tra lại trước đó, hãy nhấp vào mũi tên xuống trên nút Báo cáo và chọn tùy chọn Trình duyệt Báo cáo. Điều này sẽ hiển thị cửa sổ Trình duyệt Báo cáo, nơi hiển thị danh sách tất cả các lần kiểm tra lại đã thực hiện. Nếu bạn nhấp đúp vào dòng, báo cáo chi tiết sẽ được hiển thị.

Báo cáo mới được cải tiến rất nhiều so với báo cáo cũ. Nó bao gồm các thống kê riêng cho tất cả, phía mua và phía bán cũng như nhiều chỉ số mới. Bạn có thể nhận được trợ giúp ngắn gọn về số liệu cụ thể bằng cách di chuột qua tên trường đó. Bạn sẽ thấy mô tả trong tooltip. Các giải thích ngắn gọn cũng được cung cấp bên dưới:

Tỷ lệ tiếp xúc % - 'Tiếp xúc thị trường của hệ thống giao dịch được tính theo từng thanh. Tổng tiếp xúc thanh chia cho số thanh. Tiếp xúc của một thanh là giá trị của các vị trí mở chia cho vốn chủ sở hữu danh mục.

Tỷ lệ lợi nhuận ròng điều chỉnh rủi ro % - Lợi nhuận ròng % chia cho Tỷ lệ tiếp xúc %

Tỷ lệ lợi nhuận hàng năm % - Tỷ lệ lợi nhuận hàng năm tích lũy % (CAR)

Tỷ lệ lợi nhuận điều chỉnh rủi ro % - Tỷ lệ lợi nhuận hàng năm % chia cho Tỷ lệ tiếp xúc %

Lợi nhuận/lỗ trung bình, còn được biết đến là Dự kiến ($) - (Lợi nhuận của các giao dịch thắng + Lỗ của các giao dịch thua)/(số lượng giao dịch), đại diện cho lợi nhuận/lỗ dự kiến mỗi giao dịch.

Lợi nhuận/lỗ trung bình %, còn được biết đến là Dự kiến (%) - '(% Lợi nhuận của các giao dịch thắng + % Lỗ của các giao dịch thua)/(số lượng giao dịch), đại diện cho tỷ lệ phần trăm lợi nhuận/lỗ dự kiến mỗi giao dịch.

Số thanh trung bình giữ - tổng số thanh trong các giao dịch / số lượng giao dịch.

Sự giảm giá tối đa của giao dịch - Giảm giá lớn nhất từ đỉnh đến đáy trong bất kỳ giao dịch đơn lẻ nào. Càng thấp càng tốt.

Tỷ lệ giảm giá % tối đa của giao dịch - Giảm giá phần trăm lớn nhất từ đỉnh đến đáy trong bất kỳ giao dịch đơn lẻ nào. Càng thấp càng tốt.

Sự giảm giá tối đa của hệ thống - Giảm giá lớn nhất từ đỉnh đến đáy trong vốn chủ sở hữu danh mục. Càng thấp càng tốt.

Tỷ lệ giảm giá % tối đa của hệ thống - Giảm giá phần trăm lớn nhất từ đỉnh đến đáy trong vốn chủ sở hữu danh mục. Càng thấp càng tốt.

Hệ số phục hồi - Lợi nhuận ròng chia cho Sự giảm giá tối đa của hệ thống.

CAR/MaxDD - Tỷ lệ lợi nhuận hàng năm % tích lũy chia cho Tỷ lệ giảm giá % tối đa của hệ thống. Tốt nếu lớn hơn 2.

RAR/MaxDD - Tỷ lệ lợi nhuận điều chỉnh rủi ro chia cho Tỷ lệ giảm giá % tối đa của hệ thống. Tốt nếu lớn hơn 2.

Hệ số lợi nhuận - Lợi nhuận của các giao dịch thắng chia cho lỗ của các giao dịch thua.

Tỷ lệ trả lại - Tỷ lệ lợi nhuận trung bình / lỗ trung bình.

Sai số chuẩn - Sai số chuẩn đo lường sự biến động của đường vốn chủ sở hữu. Càng thấp càng tốt.

Tỷ lệ Rủi ro-Lợi nhuận - Đo lường mối quan hệ giữa rủi ro tiềm ẩn trong hệ thống giao dịch so với lợi nhuận tiềm năng của nó. Càng cao càng tốt. Được tính bằng độ dốc của đường vốn chủ sở hữu (lợi nhuận hàng năm dự kiến) chia cho sai số chuẩn của nó.

Chỉ số loét - Căn bậc hai của tổng bình phương các mức giảm chia cho số lượng thanh.

Chỉ số hiệu suất loét - (Lợi nhuận hàng năm - Lợi nhuận trái phiếu kho bạc)/Chỉ số loét. Hiện tại, lợi nhuận trái phiếu kho bạc được mã hóa cứng ở mức 5.4. Trong phiên bản tương lai sẽ có tùy chọn người dùng cho điều này.

Tỷ lệ Sharpe của các giao dịch - Đo lường lợi nhuận điều chỉnh rủi ro của đầu tư. Trên 1.0 là tốt, trên 2.0 là rất tốt. Thêm thông tin tại http://www.stanford.edu/~wfsharpe/art/sr/sr.htm. Cách tính: trước tiên, tỷ lệ lợi nhuận phần trăm trung bình và độ lệch chuẩn của lợi nhuận được tính toán. Sau đó, hai số liệu này được điều chỉnh theo năm bằng cách nhân chúng với tỷ lệ (Số lượng thanh mỗi năm)/(Số thanh trung bình mỗi giao dịch). Cuối cùng, tỷ lệ lợi nhuận không rủi ro (hiện tại mã hóa cứng là 5) được trừ đi từ lợi nhuận trung bình hàng năm đã điều chỉnh và sau đó chia cho độ lệch chuẩn hàng năm của lợi nhuận.

K-Ratio - Phát hiện sự không nhất quán trong lợi nhuận. Nên đạt 1.0 trở lên. Tỷ lệ K cao hơn cho thấy lợi nhuận ổn định hơn bạn có thể mong đợi từ hệ thống. Độ dốc hồi quy tuyến tính của đường vốn chủ sở hữu nhân với căn bậc hai của tổng bình phương độ lệch của số thanh chia cho sai số chuẩn của đường vốn chủ sở hữu nhân với căn bậc hai của số lượng thanh. Thêm thông tin: Stocks & Commodities V14:3 (115-118): Đo lường hiệu suất hệ thống của Lars N. Kestner.

Mã màu trong báo cáo kiểm tra lại (mới trong 5.60)

Phiên bản 5.60 mang đến báo cáo kiểm tra lại được cải tiến: mã màu cho các giá trị 'tốt' và 'xấu' trong báo cáo kiểm tra lại. Một số chỉ số trong báo cáo kiểm tra lại được mã hóa màu. Màu xanh nghĩa là "trung tính", màu xanh lá cây nghĩa là "tốt", màu đỏ nghĩa là "xấu". Các chỉ số không được mã hóa màu luôn là màu đen.

Mã màu này tất nhiên là tùy ý và nên được sử dụng chỉ như một hướng dẫn. Hãy coi 'đỏ' như một tín hiệu cảnh báo và lời khuyên để kiểm tra giá trị chi tiết hơn.

Hiện tại, các chỉ số sau đây được mã hóa màu:

Lợi nhuận ròng, Lợi nhuận ròng % - xấu < 0, tốt > 0

Lợi nhuận hàng năm %, xấu < 0, trung tính giữa 0 và 10, tốt > 10

RAR % xấu < 0, tốt > (10 / Tỷ lệ tiếp xúc)

Lợi nhuận/lỗ trung bình của tất cả các giao dịch (Dự kiến $) - xấu < 0, tốt > 0

Lợi nhuận/lỗ trung bình % của tất cả các giao dịch (Dự kiến %) - xấu < 0, tốt > 0

Tỷ lệ giảm giá % tối đa của hệ thống - xấu: giảm giá tệ hơn -30%, trung tính: giảm giá giữa -30 và -10%, tốt: -10% đến 0%

CAR/MaxDD, RAR/MaxDD - xấu < 1, trung tính giữa 1 và 2, tốt > 2

Hệ số phục hồi - xấu < 1, trung tính giữa 1 và 2, tốt > 2

Tỷ lệ trả lại - xấu < 1, trung tính giữa 1 và 2, tốt > 2

Xem Thêm:

Báo cáo kiểm tra lại cũ

Bài viết về Kiểm tra lại các ý tưởng giao dịch.

Bài viết về Kiểm tra lại danh mục đầu tư.

Bài viết về Kiểm tra lại các hệ thống cho hợp đồng tương lai.

Sử dụng phần biên tập AFL trong hướng dẫn.

Hướng dẫn nội bộ về trình kiểm tra lại (bản tin 1/2002).

Cách tối ưu hóa hệ thống giao dịch

LƯU Ý: Đây là một chủ đề khá nâng cao. Vui lòng đọc các hướng dẫn AFL trước đó trước.

Giới thiệu

Ý tưởng về tối ưu hóa rất đơn giản. Trước tiên, bạn cần có một hệ thống giao dịch, ví dụ như một giao cắt trung bình động đơn giản. Trong hầu hết các hệ thống đều có một số tham số (như chu kỳ trung bình) quyết định cách mà hệ thống đó hoạt động (ví dụ: hệ thống này có phù hợp với giao dịch dài hạn hay ngắn hạn, nó phản ứng như thế nào với cổ phiếu có độ biến động cao, v.v.). Tối ưu hóa là quá trình tìm ra các giá trị tối ưu cho những tham số đó (đem lại lợi nhuận cao nhất từ hệ thống) cho một biểu tượng cụ thể (hoặc một danh mục các biểu tượng). AmiBroker là một trong số ít các chương trình cho phép bạn tối ưu hóa hệ thống của mình trên nhiều biểu tượng cùng một lúc.

Để tối ưu hóa hệ thống của bạn, bạn phải xác định từ một đến 64 tham số để tối ưu hóa. Bạn quyết định giá trị tối thiểu và tối đa cho tham số đó và mức tăng của giá trị này. AmiBroker sẽ thực hiện nhiều kiểm tra lại hệ thống với TẤT CẢ các kết hợp có thể của các giá trị tham số. Khi quá trình này kết thúc, AmiBroker sẽ hiển thị danh sách kết quả được sắp xếp theo lợi nhuận ròng. Bạn có thể thấy các giá trị của các tham số tối ưu hóa mang lại kết quả tốt nhất.

Viết công thức AFL

Tối ưu hóa trong trình kiểm tra lại được hỗ trợ thông qua một hàm mới gọi là optimize. Cú pháp của hàm này như sau:

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

Trong đó:

- **variable** - là biến AFL thông thường nhận giá trị trả về từ hàm optimize.
- Với các chế độ kiểm tra lại thông thường, quét, khám phá và bình luận, hàm optimize trả về giá trị mặc định, vì vậy lời gọi hàm trên tương đương với: variable = default;.
- Trong chế độ tối ưu hóa, hàm optimize trả về các giá trị liên tiếp từ min đến max (bao gồm cả).
- **"Mô tả"** là một chuỗi được sử dụng để xác định biến tối ưu hóa và được hiển thị như một tên cột trong danh sách kết quả tối ưu hóa.
- **default** là giá trị mặc định mà hàm optimize trả về trong các chế độ khám phá, chỉ báo, bình luận, quét và kiểm tra lại thông thường.
- **min** là giá trị tối thiểu của biến đang được tối ưu hóa.
- **max** là giá trị tối đa của biến đang được tối ưu hóa.
- **step** là khoảng cách được sử dụng để tăng giá trị từ min đến max.

Ghi chú:

- AmiBroker hỗ trợ tối đa 64 lần gọi hàm optimize (do đó tối đa 64 biến tối ưu hóa). Lưu ý rằng nếu bạn đang sử dụng tối ưu hóa toàn diện, thì thực sự nên giới hạn số lượng biến tối ưu hóa chỉ còn vài cái.
- Mỗi lần gọi hàm optimize tạo ra (max - min)/step vòng tối ưu hóa, và nhiều lần gọi hàm optimize sẽ nhân số lượt cần thiết. Ví dụ, tối ưu hóa hai tham số với 10 bước sẽ yêu cầu 10*10 = 100 vòng tối ưu hóa.

- Chỉ gọi hàm optimize MỘT LẦN cho mỗi biến ở đầu công thức của bạn, vì mỗi lần gọi tạo ra một vòng tối ưu hóa mới.
- Tối ưu hóa nhiều biểu tượng được AmiBroker hỗ trợ hoàn toàn.
- Không gian tìm kiếm tối đa là 2^64 (10^19 = 10,000,000,000,000,000,000) kết hợp.

Ví dụ

1. **Tối ưu hóa biến đơn:**

```
sigavg = Optimize( "Trung bình tín hiệu", 9, 2, 20, 1 );


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

2. **Tối ưu hóa hai biến (thích hợp cho biểu đồ 3D)**

```
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. **Tối ưu hóa nhiều biến (3):**

```
mfast = Optimize( "MACD Nhanh", 12, 8, 16, 1 );
mslow = Optimize("MACD Chậm", 26, 17, 30, 1 );
sigavg = Optimize( "Trung bình tín hiệu", 9, 2, 20, 1 );


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

Sau khi nhập công thức, chỉ cần nhấp vào nút Tối ưu hóa trong cửa sổ "Phân tích tự động". AmiBroker sẽ bắt đầu kiểm tra tất cả các kết hợp có thể của các biến tối ưu hóa và báo cáo kết quả trong danh sách. Sau khi tối ưu hóa xong, danh sách kết quả được trình bày theo thứ tự lợi nhuận ròng %. Bạn có thể sắp xếp kết quả theo bất kỳ cột nào trong danh sách kết quả để dễ dàng tìm ra các giá trị tối ưu của tham số cho mức giảm giá thấp nhất, số lượng giao dịch thấp nhất, hệ số lợi nhuận lớn nhất, tiếp xúc thị trường thấp nhất và tỷ lệ lợi nhuận hàng năm điều chỉnh rủi ro cao nhất. Các cột cuối của danh sách kết quả trình bày các giá trị của các biến tối ưu hóa cho bài kiểm tra đã cho.

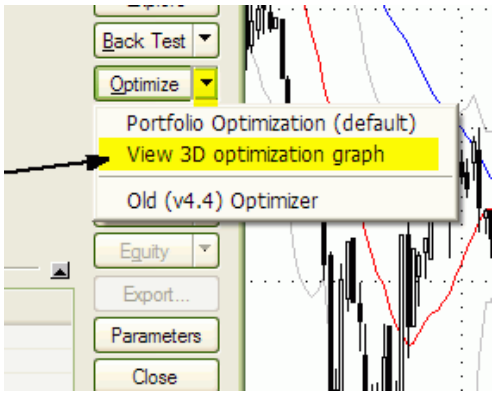
Khi bạn quyết định kết hợp tham số nào phù hợp nhất với nhu cầu của mình, tất cả những gì bạn cần làm là thay thế các giá trị mặc định trong các lời gọi hàm optimize bằng các giá trị tối ưu. Ở giai đoạn hiện tại, bạn cần nhập chúng bằng tay trong cửa sổ chỉnh sửa công thức (tham số thứ hai của lời gọi hàm optimize).

Hiển thị biểu đồ tối ưu hóa 3D hoạt hình

Để hiển thị biểu đồ tối ưu hóa 3D, bạn cần chạy tối ưu hóa hai biến trước. Tối ưu hóa hai biến cần một công thức có 2 lời gọi hàm Optimize(). Một ví dụ về công thức tối ưu hóa hai biến trông như sau:

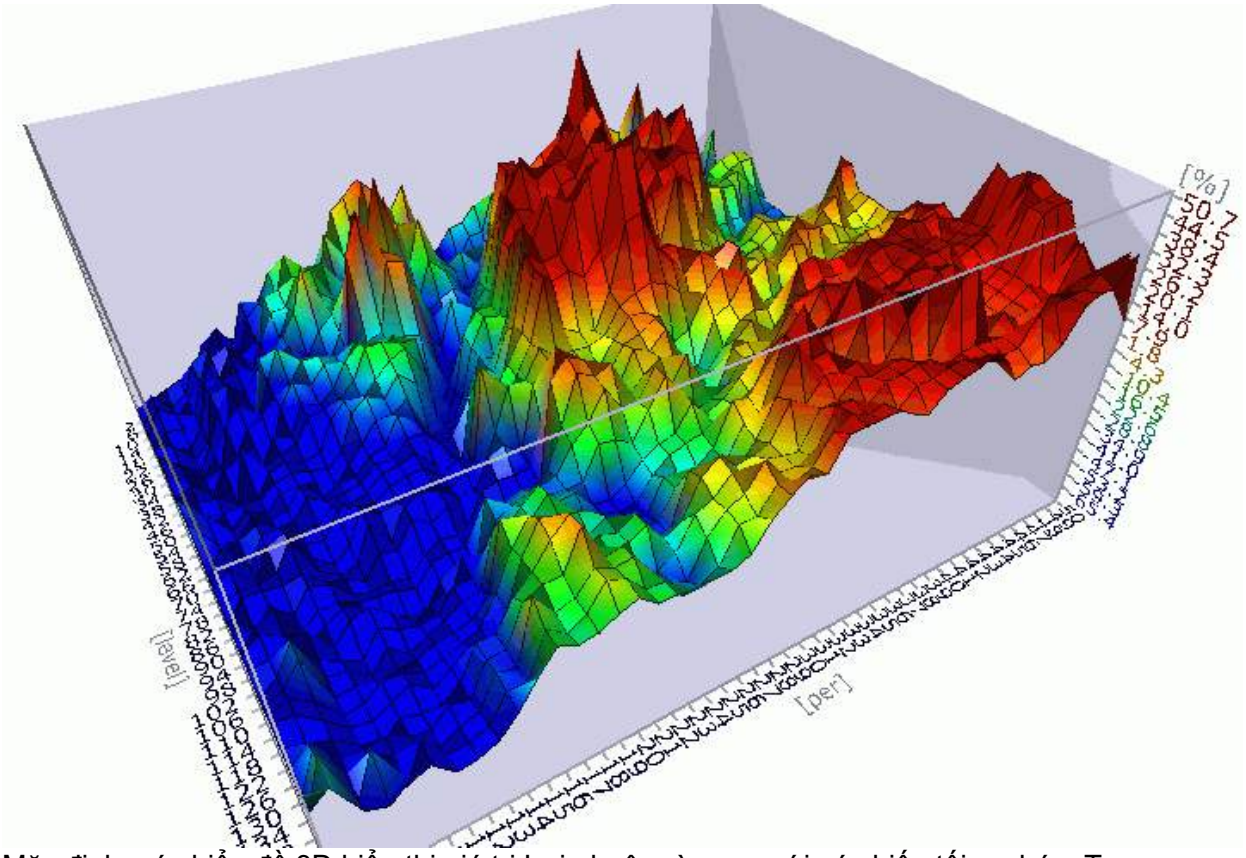
```
per = Optimize("per", 2, 5, 50, 1 );
Level = Optimize("level", 2, 2, 150, 4 );

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



Sau khi nhập công thức, bạn cần nhấp vào nút "Tối ưu hóa".

Khi quá trình tối ưu hóa hoàn tất, bạn nên nhấp vào mũi tên xuống trên nút Tối ưu hóa và chọn Xem biểu đồ tối ưu hóa 3D. Trong vài giây, một đồ thị bề mặt ba chiều đầy màu sắc sẽ xuất hiện trong cửa sổ xem biểu đồ 3D. Một ví dụ về biểu đồ 3D được tạo ra từ công thức trên được hiển thị bên dưới.



Mặc định, các biểu đồ 3D hiển thị giá trị Lợi nhuận ròng so với các biến tối ưu hóa. Tuy nhiên, bạn có thể vẽ biểu đồ bề mặt 3D cho bất kỳ cột nào trong bảng kết quả tối ưu hóa. Chỉ cần nhấp vào tiêu đề cột để sắp xếp (mũi tên xanh sẽ xuất hiện cho biết rằng kết quả tối ưu hóa đã được sắp xếp theo cột đã chọn) và sau đó chọn Xem biểu đồ tối ưu hóa 3D một lần nữa.

Bằng cách trực quan hóa cách mà các tham số của hệ thống ảnh hưởng đến hiệu suất giao dịch, bạn có thể dễ dàng quyết định những giá trị tham số nào tạo ra hiệu suất hệ thống

"mong manh" và những giá trị nào tạo ra hiệu suất "vững chắc". Các cài đặt vững chắc là các vùng trong biểu đồ 3D cho thấy những thay đổi dần dần thay vì đột ngột trong đồ thị bề mặt. Biểu đồ tối ưu hóa 3D là công cụ tuyệt vời để ngăn chặn việc điều chỉnh quá mức. Việc điều chỉnh quá mức xảy ra khi hệ thống phức tạp hơn mức cần thiết, và tất cả sự phức tạp đó được tập trung vào các điều kiện thị trường có thể không bao giờ xảy ra nữa. Những thay đổi đột ngột (hoặc đỉnh) trong biểu đồ tối ưu hóa 3D cho thấy rõ ràng các khu vực điều chỉnh quá mức.

Bắt đầu nhanh

Đây là hướng dẫn ĐƠN GIẢN NHẤT về cách sử dụng trình tối ưu hóa không đầy đủ mới (trong trường hợp này là CMA-ES).

1. Mở công thức của bạn trong Trình chỉnh sửa công thức

2. Thêm dòng đơn này vào đầu công thức của bạn: OptimizerSetEngine("cmae"); // bạn cũng có thể sử dụng "spso" hoặc "trib" tại đây
3. (Tùy chọn) Chọn mục tiêu tối ưu hóa của bạn trong Phân tích tự động, Cài đặt, tab "Walk-Forward", trường Mục tiêu tối ưu hóa. Nếu bạn bỏ qua bước này, nó sẽ tối ưu hóa cho CAR/MDD (lợi nhuận hàng năm kép chia cho mức giảm % tối đa).

và... thế là xong.

Bây giờ, nếu bạn chạy tối ưu hóa bằng công thức này, nó sẽ sử dụng trình tối ưu hóa CMA-ES tiến hóa mới (không đầy đủ).

Nó hoạt động như thế nào?

Tối ưu hóa là quá trình tìm giá trị tối thiểu (hoặc tối đa) của hàm đã cho. Bất kỳ hệ thống giao dịch nào cũng có thể được coi là một hàm của một số đối số nhất định. Đầu vào là các tham số và dữ liệu báo giá, đầu ra là mục tiêu tối ưu hóa của bạn
(ví dụ CAR/MDD). Và bạn đang tìm kiếm giá trị cực đại của hàm đã cho.

Một số thuật toán tối ưu hóa thông minh dựa trên bản chất (hành vi động vật) - thuật toán PSO hoặc quá trình sinh học - thuật toán di truyền,
và một số dựa trên các khái niệm toán học do con người đưa ra - CMA-ES.

Các thuật toán này được sử dụng trong nhiều lĩnh vực khác nhau, bao gồm cả tài chính. Nhập "tài chính PSO" hoặc "tài chính CMA-ES" vào Google và bạn sẽ tìm thấy rất nhiều thông tin.

Các phương pháp không đầy đủ (hoặc "thông minh") sẽ tìm ra tối ưu toàn cục hoặc cục bộ. Tất nhiên, mục tiêu là tìm ra tối ưu toàn cục, nhưng nếu có một đỉnh nhọn duy nhất
trong số hàng triệu kết hợp tham số, các phương pháp không đầy đủ có thể không tìm thấy đỉnh nhọn duy nhất này, nhưng xét theo quan điểm của nhà giao dịch, việc tìm ra đỉnh nhọn duy nhất là vô ích đối với giao dịch vì kết quả đó sẽ không ổn định (quá mong manh) và không thể sao chép trong giao dịch thực tế. Trong quá trình tối ưu hóa, chúng ta đang tìm

kiếm các vùng ổn định với các tham số ổn định và đây là lĩnh vực mà các phương pháp thông minh tỏa sáng.

Đối với thuật toán được sử dụng bởi tìm kiếm không đầy đủ, nó trông như sau:

a) trình tối ưu hóa tạo ra một số quần thể tham số ban đầu (thường là ngẫu nhiên)
b) AmiBroker thực hiện kiểm tra ngược cho từng tập tham số từ quần thể
c) kết quả kiểm tra ngược được đánh giá theo logic của thuật toán và quần thể mới được tạo ra dựa trên sự tiến triển của kết quả,
d) nếu tìm thấy kết quả tốt nhất mới - hãy lưu lại và chuyển sang bước b) cho đến khi đáp ứng được tiêu chí dừng

Ví dụ về tiêu chí dừng có thể bao gồm:
a) đạt đến số lần lặp tối đa đã chỉ định
b) dừng nếu phạm vi giá trị mục tiêu tốt nhất của X thế hệ cuối cùng bằng không
c) dừng nếu việc thêm vectơ độ lệch chuẩn 0,1 theo bất kỳ hướng trục chính nào không làm thay đổi giá trị của giá trị mục tiêu

d) các tiêu chí khác

Để sử dụng bất kỳ trình tối ưu hóa thông minh (không đầy đủ) nào trong AmiBroker, bạn cần chỉ định công cụ tối ưu hóa mà bạn muốn sử dụng trong công thức AFL bằng hàm OptimizerSetEngine.

OptimizerSetEngine("name")

Hàm này chọn công cụ tối ưu hóa bên ngoài được xác định theo tên. AmiBroker hiện được cung cấp với 3 công cụ: Standard Particle Swarm Optimizer ("spso"), Tribes ("trib") và CMA-ES (''cmae") - các tên trong dấu ngoặc nhọn sẽ được sử dụng trong các lệnh gọi OptimizerSetEngine.

Ngoài việc chọn công cụ tối ưu hóa, bạn có thể muốn đặt một số tham số nội bộ của công cụ này. Để thực hiện việc này, hãy sử dụng hàm OptimizerSetOption.

Hàm OptimizerSetOption("name", value)

Hàm này đặt các tham số bổ sung cho công cụ tối ưu hóa bên ngoài. Các tham số phụ thuộc vào công cụ.
Cả ba công cụ tối ưu hóa được cung cấp cùng với AmiBroker (SPSO, Trib, CMAE) đều hỗ trợ hai tham số: "Runs" (số lần chạy) và "MaxEval" (đánh giá tối đa (kiểm tra) cho mỗi lần chạy). Hành vi của từng tham số phụ thuộc vào công cụ, do đó, cùng một giá trị có thể và thường sẽ tạo ra các kết quả khác nhau với các công cụ khác nhau được sử dụng.

Sự khác biệt giữa Runs và MaxEval như sau. Đánh giá (hoặc kiểm tra) là một lần kiểm tra ngược (hoặc đánh giá giá trị hàm mục tiêu).
RUN là một lần chạy đầy đủ của thuật toán (tìm giá trị tối ưu) - thường bao gồm nhiều lần kiểm tra (đánh giá).

Mỗi lần chạy chỉ đơn giản là KHỞI ĐỘNG LẠI toàn bộ quá trình tối ưu hóa từ điểm bắt đầu mới (dân số ngẫu nhiên ban đầu mới).
Do đó, mỗi lần chạy có thể dẫn đến việc tìm ra giá trị cực đại/tần số cục bộ khác nhau (nếu không tìm thấy giá trị toàn cục). Vì vậy, tham số Runs xác định số lần chạy thuật toán tiếp theo. MaxEval là số lượng đánh giá tối đa (bactest) trong bất kỳ lần chạy nào.

Nếu vấn đề tương đối đơn giản và 1000 lần kiểm tra là đủ để tìm giá trị cực đại toàn cục, thì 5x1000 có nhiều khả năng tìm thấy giá trị cực đại toàn cục hơn vì khả năng bị kẹt trong giá trị cực đại cục bộ thấp hơn, vì các lần chạy tiếp theo sẽ bắt đầu từ dân số ngẫu nhiên ban đầu khác nhau

Việc lựa chọn giá trị tham số có thể khó khăn. Nó phụ thuộc vào vấn đề đang được kiểm tra, độ phức tạp của nó, v.v.
Bất kỳ phương pháp không đầy đủ ngẫu nhiên nào cũng không đảm bảo bạn tìm thấy giá trị max/min toàn cục, bất kể số lượng kiểm tra nếu nó nhỏ hơn
toàn diện. Câu trả lời dễ nhất là: chỉ định số lượng kiểm tra lớn nhất có thể về mặt thời gian cần thiết để hoàn thành.
Một lời khuyên đơn giản khác là nhân số lượng kiểm tra với 10 khi thêm chiều mới. Điều đó có thể dẫn đến việc ước tính quá cao số lượng
kiểm tra cần thiết, nhưng khá an toàn. Các công cụ được cung cấp được thiết kế để dễ sử dụng, do đó, các giá trị mặc định/tự động "hợp lý" được sử dụng để tối ưu hóa thường có thể được chạy mà không cần chỉ định bất kỳ điều gì (chấp nhận mặc định).

Lưu ý

Điều quan trọng là phải hiểu rằng tất cả các phương pháp tối ưu hóa thông minh đều hoạt động tốt nhất trong không gian tham số liên tục và các hàm mục tiêu tương đối trơn tru. Nếu không gian tham số là rời rạc, các thuật toán tiến hóa có thể gặp

rắc rối khi tìm giá trị tối ưu. Điều này đặc biệt đúng đối với các tham số nhị phân (bật/tắt) - chúng không phù hợp với bất kỳ phương pháp tìm kiếm nào sử dụng gradient của sự thay đổi hàm mục tiêu (như hầu hết các phương pháp thông minh). Nếu hệ thống giao dịch của bạn chứa nhiều tham số nhị phân, bạn không nên sử dụng trình tối ưu thông minh trực tiếp trên chúng. Thay vào đó, hãy thử tối ưu hóa chỉ các tham số liên tục bằng trình tối ưu thông minh và chuyển đổi các tham số nhị phân theo cách thủ công hoặc thông qua tập lệnh bên ngoài.

SPSO - Trình tối ưu hóa bầy hạt chuẩn

Trình tối ưu hóa bầy hạt chuẩn dựa trên mã SPSO2007 được cho là tạo ra kết quả tốt miễn là các tham số chính xác (tức là Chạy, MaxEval) được cung cấp cho một vấn đề cụ thể.
Việc chọn đúng tùy chọn cho trình tối ưu hóa PSO có thể rất khó khăn do đó kết quả có thể khác nhau đáng kể tùy từng trường hợp.
SPSO.dll đi kèm với mã nguồn đầy đủ bên trong thư mục con "ADK". Mã ví dụ cho Trình tối ưu hóa bầy hạt chuẩn:
(tìm giá trị tối ưu trong 1000 lần thử nghiệm trong không gian tìm kiếm gồm 10000 kết hợp)

```
OptimizerSetEngine("spso"); OptimizerSetOption("Chạy", 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 là phiên bản thích ứng, không tham số của PSO (particle swarm optimization) - trình tối ưu hóa không đầy đủ. Để biết thông tin khoa học, hãy xem:
http://www.particleswarm.info/Tribes_2006_Cooren.pdf

Về lý thuyết, nó sẽ hoạt động tốt hơn PSO thông thường, vì nó có thể tự động điều chỉnh kích thước bầy đàn và chiến lược thuật toán theo vấn đề đang được giải quyết.

Thực tế cho thấy hiệu suất của nó khá giống với PSO.

Plugin Tribes.DLL triển khai biến thể "Tribes-D" (tức là không có thứ nguyên). Dựa trên http://clerc.maurice.free.fr/pso/Tribes/TRIBES-D.zip của Maurice Clerc. Mã nguồn gốc được sử dụng với sự cho phép của tác giả
Tribes.DLL đi kèm với mã nguồn đầy đủ (bên trong thư mục "ADK") Các tham số được hỗ trợ:
"MaxEval" - số lượng đánh giá tối đa (kiểm tra ngược) cho mỗi lần chạy (mặc định = 1000).

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

Bạn nên tăng số lượng đánh giá với số chiều tăng dần (số tham số tối ưu hóa).
Mặc định 1000 phù hợp với 2 hoặc tối đa 3 chiều.

"Runs" - số lần chạy (khởi động lại). (mặc định = 5)
Bạn có thể để số lần chạy ở giá trị mặc định là 5. Theo mặc định, số lần chạy (hoặc khởi động lại) được đặt thành 5.

Để sử dụng trình tối ưu hóa Tribes, bạn chỉ cần thêm một dòng vào mã của mình:

```
OptimizerSetEngine("trib");
OptimizerSetOption("MaxEval", 5000 ); // 5000 đánh giá tối đa
```

CMA-ES - Chiến lược tiến hóa thích ứng ma trận hiệp phương sai

CMA-ES (Chiến lược tiến hóa thích ứng ma trận hiệp phương sai) là một công cụ tối ưu hóa tiên tiến không đầy đủ. Để biết thông tin khoa học, hãy xem:
http://www.bionik.tu-berlin.de/user/niko/cmaesintro.html

Theo các tiêu chuẩn khoa học, nó vượt trội hơn chín chiến lược tiến hóa phổ biến nhất khác (như PSO, tiến hóa di truyền và khác biệt).
http://www.bionik.tu-berlin.de/user/niko/cec2005.html

Phần bổ trợ CMAE.DLL triển khai biến thể "Toàn cầu" của tìm kiếm với một số lần khởi động lại với quy mô dân số tăng dần
CMAE.DLL đi kèm với mã nguồn đầy đủ (bên trong thư mục "ADK")

Theo mặc định, số lần chạy (hoặc khởi động lại) được đặt thành 5.
Bạn nên để nguyên số lần khởi động lại mặc định.

Bạn có thể thay đổi nó bằng cách sử dụng lệnh gọi OptimizerSetOption("Runs", N), trong đó N phải nằm trong phạm vi 1..10. Không nên chỉ định nhiều hơn 10 lần chạy, mặc dù có thể.
Lưu ý rằng mỗi lần chạy sử dụng kích thước dân số GẤP ĐÔI so với lần chạy trước đó nên nó tăng theo cấp số nhân. Do đó, với 10 lần chạy, bạn sẽ có dân số lớn hơn 2^10 (1024 lần) so với lần chạy đầu tiên.

Có một tham số khác là "MaxEval". Giá trị mặc định là ZERO, nghĩa là plugin sẽ tự động tính toán MaxEval cần thiết. Bạn KHÔNG nên tự mình định nghĩa MaxEval vì mặc định hoạt động tốt.

Thuật toán đủ thông minh để giảm thiểu số lượng đánh giá cần thiết và hội tụ rất nhanh đến điểm giải pháp, do đó thường tìm ra giải pháp nhanh hơn các chiến lược khác.

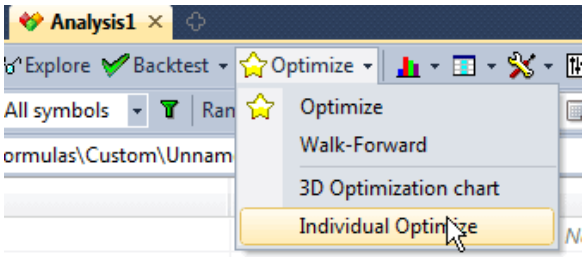
Bình thường, plugin sẽ bỏ qua một số bước đánh giá nếu phát hiện ra giải pháp đã được tìm thấy, do đó bạn không nên ngạc nhiên khi thanh tiến trình tối ưu hóa có thể di chuyển rất nhanh tại một số điểm. Plugin cũng có khả năng tăng số bước so với giá trị ước tính ban đầu nếu cần để tìm ra giải pháp. Do bản chất thích ứng của nó, "thời gian ước tính còn lại" và/hoặc "số bước" được hộp thoại tiến trình hiển thị chỉ là "phỏng đoán tốt nhất tại thời điểm đó" và có thể thay đổi trong quá trình tối ưu hóa.

Để sử dụng trình tối ưu hóa CMA-ES, bạn chỉ cần thêm một dòng vào mã của mình:
OptimizerSetEngine("cmae");
Thao tác này sẽ chạy tối ưu hóa với các thiết lập mặc định phù hợp với hầu hết các trường hợp.

Cần lưu ý, giống như nhiều thuật toán tìm kiếm không gian liên tục, việc giảm tham số "bước" trong các lệnh gọi hàm Optimize() không ảnh hưởng đáng kể đến thời gian tối ưu hóa. Điều duy nhất quan trọng là "chiều" của vấn đề, tức là số lượng các tham số khác nhau (số lần gọi hàm tối ưu hóa). Có thể thiết lập số "bước" cho mỗi tham số mà không ảnh hưởng đến thời gian tối ưu hóa, do đó hãy sử dụng độ phân giải tốt nhất mà bạn muốn. Về lý thuyết, thuật toán sẽ có thể tìm ra giải pháp trong tối đa 900*(N+3)*(N+3) lần kiểm tra ngược trong đó "N" là chiều. Trên thực tế, nó hội tụ NHANH HƠN RẤT NHIỀU. Ví dụ, giải pháp trong không gian tham số 3 (N=3) chiều (ví dụ 100*100*100 = 1 triệu bước toàn diện) có thể được tìm thấy chỉ trong 500-900 bước CMA-ES.

Tối ưu hóa riêng lẻ đa luồng

Bắt đầu từ AmiBroker 5.70 ngoài đa luồng nhiều ký hiệu, bạn có thể thực hiện tối ưu hóa ký hiệu đơn đa luồng. Để truy cập chức năng này, hãy nhấp vào mũi tên thả xuống bên cạnh nút "Tối ưu hóa" trong cửa sổ Phân tích mới và chọn "Tối ưu hóa riêng lẻ".



"Individual Optimize" sẽ sử dụng tất cả các lõi bộ xử lý có sẵn để thực hiện tối ưu hóa ký hiệu đơn, giúp tối ưu hóa nhanh hơn nhiều so với tối ưu hóa thông thường.

Ở chế độ "Ký hiệu hiện tại", chế độ này sẽ thực hiện tối ưu hóa trên một ký hiệu. Ở chế độ "Tất cả ký hiệu" và "Lọc", chế độ này sẽ xử lý tất cả các ký hiệu theo trình tự, tức là tối ưu hóa hoàn chỉnh đầu tiên cho ký hiệu đầu tiên, sau đó tối ưu hóa cho ký hiệu thứ hai, v.v.

Hạn chế:
1. Trình kiểm tra ngược tùy chỉnh KHÔNG được hỗ trợ (chưa)
2. Công cụ tối ưu hóa thông minh KHÔNG được hỗ trợ - chỉ tối ưu hóa TOÀN DIỆN hoạt động.

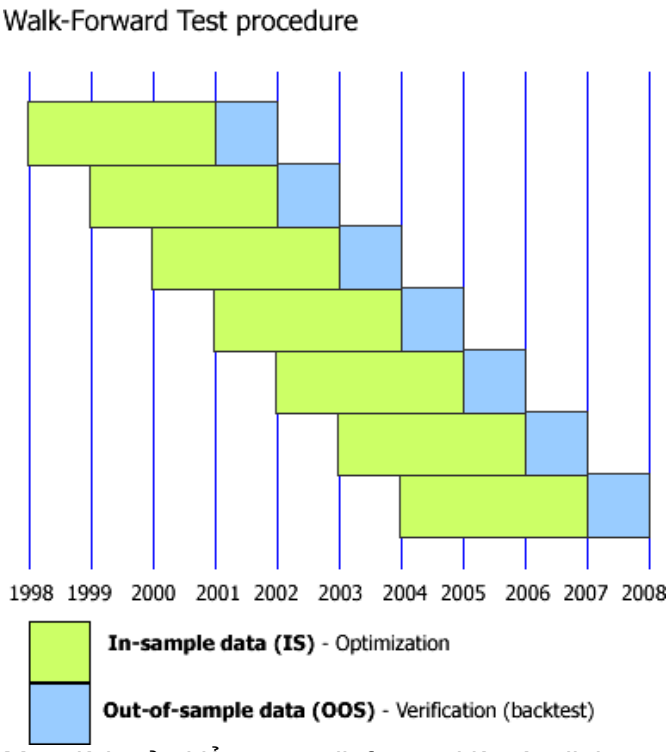
Để biết giải thích về những hạn chế này, hãy xem Hướng dẫn: Sử dụng hiệu quả đa luồng.

Cuối cùng, chúng ta có thể loại bỏ hạn chế (1) - khi AmiBroker được thay đổi để trình kiểm tra ngược tùy chỉnh không sử dụng OLE nữa. Nhưng (2) có thể sẽ tồn tại trong thời gian dài.

Kiểm tra tiến trình

AmiBroker 5.10 có chế độ kiểm tra tiến trình tự động.

Kiểm tra Walk forward tự động là một kỹ thuật thiết kế và xác thực hệ thống trong đó bạn tối ưu hóa các giá trị tham số trên một phân đoạn dữ liệu thị trường trong quá khứ ("trong mẫu"), sau đó xác minh hiệu suất của hệ thống bằng cách kiểm tra nó theo thời gian trên dữ liệu sau phân đoạn tối ưu hóa ("ngoài mẫu"). Bạn đánh giá hệ thống dựa trên mức độ hiệu quả của nó trên dữ liệu thử nghiệm ("ngoài mẫu"), chứ không phải dữ liệu mà nó được tối ưu hóa. Quá trình này có thể được lặp lại trong các phân đoạn thời gian tiếp theo. Hình minh họa sau đây cho thấy cách thức hoạt động của quy trình.



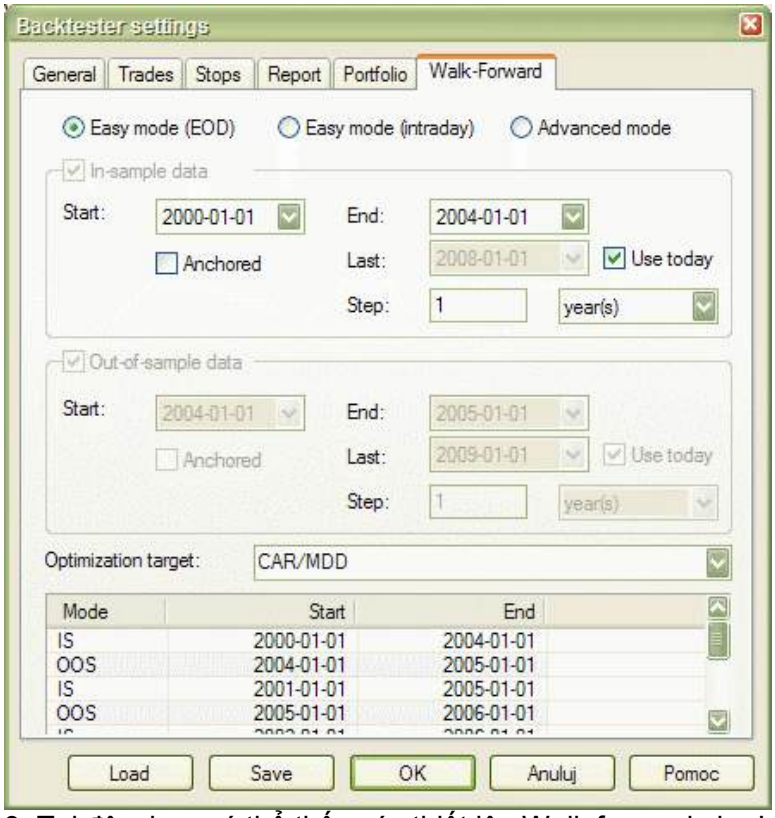
Mục đích của kiểm tra walk forward là xác định xem hiệu suất của hệ thống giao dịch được tối ưu hóa có thực tế hay là kết quả của việc điều chỉnh đường cong. Hiệu suất của hệ thống có thể được coi là thực tế nếu nó có giá trị dự đoán và hoạt động tốt trên dữ liệu thị trường chưa biết (ngoài mẫu). Khi hệ thống được thiết kế đúng cách, hiệu suất giao dịch theo thời gian thực phải liên quan đến hiệu suất được phát hiện trong quá trình tối ưu hóa. Nếu hệ thống hoạt động trong giao dịch thực tế, trước tiên nó phải vượt qua bài kiểm tra walk forward. Nói cách khác, chúng tôi không thực sự quan tâm đến kết quả trong mẫu vì chúng luôn tốt (hoặc nên tốt). Điều quan trọng là hiệu suất của hệ thống ngoài mẫu. Đây là ước tính thực tế về cách hệ thống sẽ hoạt động trong giao dịch thực tế và sẽ nhanh chóng phát hiện ra bất kỳ vấn đề nào về đường cong phù hợp. Nếu hiệu suất ngoài mẫu kém thì bạn không nên giao dịch hệ thống như vậy.

Tiền đề của việc thực hiện nhiều bước tối ưu hóa/kiểm tra theo thời gian là quá khứ gần đây là nền tảng tốt hơn để lựa chọn giá trị tham số hệ thống so với quá khứ xa xôi. Chúng tôi hy vọng rằng các giá trị tham số được chọn trên phân đoạn tối ưu hóa sẽ phù hợp với các điều kiện thị trường ngay sau đó. Điều này có thể đúng hoặc không khi thị trường trải qua chu kỳ tăng/giảm, vì vậy cần cẩn thận khi lựa chọn độ dài của giai đoạn trong mẫu. Để biết thêm thông tin về thiết kế và xác minh hệ thống bằng quy trình chuyển tiếp và tất cả các vấn đề

liên quan, chúng tôi có thể giới thiệu cuốn sách của Howard Bandy: "Hệ thống giao dịch định lượng" (xem các liên kết trên trang AmiBroker).

Để sử dụng tối ưu hóa Walk-Forward, vui lòng làm theo các bước sau:

1. Vào Tools->Automatic Analysis
2. Nhấp vào nút Settings, sau đó chuyển sang tab Walk-Forward



3. Tại đây, bạn có thể thấy các thiết lập Walk forward cho In-sample optimization, out-of-sample backtest

Ngày bắt đầu và ngày kết thúc đánh dấu giai đoạn đầu tiên bắt đầu/kết thúc
Giai đoạn này sẽ được chuyển tiếp theo Bước cho đến khi Kết thúc đạt đến Ngày cuối cùng.

Ngày bắt đầu cũng có thể chuyển tiếp theo từng bước hoặc có thể được neo (hằng số) nếu bật kiểm tra Neo.

Nếu bạn đánh dấu Sử dụng hôm nay thì Ngày cuối cùng đã nhập sẽ bị bỏ qua và HÔM NAY (ngày hiện tại) sẽ được sử dụng thay thế.

Theo mặc định, "CHẾ ĐỘ DỄ DÀNG" được chọn để đơn giản hóa quá trình thiết lập các tham số WF.

Giả định rằng:
a) Phân đoạn ngoài mẫu theo ngay sau phân đoạn trong mẫu

b) độ dài của phân đoạn ngoài mẫu bằng với bước tiến

Dựa trên hai giả định này, chế độ "EASY" lấy ngày KẾT THÚC trong mẫu và đặt ngày BẮT ĐẦU ngoài mẫu thành ngày tiếp theo. Sau đó thêm BƯỚC trong mẫu và đây sẽ trở thành ngày KẾT THÚC ngoài mẫu.

Các giá trị bước trong mẫu và ngoài mẫu được đặt thành cùng một giá trị. Chế độ "EASY" đảm bảo tính chính xác của các thiết lập thủ tục WF.

Bạn nên sử dụng chế độ Dễ (EOD) khi thử nghiệm trên dữ liệu cuối ngày hoặc chế độ Dễ (Trong ngày) khi thử nghiệm trên dữ liệu trong ngày. Sự khác biệt là ở chế độ EOD, ngày KẾT THÚC của kỳ trước và ngày BẮT ĐẦU của kỳ tiếp theo là giống nhau - do đó tránh được khoảng cách
giữa các kỳ. Chế độ trong ngày đặt ngày BẮT ĐẦU của kỳ tiếp theo là NGÀY TIẾP THEO sau KẾT THÚC của kỳ trước. Điều đó đảm bảo
ngày ranh giới không được tính hai lần khi thử nghiệm trên dữ liệu trong ngày.

Ở chế độ Nâng cao, người dùng có toàn quyền kiểm soát tất cả các giá trị, trong phạm vi mà chúng có thể không cấu thành quy trình WF hợp lệ.
Giao diện cho phép vô hiệu hóa có chọn lọc các giai đoạn trong mẫu và ngoài mẫu bằng cách sử dụng hộp kiểm ở trên cùng (đối với những thứ đặc biệt như chạy các thử nghiệm ngược tuần tự mà không cần tối ưu hóa).
Tất cả các cài đặt được phản ánh ngay lập tức trong danh sách XEM TRƯỚC hiển thị tất cả các phân đoạn IS/OOS đã tạo và ngày của chúng.
4. Trường "Mục tiêu tối ưu hóa" xác định TÊN CỘT báo cáo tối ưu hóa
sẽ được sử dụng để sắp xếp kết quả và tìm ra kết quả TỐT NHẤT. Có thể sử dụng bất kỳ cột tích hợp nào (như xuất hiện trong đầu ra tối ưu hóa) hoặc bạn có thể sử dụng bất kỳ số liệu tùy chỉnh nào mà bạn xác định
trong trình kiểm tra ngược tùy chỉnh. Mặc định là CAR/MDD, tuy nhiên bạn có thể chọn bất kỳ số liệu tích hợp nào khác từ tổ hợp.
Bạn cũng có thể NHẬP bất kỳ số liệu tùy chỉnh nào mà bạn đã thêm thông qua giao diện trình kiểm tra ngược tùy chỉnh.
5. Sau khi bạn đã xác định cài đặt Walk-Forward, vui lòng chuyển đến Automatic Analysis và
6. nhấn MŨI TÊN thả xuống trên nút Optimize và chọn “Walk Forward Optimization”

Điều này sẽ chạy trình tự tối ưu hóa và kiểm tra ngược và kết quả sẽ được hiển thị trong tài liệu “Walk Forward” đang mở trong khung ứng dụng chính. Khi tối ưu hóa đang chạy, bạn có thể nhấp vào nút “MINIMIZE” trên hộp thoại Progress để thu nhỏ nó - điều này cho phép xem đầu ra Walk Forward trong các bước tối ưu hóa.

Vốn chủ sở hữu kết hợp IN-SAMPLE và OUT-OF-SAMPLE

Cổ phiếu kết hợp in-sample và out-sample có sẵn theo mã chứng khoán tổng hợp ~~~ISEQUITY và ~~~OSEQUITY (các giai đoạn liên tiếp của IS và OOS được nối lại và chia tỷ lệ để duy trì tính liên tục của dòng vốn chủ sở hữu - cách tiếp cận này giả định rằng nói chung bạn đang gộp lợi nhuận).

Để hiển thị vốn chủ sở hữu IS và OOS, bạn có thể sử dụng ví dụ sau:

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

Báo cáo tóm tắt OUT-OF-SAMPLE (mới trong 5.60)
Phiên bản 5.60 mang đến báo cáo tóm tắt walk-forward mới bao gồm tất cả các bước ngoài mẫu. Báo cáo này hiển thị trong Report Explorer dưới dạng bước cuối cùng và có kiểu "PS". Đã có những thay đổi đáng kể đối với thử nghiệm walk-forward được thực hiện để cho phép báo cáo tóm tắt out-of-mẫu. Thay đổi quan trọng nhất là mỗi thử nghiệm out-of-mẫu tiếp theo sử dụng vốn chủ sở hữu ban đầu bằng với vốn chủ sở hữu kết thúc bước trước đó. (Trước đây, nó sử dụng vốn chủ sở hữu ban đầu không đổi). Thay đổi này là bắt buộc để tính toán đúng tất cả các số liệu thống kê/số liệu trong tất cả các phần của thử nghiệm out-of-mẫu.

Báo cáo tóm tắt cho thấy lưu ý rằng các số liệu tích hợp thể hiện chính xác tất cả các bước ngoài mẫu nhưng các số liệu tùy chỉnh tóm tắt được tạo thành bằng phương pháp do người dùng xác định:
1 giá trị bước đầu tiên, 2 giá trị bước cuối cùng, 3 tổng, 4 trung bình, 5 tối thiểu, 6 tối đa.

Theo mặc định, báo cáo tóm tắt hiển thị giá trị bước cuối cùng của số liệu tùy chỉnh TRỪ KHI người dùng chỉ định phương pháp kết hợp khác trong lệnh gọi bo.AddCustomMetrics().

bo.AddCustomMetrics hiện có tham số tùy chọn mới - CombineMethod

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

Phương pháp này thêm số liệu tùy chỉnh vào báo cáo kiểm thử ngược, "tóm tắt" kiểm thử ngược và danh sách kết quả tối ưu hóa. Title là tên của số liệu sẽ được hiển thị trong báo cáo, Value là giá trị của số liệu, các đối số tùy chọn LongOnlyValue, ShortOnlyValue cho phép cung cấp giá trị cho các cột dài/chỉ ngắn bổ sung trong báo cáo kiểm thử ngược. Đối số cuối cùng DecPlaces kiểm soát số lượng chữ số thập phân sẽ được sử dụng để hiển thị giá trị.

Các giá trị CombineMethod được hỗ trợ là:
1 giá trị bước đầu tiên, - báo cáo tóm tắt sẽ hiển thị giá trị của số liệu tùy chỉnh từ bước đầu tiên ngoài mẫu
2 giá trị bước cuối cùng (mặc định), - báo cáo tóm tắt sẽ hiển thị giá trị của số liệu tùy chỉnh từ bước cuối cùng ngoài mẫu
3 tổng, - báo cáo tóm tắt sẽ hiển thị tổng các giá trị của số liệu tùy chỉnh từ tất cả các bước ngoài mẫu
4 trung bình, - báo cáo tóm tắt sẽ hiển thị giá trị trung bình của các giá trị của số liệu tùy chỉnh từ tất cả các bước ngoài mẫu
5 tối thiểu, - báo cáo tóm tắt sẽ hiển thị giá trị nhỏ nhất của số liệu tùy chỉnh từ tất cả các bước ngoài mẫu
6 tối đa. - báo cáo tóm tắt sẽ hiển thị giá trị lớn nhất của số liệu tùy chỉnh từ tất cả các bước ngoài mẫu

Lưu ý rằng một số phương pháp tính toán số liệu phức tạp và ví dụ, việc tính trung bình chúng sẽ không dẫn đến biểu diễn chính xác về mặt toán học của tất cả các thử nghiệm ngoài mẫu. Tóm tắt của tất cả các số liệu tích hợp là chính xác về mặt toán học ngay khi xuất xưởng (tức là chúng *không* phải là giá trị trung bình, mà là các số liệu được tính toán chính xác bằng phương pháp phù hợp với giá trị đã cho). Điều này trái ngược với các số liệu tùy chỉnh, vì chúng
có thể do người dùng xác định và tùy thuộc vào người dùng để chọn phương pháp 'kết hợp', và vẫn có thể xảy ra trường hợp không có phương pháp nào phù hợp.
Vì lý do đó, báo cáo bao gồm ghi chú giải thích phương pháp do người dùng xác định nào đã được sử dụng để kết hợp các số liệu tùy chỉnh.

Hệ thống kiểm tra ngược cho hợp đồng tương lai

Giới thiệu

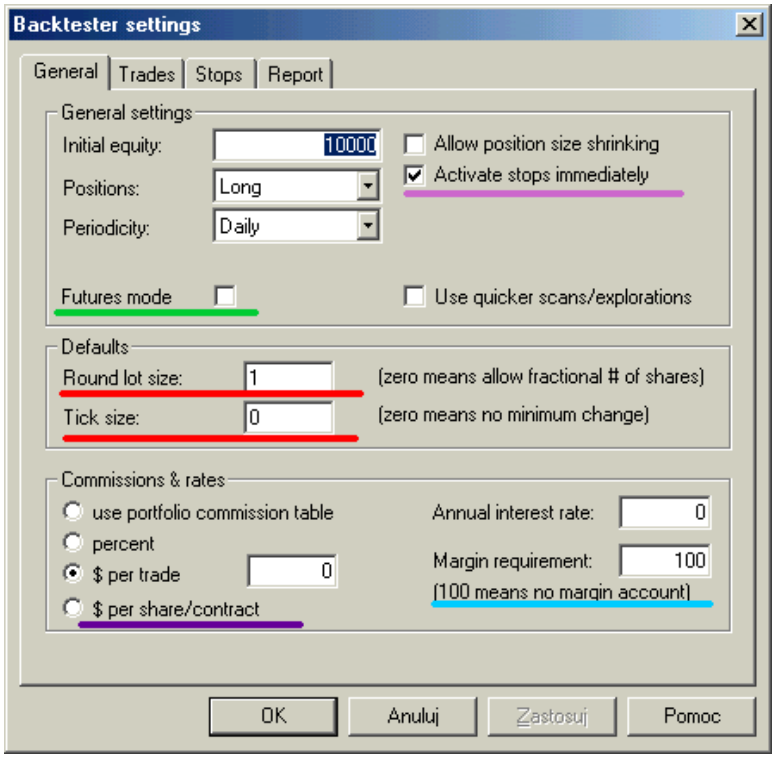
Trước khi đọc bài viết này, trước tiên bạn nên đọc phần "Kiểm tra ngược ý tưởng giao dịch của bạn" vì nó cung cấp bối cảnh cần thiết về kiểm tra ngược nói chung.

Khi bạn mở vị thế mua cổ phiếu, bạn chỉ cần mua một số lượng cổ phiếu nhất định ở mức giá nhất định, sau một thời gian, bạn bán chúng và lợi nhuận của bạn được tính bằng chênh lệch giữa giá bán và giá mua nhân với số lượng cổ phiếu. Nếu bạn muốn mở vị thế mua hợp đồng tương lai, bạn phải trả tiền ký quỹ - ký quỹ - cho mỗi hợp đồng. Ký quỹ chỉ là một phần nhỏ của toàn bộ giá trị hợp đồng (ví dụ: 10%). Vì vậy, bạn có thể mua 10 hợp đồng mà không phải trả quá giá trị đầy đủ của một hợp đồng. Điều này mang lại cho bạn đòn bẩy khiến giao dịch tương lai rủi ro hơn giao dịch cổ phiếu. Khi giá hợp đồng thay đổi, lợi nhuận/tổn thất của bạn cũng thay đổi theo. Nếu giá trị điểm của hợp đồng là 1 thì mỗi 1$ thay đổi trong giá hợp đồng tương ứng với lợi nhuận/tổn thất 1$ cho mỗi hợp đồng - giống như trong cổ phiếu. Nhưng hợp đồng tương lai có thể có giá trị điểm khác với 1. Ví dụ, nếu giá trị điểm là 5 thì mỗi 1 điểm thay đổi trong giá hợp đồng tương ứng với lợi nhuận/tổn thất 5$ trong vốn chủ sở hữu của bạn. Khi bạn đóng vị thế, bạn sẽ nhận lại được tiền ký quỹ, do đó lợi nhuận/tổn thất của bạn được tính bằng số lượng hợp đồng nhân với giá trị điểm nhân với chênh lệch giữa giá bán và giá mua.

Chế độ tương lai của backtester

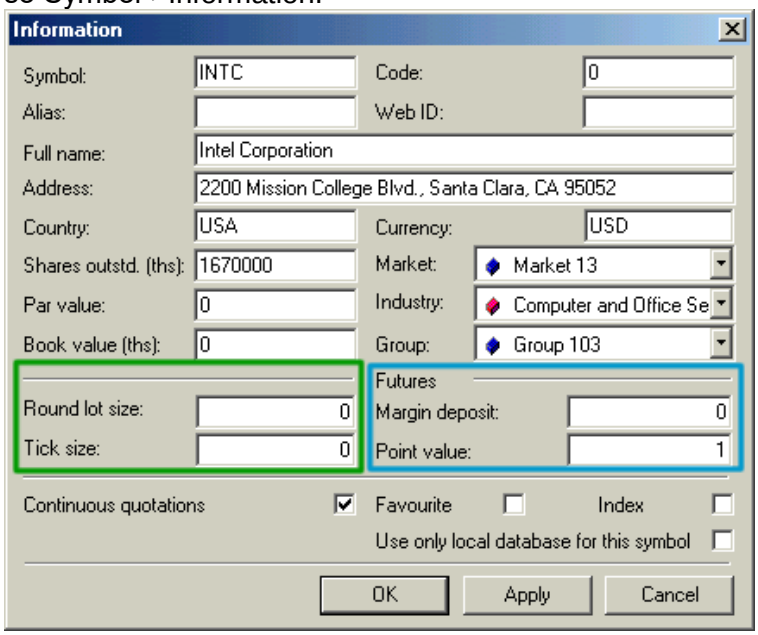
Có 3 cài đặt chỉ dành cho hợp đồng tương lai trong backtester:

• Hộp kiểm chế độ tương lai (Trang Cài đặt-Chung)
• Tiền ký quỹ (Trang Biểu tượng-Thông tin)
• Giá trị điểm (Trang Biểu tượng-Thông tin)



Hộp kiểm chế độ tương lai trong trang cài đặt (được gạch chân bằng đường màu xanh lá cây trong hình trên) là chìa khóa để

backtester hợp đồng tương lai. Nó hướng dẫn backtester sử dụng tiền gửi ký quỹ và giá trị điểm trong các phép tính. Các thiết lập còn lại là theo từng ký hiệu và có thể truy cập từ cửa sổ Symbol->Information.



Tiền ký quỹ

Ký quỹ là số tiền cần thiết để mở một vị thế hợp đồng duy nhất. Bạn có thể chỉ định ký quỹ theo ký hiệu trong trang Thông tin ký hiệu (hình ảnh bên trên). Các giá trị dương mô tả giá trị ký quỹ tính bằng đô la, trong khi giá trị âm thể hiện giá trị ký quỹ dưới dạng phần trăm giá hợp đồng. Giá trị ký quỹ bằng 0 được sử dụng cho cổ phiếu (không có ký quỹ). Ký quỹ cũng có thể được chỉ định trong công thức bằng cách sử dụng biến dự trữ MarginDeposit:

MarginDeposit = 675;

Trong chế độ Hợp đồng tương lai, thiết lập ký quỹ được sử dụng để xác định số lượng hợp đồng có thể mua. Giả sử vốn chủ sở hữu ban đầu của bạn được đặt thành 50.000 đô la và bạn muốn đầu tư tới 20% vốn chủ sở hữu vào một giao dịch duy nhất và tiền ký quỹ là 675 đô la. Trong trường hợp đó, quy mô vị thế "mong muốn" của bạn là 50.000 * 0,2 = 10.000. Với điều kiện là bạn đã đặt kích thước lô tròn thành 1, backtester sẽ "mua" 10000/675 = (số nguyên)14.8148 = 14 hợp đồng và giá trị vị thế thực sẽ là 9450 đô la (18,9% vốn chủ sở hữu ban đầu).

Để mô phỏng điều này trong AmiBroker, bạn sẽ cần nhập 50000 vào trường Vốn chủ sở hữu ban đầu trong backtester, bật chế độ tương lai và thiết lập các tham số còn lại trong công thức của bạn:

PositionSize = -20; // sử dụng 20% vốn chủ sở hữu MarginDeposit = 675; // bạn cũng có thể đặt giá trị này trong trang Thông tin ký hiệu
RoundLotSize = 1; // bạn cũng có thể đặt giá trị này trong trang Cài đặt

Tất cả các giao dịch tiếp theo sẽ sử dụng cùng một logic nhưng vị thế sẽ được định cỡ theo vốn chủ sở hữu tích lũy hiện tại thay vì mức vốn chủ sở hữu ban đầu, trừ khi bạn chỉ định kích thước vị thế cố định trong công thức của mình (ví dụ: PositionSize = 10000).

Giá trị điểm

Giá trị điểm là thiết lập theo ký hiệu (có thể định nghĩa trong cửa sổ Thông tin ký hiệu - (hình ảnh ở trên)) xác định số tiền lợi nhuận tạo ra từ một hợp đồng khi giá tăng một điểm. Ví dụ: đồng được báo giá theo xu cho một pound, giá báo giá là 84,65 (hoặc 8465) bằng 84 xu và 65/100 xu cho một pound. Thay đổi +.37 hoặc 37 biểu thị 37/100 xu, thông thường bạn sẽ nghe thấy giá báo là 37 điểm. Nhưng vì giá trị điểm của đồng là 2,5 nên mỗi lần thay đổi điểm sẽ cho lãi/lỗ 2,5 đô la, vì vậy trong ví dụ này, lãi/lỗ trong ngày sẽ là 2,5 * 37 = 92,50 đô la.

Bạn cũng có thể thiết lập giá từ cấp công thức bằng cách sử dụng biến được đặt trước PointValue, ví dụ:

```
PointValue = 2,5;
```

Lưu ý: Khi bạn tải cơ sở dữ liệu cũ AmiBroker đặt trước trường giá trị điểm thành 1 và giả định rằng theo mặc định, 1 điểm biểu thị một đô la nên một đô la thay đổi sẽ tạo ra một đô la lãi/lỗ. Điều này được thực hiện để đảm bảo rằng bạn nhận được kết quả chính xác ngay cả khi bạn (vô tình) chạy thử nghiệm chế độ tương lai trên cổ phiếu.

Lưu ý 2: Mặc dù thiết lập giá trị điểm ảnh hưởng (nhân) lợi nhuận/lỗ nhưng KHÔNG ảnh hưởng đến các điểm dừng tích hợp. Các điểm dừng LUÔN hoạt động chỉ dựa trên biến động giá. Vì vậy, bạn nên biết rằng việc thiết lập điểm dừng mục tiêu lợi nhuận 10% sẽ tạo ra lợi nhuận 25% cho giao dịch thoát khỏi điểm dừng này khi giá trị điểm được đặt thành 2,5.

Các trường hợp đơn giản

Kiểm tra chỉ điểm

Kiểm tra chỉ điểm tương đương với việc chỉ giao dịch một hợp đồng. Điều này có thể dễ dàng thực hiện bằng cách sử dụng chế độ Tương lai của trình kiểm tra ngược và thêm một dòng sau vào công thức của bạn:

```
PositionSize = MarginDeposit = 1;
```

Giao dịch 'n' hợp đồng

Theo cách tương tự, bạn có thể thiết lập công thức của mình để nó luôn giao dịch 7 hợp đồng. Tất cả những gì bạn cần làm là thêm phần sau vào công thức của mình:

```
NumContracts = 7;
PositionSize = NumContracts * MarginDeposit;
```

Mô phỏng Monte Carlo cho hệ thống giao dịch của bạn

LƯU Ý: Chủ đề nâng cao. Đảm bảo đọc các phần trước của hướng dẫn trước. Để diễn giải đúng kết quả mô phỏng Monte Carlo, bạn cần đọc phần này của hướng dẫn. Các thiết lập không tầm thường và
các chi tiết không rõ ràng được giải thích bên dưới. Vui lòng không bỏ qua.

Giới thiệu

Nói chung, các phương pháp "Monte Carlo" đại diện cho lớp thuật toán máy tính rộng sử dụng lấy mẫu ngẫu nhiên lặp lại để thu được các đặc tính thống kê của quy trình nhất định. Phương pháp này được phát minh bởi nhà toán học người Ba Lan Stanislaw Ulam làm việc cho các dự án vũ khí hạt nhân tại phòng thí nghiệm Los Alamos. Vì không thể phân tích các quy trình vật lý phức tạp bằng các phương pháp toán học thông thường, ông nghĩ rằng mình có thể thiết lập một loạt các thí nghiệm ngẫu nhiên, quan sát kết quả và sử dụng chúng để suy ra các đặc tính thống kê của quy trình.

Có thể tìm hiểu thêm về phương pháp Monte Carlo nói chung tại đây:
https://en.wikipedia.org/wiki/Monte_Carlo_method

Trong quá trình phát triển hệ thống giao dịch, mô phỏng Monte Carlo đề cập đến quá trình sử dụng các chuỗi giao dịch mô phỏng ngẫu nhiên để đánh giá các thuộc tính thống kê của hệ thống giao dịch.

Có nhiều cách để thực hiện các phép tính thực tế khác nhau khi nói đến các chi tiết triển khai, nhưng có lẽ cách đơn giản và đáng tin cậy nhất là phương pháp khởi động thực hiện lấy mẫu ngẫu nhiên với việc thay thế danh sách giao dịch thực tế được tạo ra bởi kiểm tra ngược.

Xem https://en.wikipedia.org/wiki/Bootstrapping_(statistics) để biết thảo luận chi tiết về phương pháp khởi động.

Nhiều phương pháp mô phỏng Monte Carlo khác nhau cho phép xác minh tính mạnh mẽ của hệ thống giao dịch, tìm ra xác suất phá sản và con người

AmiBroker hoạt động như thế nào?

Để thực hiện mô phỏng Monte Carlo (hoặc thử nghiệm bootstrap) cho hệ thống giao dịch của bạn, AmiBroker thực hiện như sau:

A. Tạo bộ đầu vào

A.1 Thực hiện kiểm tra ngược hệ thống giao dịch của bạn để tạo ra bộ N giao dịch ban đầu

B. Lặp lại (hơn 1000 lần)

B.1 chọn ngẫu nhiên các giao dịch từ danh sách giao dịch ban đầu để tạo ra bộ N giao dịch mới, ngẫu nhiên (gọi là 'thực hiện')

Bộ ngẫu nhiên này chứa cùng số lượng giao dịch, chúng được sắp xếp ngẫu nhiên và một số giao dịch ban đầu có thể bị bỏ qua và một số được sử dụng nhiều hơn một lần (hoán vị với sự lặp lại hoặc lấy mẫu ngẫu nhiên với sự thay thế).

Vì số lượng thực hiện duy nhất là N^N (do đó chỉ với 100 giao dịch đầu vào, chúng ta có 100100 thực hiện duy nhất), với số lượng giao dịch đủ lớn (>100), khả năng chọn chuỗi giống hệt như ban đầu hầu như bằng không.

B.2 tuần tự thực hiện tính toán lãi/lỗ cho mỗi giao dịch được chọn ngẫu nhiên, sử dụng quy mô vị thế do người dùng xác định để tạo vốn chủ sở hữu hệ thống

B.3 ghi lại vốn chủ sở hữu hệ thống trong phân phối

C. Hậu xử lý

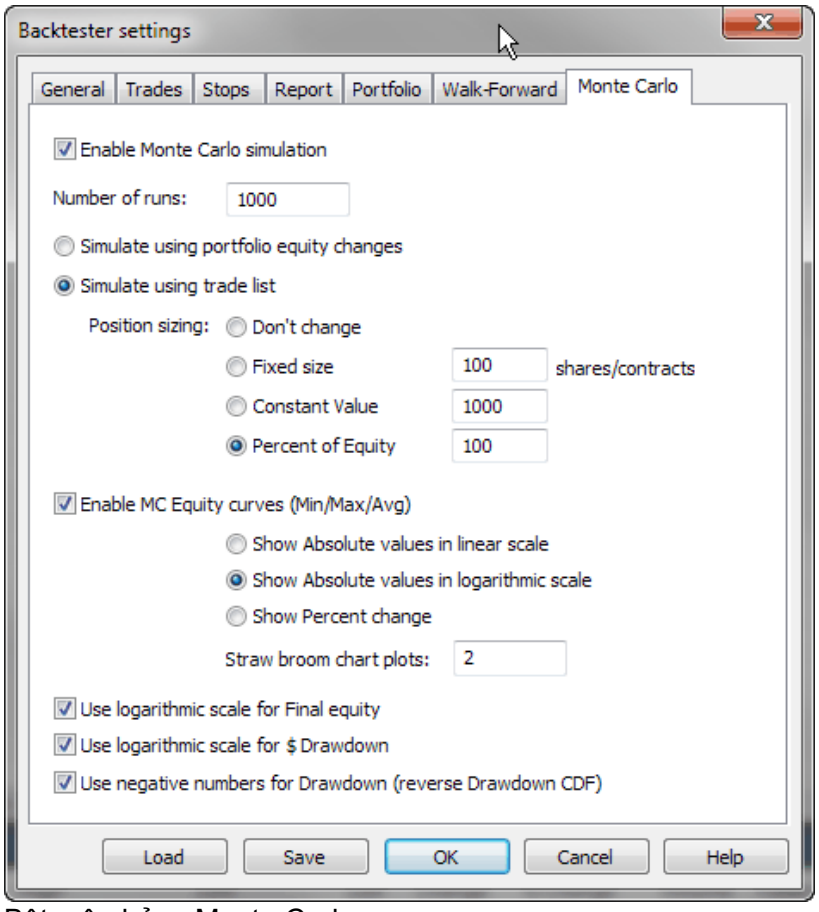
C.1 Xử lý dữ liệu thu được trong B để tạo số liệu thống kê và biểu đồ phân phối

Tất cả những điều trên xảy ra khi bạn nhấn nút Kiểm tra ngược trong cửa sổ Phân tích mới. Trình mô phỏng Monte Carlo của AmiBroker nhanh đến mức thường chỉ mất một phần nhỏ giây so với quy trình kiểm tra ngược thông thường.

Cần lưu ý rằng các giao dịch mô phỏng trong quá trình khởi động được thực hiện tuần tự. Nếu hệ thống giao dịch ban đầu của bạn giao dịch nhiều vị thế cùng một lúc (do đó một số hoặc tất cả các giao dịch bị chồng chéo), thì có thể dẫn đến việc hệ thống rút tiền ít hơn được báo cáo bằng thử nghiệm khởi động, vì việc rút tiền từ các giao dịch riêng lẻ sẽ xảy ra tuần tự (không song song như các giao dịch chồng chéo).

Cài đặt

Cách thức hoạt động của trình mô phỏng Monte Carlo có thể được kiểm soát từ trang Cài đặt phân tích, tab "Monte Carlo":



Bật mô phỏng Monte Carlo

hộp kiểm này kiểm soát bất kỳ khi nào mô phỏng MC được thực hiện tự động như một phần của thử nghiệm ngược (ngay sau khi thử nghiệm ngược tạo danh sách giao dịch)

Số lần chạy

xác định số lần mô phỏng MC cần chạy (phải là 1000 hoặc nhiều hơn)

Mô phỏng bằng cách sử dụng thay đổi vốn chủ sở hữu danh mục đầu tư

tùy chọn này khiến mô phỏng MC sử dụng phần trăm thay đổi vốn chủ sở hữu danh mục đầu tư theo từng thanh thay vì các giao dịch riêng lẻ. Những thay đổi vốn chủ sở hữu riêng lẻ đó được chọn ngẫu nhiên và hoán đổi để tạo ra lần chạy mô phỏng. Trong chế độ này, thay đổi vốn chủ sở hữu theo từng thanh được tính theo tỷ lệ (do đó, mức tăng 10% được biểu thị là 1,1), được chọn ngẫu nhiên và nhân tích lũy. Cài đặt này cho phép xử lý các tình huống khi bạn có nhiều giao dịch chồng chéo trong hệ thống của mình và không yêu cầu bất kỳ cài đặt đặc biệt nào để định cỡ vị thế.

Mô phỏng bằng cách sử dụng danh sách giao dịch

tùy chọn này khiến mô phỏng MC sử dụng các giao dịch riêng lẻ từ thử nghiệm ngược ban đầu để tạo lần chạy mô phỏng. Để thực hiện mô phỏng ở chế độ này, trình mô phỏng MC sẽ chọn ngẫu nhiên các giao dịch gốc và áp dụng quy mô vị thế mới như được định nghĩa bên dưới. Chế độ này hữu ích trong trường hợp bạn không có các giao dịch chồng chéo.

Quy mô vị thế

xác định phương pháp quy mô vị thế được trình mô phỏng MC sử dụng ở chế độ "danh sách giao dịch":

Không thay đổi - sử dụng quy mô vị thế gốc như đã sử dụng trong quá trình kiểm tra ngược. Hãy nhớ rằng nó luôn sử dụng giá trị đô la gốc của giao dịch (hoặc bất kỳ loại tiền tệ nào bạn sử dụng), ngay cả khi công thức của bạn đang sử dụng phần trăm vốn chủ sở hữu danh mục đầu tư.

Quy mô cố định - sử dụng số lượng cổ phiếu/hợp đồng cố định cho mỗi giao dịch

Giá trị hằng số - sử dụng số tiền đô la cố định để mở bất kỳ giao dịch nào

Phần trăm vốn chủ sở hữu - sử dụng phần trăm xác định của giá trị vốn chủ sở hữu mô phỏng hiện tại. Hãy cẩn thận khi sử dụng cài đặt này - nó khiến quy mô vị thế của một giao dịch phụ thuộc vào lợi nhuận của các giao dịch trước đó (lợi nhuận gộp) và tạo ra sự phụ thuộc tuần tự. Nó cũng có thể dẫn đến hiệu ứng gộp bổ sung khi bạn có các giao dịch chồng chéo trong quá trình kiểm tra ngược ban đầu của mình vì bootstrap thực hiện các giao dịch theo trình tự (do đó chúng không chồng chéo). Vì lý do này, việc sử dụng nó bị giới hạn trong các trường hợp không xảy ra giao dịch chồng chéo.

Kích hoạt đường cong vốn chủ sở hữu MC (Tối thiểu/Tối đa/Trung bình)

bật biểu đồ vốn chủ sở hữu MC (bao gồm các biểu đồ vốn chủ sở hữu cao nhất, thấp nhất và trung bình cùng với biểu đồ vốn chủ sở hữu chỗi rơm).
Lưu ý rằng các đường màu xanh lá cây và màu đỏ (vốn chủ sở hữu tối thiểu/tối đa) không thực sự là các vốn chủ sở hữu "tốt nhất" và "tệ nhất" duy nhất. Chúng là các điểm cao nhất (tối đa) và thấp nhất (tối thiểu) từng thanh của TẤT CẢ các vốn chủ sở hữu được tạo ra trong MC.
Vì vậy, chúng thực sự là các điểm tốt nhất từ tất cả các vốn chủ sở hữu và các điểm tệ nhất từ tất cả các vốn chủ sở hữu. Và đường màu xanh lam (trung bình) là trung bình từ tất cả các đường vốn chủ sở hữu (tất cả các lần chạy).

Hiển thị giá trị tuyệt đối theo thang tuyến tính - hiển thị cổ phiếu theo giá trị đô la tuyệt đối bằng biểu đồ thang tuyến tính

Hiển thị giá trị tuyệt đối theo thang logarit - hiển thị cổ phiếu theo giá trị đô la tuyệt đối bằng biểu đồ bán logarit

Hiển thị phần trăm thay đổi - hiển thị cổ phiếu theo "tỷ lệ thay đổi" kể từ đầu

Biểu đồ chổi rơm - xác định số lượng cổ phiếu thử nghiệm riêng lẻ nên được biểu thị dưới dạng 'biểu đồ chổi rơm' (số lượng lớn có thể làm chậm quá trình xử lý/vẽ)

Sử dụng thang logarit cho Cổ phiếu cuối cùng

Hiển thị biểu đồ CDF vốn chủ sở hữu cuối cùng bằng thang bán logarit thay vì tuyến tính

Sử dụng thang logarit cho $Drawdown

Hiển thị biểu đồ CDF rút tiền đô la bằng thang bán logarit thay vì tuyến tính

Sử dụng số âm cho Drawdown (CDF rút tiền ngược)

Khi tùy chọn này được bật, cả rút tiền đô la và phần trăm đều được báo cáo là số âm. Điều này cũng ảnh hưởng đến phân phối CDF. Nó đảo ngược thứ tự của cột "rút tiền" trong bảng MC và đảo ngược ý nghĩa (tức là giá trị phần trăm 10% có nghĩa là có 10% khả năng rút tiền bằng hoặc tệ hơn (tiêu cực hơn) so với số tiền được trình bày. Khi tắt tùy chọn này (như trong các phiên bản cũ), các khoản rút tiền được báo cáo là các số lớn hơn không (dương) và giá trị phần trăm 10% có nghĩa là có 10% khả năng rút tiền bằng hoặc tốt hơn (nhỏ hơn) so với số tiền được trình bày.

Thực hành tốt nhất

Để loại bỏ rủi ro tương quan nối tiếp ảnh hưởng đến kết quả mô phỏng Monte Carlo, chúng tôi khuyến khích sử dụng quy mô vị thế cố định (giá trị đô la cố định của các giao dịch hoặc số lượng cổ phiếu/hợp đồng cố định), do đó thứ tự giao dịch nhất định xảy ra theo trình tự ban đầu không ảnh hưởng đến lợi nhuận/tổn thất do tính gộp.

Ngoài ra, tùy thuộc vào thời điểm hệ thống của bạn mở nhiều vị thế chồng chéo, hãy chọn phương pháp mô phỏng như sau
• Mô phỏng bằng danh sách giao dịch - đối với các hệ thống có giao dịch không chồng chéo hoặc
• Mô phỏng bằng cách sử dụng thay đổi vốn chủ sở hữu danh mục đầu tư: đối với các hệ thống có giao dịch chồng chéo (các vị trí đồng thời)

Diễn giải kết quả

Kết quả mô phỏng Monte Carlo được hiển thị trên trang "Monte Carlo" của báo cáo Backtest.

Ở đầu trang, chúng ta có thể thấy một bảng cung cấp giá trị của một số số liệu thống kê chính có được từ biểu đồ phân phối tích lũy (CDF) của kết quả mô phỏng Monte Carlo.

Sau đây là các kết quả mẫu (các điểm nổi bật được thêm thủ công để minh họa). Vốn chủ sở hữu ban đầu là 10000 trong ví dụ này. Thử nghiệm được thực hiện trong 7 năm (dữ liệu EOD).

Phương trình	Tỷ lệ phần trăm	Lợi nhuận hàng năm của vốn chủ sở hữu cuối cùng	Giảm tối đa $	Giảm tối đa %	Thấp nhất
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

Cột đầu tiên hiển thị mức phần trăm (giá trị mà phần trăm quan sát thử nghiệm (thực tế) nhất định nằm bên dưới). Vì vậy, hãy nói rằng phần trăm thứ 10 cho chúng ta biết rằng 10% thời gian giá trị quan sát được thấp hơn số tiền hiển thị. Ví dụ, giá trị lợi nhuận hàng năm ở phần trăm thứ 10 (trong trường hợp này là -0,41%) có nghĩa là 10% thử nghiệm (thực tế) có lợi nhuận hàng năm ít hơn hoặc bằng mức hiển thị (-0,41%). Vì vậy, chúng ta có thể nói rằng có khoảng 10% khả năng hệ thống của chúng ta sẽ không kiếm được tiền (sẽ không hòa vốn). Con số rút tiền tối đa ở phần trăm thứ 90 (38,48%) có nghĩa là trong 90% trường hợp, mức rút tiền sẽ thấp hơn 38,48%. Nói cách khác, chúng ta có thể nói rằng có 10% khả năng nó sẽ cao hơn thế. Nếu chúng ta xem xét kỹ hơn trong bảng, chúng ta cũng có thể nhận thấy rằng trong 99% trường hợp, mức giảm sẽ thấp hơn 63,82%. Điều quan trọng cần lưu ý là bảng trên được tạo ra với tùy chọn "Sử dụng số âm để giảm" được TẮT.

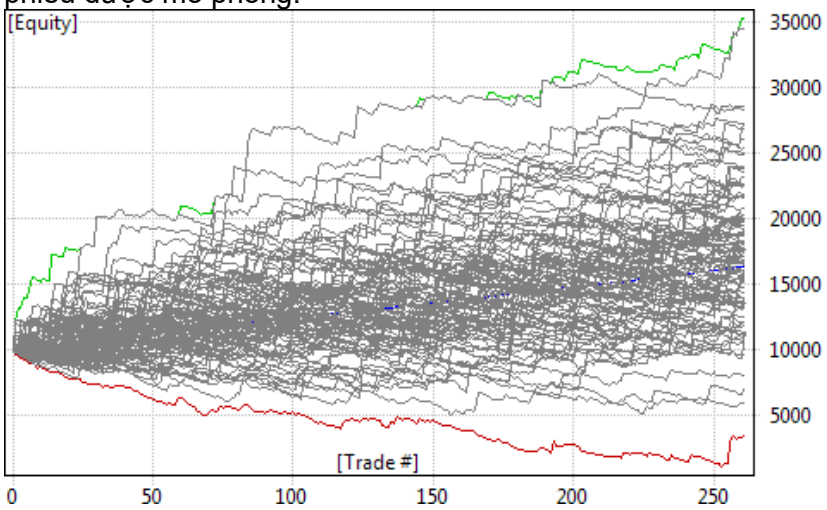
Nếu chúng ta BẬT tùy chọn "Sử dụng số âm để giảm", tất cả các số giảm sẽ trở thành số âm và thứ tự sẽ bị đảo ngược và ý nghĩa của cột giảm cũng sẽ bị đảo ngược, như trong bảng bên dưới:

Phương trình	Tỷ lệ phần trăm	Lợi nhuận hàng năm của vốn chủ sở hữu cuối cùng	Giảm tối đa $	Giảm tối đa %	Thấp nhất
1%	5706	-7.37%	-7685	**-63.82%**	3618
5%	7987	-3.02%	-5292	-45.47%	5853
10%	9706	**-0.41%**	-4626	**-38.48%**	6690
25%	12851	3.48%	-3563	-27.63%	8107
50%	16174	6.78%	-2747	-19.77%	9135
75%	19632	9.64%	-2136	-14.38%	9640
90%	23258	12.21%	-1726	-11.32%	9922
95%	25269	13.48%	-1549	-9.76%	10000
99%	29139	15.71%	-1302	-7.23%	10000

Lần này, ý nghĩa của cột drawdown bị đảo ngược - nó cho bạn biết rằng drawdown sẽ tệ hơn (tiêu cực hơn) so với số tiền đã chỉ định, giá trị phần trăm 99% là -7,23% có nghĩa là trong 99% trường hợp, bạn sẽ thấy drawdown tệ hơn (tiêu cực hơn) so với -7,23%. Giá trị phần trăm 1% là -63,82% cho bạn biết rằng trong 1% trường hợp, bạn sẽ thấy drawdown bằng hoặc tệ hơn (tiêu cực hơn) so với -63,82%

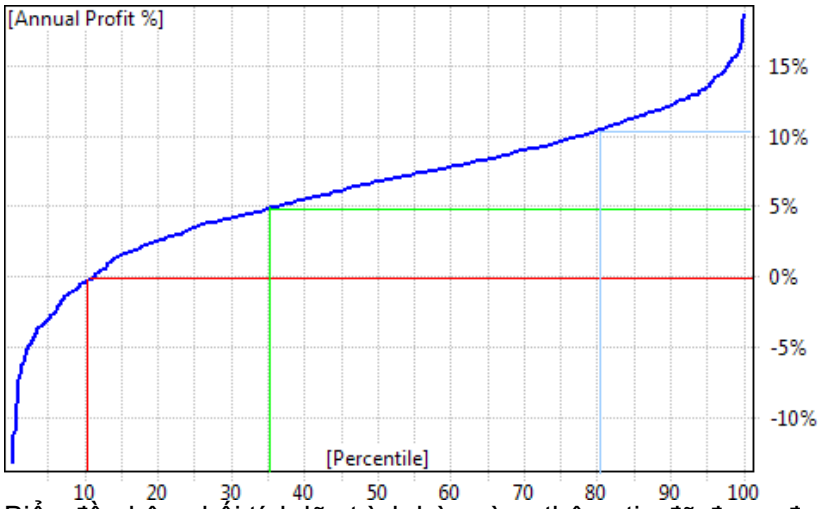
Theo cách này, bảng có thể được đọc "theo hàng" và phần đầu của bảng (phần trăm nhỏ) đề cập đến các kịch bản "bi quan".

Bên dưới bảng, chúng ta có thể tìm thấy biểu đồ min/avg/max + straw broom của các cổ phiếu được mô phỏng:



Lưu ý rằng các đường màu xanh lá cây và màu đỏ (cổ phiếu tối thiểu/tối đa) không thực sự là cổ phiếu "tốt nhất" và "tệ nhất" duy nhất. Chúng là các điểm cao nhất (tối đa) và thấp nhất (tối thiểu) theo từng thanh của TẤT CẢ các cổ phiếu được tạo ra trong MC. Vì vậy, chúng thực sự là các điểm tốt nhất từ tất cả các cổ phiếu và các điểm tệ nhất từ tất cả các cổ phiếu. Và đường màu xanh lam (trung bình) là mức trung bình từ tất cả các đường cổ phiếu (tất cả các lần chạy). 'Đám mây' các đường màu xám biểu thị các cổ phiếu thử nghiệm riêng lẻ - như chúng ta có thể thấy, cùng một hệ thống giao dịch có thể tạo ra các kết quả khác nhau khi điều kiện thị trường thay đổi và mô phỏng MC cố gắng mô phỏng các kết quả khác nhau và cung cấp cho bạn một số thông tin thống kê về mức độ tệ/tốt của nó.

Sau biểu đồ chổi rơm, bạn có thể tìm thấy biểu đồ hàm phân phối tích lũy (CDF) của cổ phiếu cuối cùng, CAR, mức giảm và cổ phiếu thấp nhất (một lần nữa các đường chú thích màu xanh lá cây và màu đỏ được thêm thủ công):



Biểu đồ phân phối tích lũy trình bày cùng thông tin đã được đưa vào bảng ở đầu trang "Monte Carlo" nhưng ở dạng đồ họa. Một lần nữa, khi chúng ta xem biểu đồ phân phối lợi

nhuận hàng năm % (CAR), chúng ta có thể thấy rằng trong khoảng 10% trường hợp, hệ thống của chúng ta sẽ không hòa vốn (tạo ra CAR âm). Chúng ta cũng có thể thấy rằng trong khoảng 35% trường hợp, CAR của chúng ta sẽ dưới 5%. Lợi nhuận trên 10% mỗi năm chỉ xảy ra ở 20% thử nghiệm hàng đầu.

Tất cả các biểu đồ khác trong trang MC đều được xây dựng giống nhau và bạn có thể đọc chúng bằng cùng một phương pháp.

Biểu đồ vốn chủ sở hữu cuối cùng hiển thị hàm phân phối tích lũy của giá trị cuối cùng của vốn chủ sở hữu (vào cuối giai đoạn thử nghiệm)

Biểu đồ lợi nhuận hàng năm hiển thị hàm phân phối tích lũy của tỷ lệ phần trăm lợi nhuận hàng năm kép của thử nghiệm

Biểu đồ Max. Drawdown $ và Max. Drawdown % hiển thị hàm phân phối tích lũy của các lần drawdown (khoảng cách từ đỉnh đến đô la valey/phần trăm tối đa) trong quá trình thử nghiệm

Biểu đồ Lowest Equity hiển thị hàm phân phối tích lũy của vốn chủ sở hữu thấp nhất từng gặp trong quá trình thử nghiệm

Làm thế nào để kiểm soát nó từ cấp công thức?

Ngoài việc sử dụng hộp thoại Cài đặt, bạn có thể kiểm soát trình mô phỏng Monte Carlo bằng hàm SetOption(). Bạn cũng có thể lấy các giá trị đó bằng hàm GetOption.

SetOption("MCEnable", 0); // value == 0 vô hiệu hóa mô phỏng MC SetOption("MCEnable", 1); // value == 1 chỉ bật MC trong các lần kiểm tra ngược danh mục đầu tư (mặc định) SetOption("MCEnable", 2); // value == 2 buộc phải bật MC ở mọi nơi (trong mọi chế độ bao gồm cả tối ưu hóa - CHẬM!)

Lưu ý rằng việc bật MC trong quá trình tối ưu hóa không được khuyến khích trừ khi bạn thực sự sử dụng số liệu MC làm mục tiêu tối ưu hóa thông qua trình kiểm tra ngược tùy chỉnh
hoặc sử dụng phân phối MC trong quá trình tối ưu hóa. Quy trình Monte Carlo tốn kém về mặt tính toán và trong khi vài trăm mili giây được thêm vào một lần kiểm tra ngược không quan trọng lắm, trong trường hợp tối ưu hóa khi chúng được nhân với số bước, bạn có thể dễ dàng tăng thời gian tối ưu hóa theo cấp số nhân. Vì vậy, trừ khi bạn THỰC SỰ cần phân phối MC làm mục tiêu tối ưu hóa và số liệu tùy chỉnh, KHÔNG bật MC trong quá trình tối ưu hóa.

SetOption("MCRuns", 1000); // định nghĩa số lần chạy mô phỏng MC (thực hiện) Các tham số MC khác có thể được thiết lập bằng SetOption và lấy lại bằng GetOption:
• "MCChartEquityCurves" (đúng/sai)
• "MCStrawBroomLines" (0..100)
• "MCPosSizePctEquity" (0..100)
• "MCPosSizeMethod" - 0 - không thay đổi, 1 - kích thước cố định, 2 - số lượng không đổi, 3 - phần trăm vốn chủ sở hữu,

• "MCPosSizeShares" (số),
• "MCPosSizeValue" (số)
• "MCPosSizePctEquity" (số)
• "MCUseEquityChanges" (số), 1 có nghĩa là sử dụng thay đổi vốn chủ sở hữu thay vì danh sách giao dịch
• "MCChartEquityScale" (số), 1 cho thang logarit, 0 cho tuyến tính scale
• "MCLogScaleFinalEquity" (số), 1 cho thang logarit, 0 cho thang tuyến tính
• "MCLogScaleDrawdown" (số), 1 cho thang logarit, 0 cho thang tuyến tính
• "MCNegativeDrawdown" (số), 1 - sử dụng số âm để rút tiền (CDF rút tiền ngược)

Làm thế nào để thêm số liệu tùy chỉnh dựa trên phân phối thử nghiệm MC vào báo cáo kiểm tra ngược?

Ngoài báo cáo MC tích hợp, bạn có thể thêm số liệu tùy chỉnh của riêng mình vào báo cáo bằng phương thức GetMonteCarloSim() của đối tượng Backtester và đối tượng MonteCarloSim mà hàm này trả về. Nếu bạn mới biết đến số liệu tùy chỉnh, vui lòng tham khảo phần "Cách thêm số liệu tùy chỉnh vào báo cáo kiểm tra ngược" của hướng dẫn này trước.

Đối tượng MonteCarloSim có một hàm GetValue("field", percentile) cho phép truy cập các giá trị CDF. Các giá trị "trường" khả dụng là:

• "FinalEquity"
• "CAR"
• "LowestEquity"
• "MaxDrawdown"
• "MaxPercDrawdown"

Đây là mã mẫu trình bày cách thêm FinalEquity và CAR ở phần trăm thứ 30 vào báo cáo:

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

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

bo.Backtest(); // chạy quy trình kiểm tra ngược mặc định

// truy cập kết quả Monte Carlo
// lưu ý 1: có thể là NULL nếu MC KHÔNG được bật
// lưu ý 2: Kết quả MC khả dụng sau Backtest() hoặc PostProcess

// khi mô phỏng MC được thực hiện trong giai đoạn cuối của quá trình xử lý hậu kỳ mc = bo.GetMonteCarloSim();

if( mc )
{
```

```
// lấy phần trăm thứ 30 của vốn chủ sở hữu cuối cùng và phân phối CAR
bo.AddCustomMetric( "FinalEq30", mc.GetValue( "FinalEquity", 30 ) ); bo.AddCustomMetric(
"CAR30", mc.GetValue( "CAR", 30 ) );


// bạn cũng có thể kết hợp số liệu thống kê MC với số liệu thống kê bình thường st =
bo.GetPerformanceStats(0);
bo.AddCustomMetric( "CAR30/MDD", mc.GetValue( "CAR", 30 ) / st.GetValue(
"MaxSystemDrawdownPercent" ) );
}
}
```

Sau khi số liệu tùy chỉnh được thêm vào, nó có thể được sử dụng làm mục tiêu Tối ưu hóa (đừng quên thay đổi MCEnable thành 2) và được sử dụng trong quy trình kiểm tra Walk Forward như hàm mục tiêu. Để chọn số liệu tùy chỉnh làm mục tiêu tối ưu hóa, bạn cần nhập tên của số liệu chính xác như tên xuất hiện trong lệnh gọi AddCustomMetric vào trường "Mục tiêu tối ưu hóa" trong hộp thoại Cài đặt, trang Walk Forward. Theo cách này, bạn có thể chạy thử nghiệm tối ưu hóa/đi bộ về phía trước được chỉ đạo bởi các giá trị phân phối mô phỏng MC. Ví dụ, thay vì sử dụng CAR/MDD, bạn có thể sử dụng CAR30/MDD (MC CAR phần trăm thứ 30 chia cho mức giảm tối đa của hệ thống).

Còn thử nghiệm ngẫu nhiên Monte Carlo thay vì thử nghiệm bootstrap thì sao?

Thử nghiệm ngẫu nhiên Monte Carlo khác với thử nghiệm bootstrap vì nó không sử dụng danh sách giao dịch thực tế (đã thực hiện) từ thử nghiệm ngược mà cố gắng sử dụng "tất cả các lợi nhuận riêng lẻ bất cứ khi nào chúng được thực hiện hoặc giả thuyết". Ví dụ, khi hệ thống giao dịch tạo ra nhiều tín hiệu hơn mức chúng ta thực sự có thể giao dịch do sức mua hạn chế, thì chúng ta phải chọn giao dịch nào sẽ thực hiện và giao dịch nào sẽ bỏ qua. Thông thường, lựa chọn này là một phần của hệ thống giao dịch và trong AmiBroker, biến PositionScore cho người thử nghiệm ngược biết vị thế nào được ưu tiên và nên giao dịch. Trong thử nghiệm ngẫu nhiên, thay vì sử dụng một số PositionScore phân tích/xác định, bạn sử dụng một PositionScore ngẫu nhiên. Nếu có nhiều tín hiệu mở vị thế hơn mức chúng ta có thể tiếp nhận, quá trình này sẽ dẫn đến các lựa chọn giao dịch ngẫu nhiên. Bây giờ, bằng cách sử dụng hàm Optimize() và PositionScore ngẫu nhiên, chúng ta có thể chạy hàng nghìn lựa chọn ngẫu nhiên như vậy để tạo ra thử nghiệm ngẫu nhiên Monte Carlo:

```
step = Optimize( "step", 1, 1, 1000, 1 ); // 1000 backtests
// với các lựa chọn giao dịch ngẫu nhiên từ vũ trụ rộng lớn (hãy đảm bảo bạn chạy nó trên
các danh sách theo dõi lớn)
PositionScore = mtRandom();
```

Thử nghiệm ngẫu nhiên có một nhược điểm lớn: không thể sử dụng trong nhiều trường hợp. Khi hệ thống không tạo ra đủ tín hiệu cho mỗi thanh, không có nhiều (nếu có) để lựa chọn. Ngoài ra, quan trọng hơn, ngẫu nhiên MC đưa ra giả định sai lầm rằng tất cả "cơ hội giao dịch" (tín hiệu) đều bình đẳng. Trong nhiều trường hợp, chúng không bình đẳng. Hệ thống giao dịch của chúng tôi thường có cách cụ thể, xác định để chọn giao dịch từ nhiều cơ hội bằng một số loại xếp hạng/ghi điểm. Khi hệ thống sử dụng điểm số (xếp hạng) làm thành phần cốt lõi của hệ thống (hệ thống luân phiên thực hiện điều đó) - nếu bạn thay thế điểm số phân tích bằng số ngẫu nhiên, bạn chỉ đang kiểm tra tiếng ồn trắng chứ không phải hệ thống.

Kim tự tháp (mở rộng quy mô vào/ra) và nhiều loại tiền tệ trong trình kiểm tra ngược danh mục đầu tư

QUAN TRỌNG: Vui lòng đọc trước Hướng dẫn: Kiểm tra ngược ý tưởng giao dịch của bạn và Kiểm tra ngược danh mục đầu tư

Bắt đầu từ phiên bản 4.70, trình kiểm tra ngược danh mục đầu tư cho phép mở rộng vị thế và hỗ trợ nhiều loại tiền tệ. Lưu ý rằng các tính năng nâng cao này chỉ được hỗ trợ bởi trình kiểm tra ngược PORTFOLIO. Trình kiểm tra ngược chứng khoán đơn cũ và hàm equity() chứng khoán đơn KHÔNG hỗ trợ các tính năng này.

Pyramiding / Scaling

Hai hằng số đặc biệt: sigScaleIn / sigScaleOut được thêm vào để cung cấp phương tiện cho backtester biết khi nào bạn muốn scale-in/out

Tất cả những gì bạn phải làm để triển khai pyraminding là:
- Gán sigScaleIn cho biến BUY/SHORT nếu bạn muốn scale-in (tăng kích thước của) vị thế LONG/SHORT
- Gán sigScaleOut cho biến BUY/SHORT nếu bạn muốn scale-out (giảm kích thước của) vị thế LONG/SHORT

Kích thước scaleing được xác định bởi biến PositionSize, trong trường hợp scaleing, biến này không xác định kích thước vị thế tuyệt đối mà là tăng hoặc giảm đô la.

QUAN TRỌNG: Xin lưu ý rằng backtester coi giao dịch mà bạn scale-in/out là giao dịch DUY NHẤT (tức là sẽ hiển thị một hàng trong danh sách giao dịch). Sự khác biệt duy nhất so với giao dịch thông thường là nó sẽ tính giá nhập trung bình (và tỷ giá hối đoái nhập trung bình) dựa trên tất cả các mục nhập một phần và giá thoát trung bình (và tỷ giá hối đoái thoát trung bình) dựa trên tất cả các mục nhập/thoát ngang giá và sẽ hiển thị giá trung bình trong trường giá nhập/thoát. Tất nhiên, hoa hồng được áp dụng chính xác cho mỗi mục nhập/thoát (một phần) tùy thuộc vào quy mô mua/bán một phần.

Nếu bạn muốn xem chi tiết về việc mở rộng quy mô, bạn phải chạy thử nghiệm ngược ở chế độ "NHẬT KÝ CHI TIẾT" vì chỉ khi đó bạn mới thấy cách mở rộng quy mô hoạt động như thế nào và giá trung bình được tính như thế nào.

Cũng lưu ý rằng hỗ trợ mở rộng quy mô và nhiều loại tiền tệ chỉ khả dụng trong trình kiểm tra ngược danh mục đầu tư. Trình kiểm tra ngược cũ cũng như hàm Equity() KHÔNG xử lý việc mở rộng quy mô hoặc nhiều loại tiền tệ (chúng chỉ bỏ qua các lệnh mở rộng quy mô).

Ví dụ dễ hiểu:

Ví dụ 1: trung bình chi phí đô la (mỗi tháng bạn mua cổ phiếu với số tiền cố định)
FixedDollarAmount = 500;
MonthBegin = Month() != Ref(Month(), -1); FirstPurchase = Cum(MonthBegin) == 1;
Buy = IIf(FirstPurchase, 1, // True (hoặc 1) biểu thị tín hiệu mua thông thường IIf(MonthBegin, sigScaleIn, // mỗi tháng tăng vị thế

0)); // nếu không thì không có tín hiệu

Sell = 0; // chúng tôi không bán

PositionSize = FixedDollarAmount;

Ví dụ 2: trung bình chi phí đô la
(công thức đơn giản hóa vì AB coi sigScaleIn đầu tiên là mua)

FixedDollarAmount = 500;
MonthBegin = Month() != Ref(Month(), -1); FirstPurchase = Cum(MonthBegin) == 1;
Buy = IIf(MonthBegin, sigScaleIn, 0); // mỗi tháng tăng vị thế

Bán = 0; // chúng tôi không bán

PositionSize = FixedDollarAmount;

Ví dụ 3: tăng vị thế khi lợi nhuận tạo ra từ giao dịch không có kim tự tháp trở nên lớn hơn 5% và giảm vị thế khi thua lỗ lớn hơn -5%

// ngưỡng thay đổi vốn chủ sở hữu phần trăm khi thực hiện kim tự tháp PyramidThreshold = 5;

// các quy tắc giao dịch thông thường (không có kim tự tháp)
Mua = Cross(MACD(), Signal());
Bán = Cross(Signal(), MACD());

e = Equity(1); // tạo vốn chủ sở hữu mà không có hiệu ứng kim tự tháp PcntProfit = 100 * (e - ValueWhen(Buy, e))/ValueWhen(Buy, e); InTrade = Flip(Buy, Sell);
// ExRem được sử dụng ở đây để đảm bảo rằng việc mở rộng quy mô xảy ra
// chỉ một lần kể từ khi tham gia giao dịch
DoScaleIn = ExRem(InTrade AND PcntProfit > PyramidThreshold, Sell); DoScaleOut = ExRem(InTrade AND PcntProfit < -PyramidThreshold, Sell);

// sửa đổi các quy tắc để xử lý kim tự tháp
Buy = Buy + sigScaleIn * DoScaleIn + sigScaleOut * DoScaleOut;

PositionSize = IIf(DoScaleOut, 500, 1000); // nhập và mở rộng kích thước $1000, mở rộng kích thước: $500

Ví dụ 4: thoát một phần (mở rộng) khi dừng mục tiêu lợi nhuận

Ví dụ về mã thoát 50% ở mục tiêu lợi nhuận đầu tiên, 50% ở mục tiêu lợi nhuận tiếp theo và mọi thứ ở điểm dừng theo sau:

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


// hệ thống sẽ thoát
// 50% vị thế nếu lệnh dừng MỤC TIÊU LỢI NHUẬN ĐẦU TIÊN bị chạm
// 50% vị thế là lệnh dừng MỤC TIÊU LỢI NHUẬN THỨ HAI bị chạm
// 100% vị thế nếu lệnh DỪNG THEO DÕI bị chạm


FirstProfitTarget = 10; // lợi nhuận SecondProfitTarget = 20; // theo phần trăm TrailingStop =
10; // cũng theo phần trăm


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 )
{
// mục tiêu lợi nhuận đầu tiên đạt được - scale-out exit = 1;
Buy[ i ] = sigScaleOut;
}


if( exit == 1 AND
High[ i ] >= ( 1 + SecondProfitTarget * 0.01 ) * priceatbuy )
{
// mục tiêu lợi nhuận thứ hai đạt được - exit exit = 2;
SellPrice[ i ] = Max( Open[ i ], ( 1 + SecondProfitTarget * 0.01 ) *
priceatbuy );
}


if( Low[ i ] <= ( 1 - TrailingStop * 0.01 ) * highsincebuy )
{
// trailing stop hit - exit exit = 3;
SellPrice[ i ] = Min( Open[ i ], ( 1 - TrailingStop * 0.01 ) * highsincebuy );
}


if( exit >= 2 )
```

```
{
Buy[ i ] = 0;
Sell[ i ] = exit + 1; // đánh dấu mã thoát thích hợp exit = 0;
priceatbuy = 0; // đặt lại giá highsincebuy = 0;
}
}
}
```

```
SetPositionSize( 50, spsPercentOfEquity );
SetPositionSize( 50, spsPercentOfPosition * ( Buy == sigScaleOut ) ); // mở rộng 50% vị thế
```

Hỗ trợ nhiều loại tiền tệ

Trình kiểm tra ngược danh mục đầu tư cho phép kiểm tra ngược hệ thống trên các chứng khoán được tính bằng các loại tiền tệ khác nhau. Nó bao gồm khả năng sử dụng tỷ giá tiền tệ lịch sử (biến đổi). Tỷ giá tiền tệ có thể được xác định trong trang "Tiền tệ" trong tùy chọn. Tiền tệ mà ký hiệu được chỉ định có thể được nhập vào trang Symbol->Information.

Trang "Tiền tệ" trong Tùy chọn - cho phép xác định tiền tệ cơ sở và tỷ giá hối đoái (cố định hoặc động) cho các loại tiền tệ khác nhau. Điều này cho phép có được kết quả kiểm tra ngược chính xác khi kiểm tra chứng khoán được chỉ định bằng loại tiền tệ khác với loại tiền tệ danh mục đầu tư cơ sở của bạn.
Làm thế nào để AB biết tôi muốn báo giá cố định hay động? Có các yêu cầu sau để sử dụng điều chỉnh tiền tệ:

a) Symbol->Information, trường "Tiền tệ" hiển thị loại tiền tệ khác với loại tiền tệ CƠ SỞ
b) Loại tiền tệ phù hợp (được xác định trong Symbol) có mục nhập khớp trong trang Tùy chọn->Tiền tệ
c) tỷ giá động "FX SYMBOL" được xác định trong tùy chọn TỒN TẠI trong cơ sở dữ liệu của bạn và CÓ BÁO GIÁ cho mỗi ngày trong phạm vi phân tích.

Hộp kiểm "INVERSE" trong tùy chọn có nghĩa là gì? Ví dụ, hãy lấy EURUSD.
Khi "USD" là loại tiền tệ CƠ SỞ của bạn thì tỷ giá hối đoái EUR sẽ là "thẳng" EURUSD fx (tức là 1,3).
Nhưng khi "EUR" là loại tiền tệ CƠ SỞ của bạn thì tỷ giá hối đoái USD sẽ là NGƯỢC của EURUSD (tức là 1/1,3). Ngược lại sẽ đúng với tỷ giá FX như USDJPY (đã là "nghịch đảo").

Sử dụng cảnh báo dựa trên công thức

Giới thiệu

AmiBroker cho phép bạn xác định cảnh báo dựa trên công thức. Khi cảnh báo được kích hoạt, một văn bản có thể được hiển thị,
âm thanh do người dùng xác định được phát lại, thông báo qua email có thể được gửi và bất kỳ ứng dụng bên ngoài nào có thể được khởi chạy. Tất cả những điều này được xử lý bởi một hàm AlertIF duy nhất.
Theo mặc định, tất cả các cảnh báo đều tạo ra văn bản được hiển thị trong cửa sổ Alert Output. Để hiển thị cửa sổ này, bạn phải chọn menu Window->Alert Output.
Ngoài ra còn có cửa sổ Easy Alerts cho phép bạn xác định các cảnh báo đơn giản không yêu cầu bất kỳ mã hóa nào (nhưng không cung cấp tính linh hoạt đầy đủ của hàm AlertIf).

Cài đặt

Các cài đặt liên quan đến cảnh báo có trong tab "Alerts" của cửa sổ Tools->Preferences.

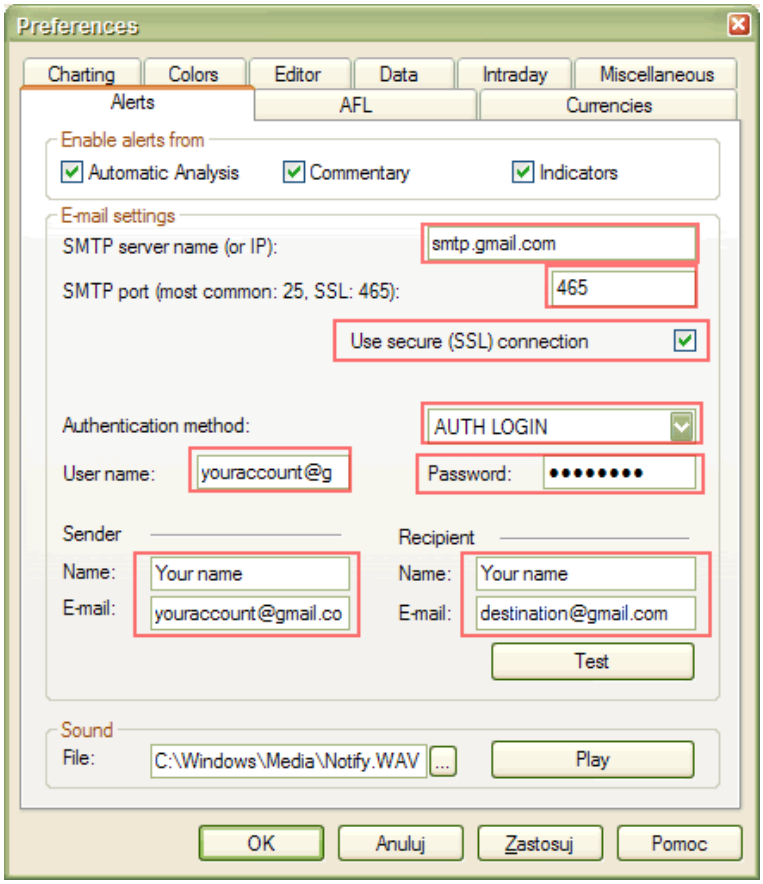
Nó cho phép xác định cài đặt tài khoản email, kiểm tra đầu ra âm thanh và xác định phần nào của AmiBroker có thể tạo cảnh báo thông qua hàm AlertIF.

Trang cài đặt email hiện cho phép lựa chọn giữa các lược đồ ủy quyền phổ biến nhất như: AUTH LOGIN (phổ biến nhất), POP3-before-SMPT (phổ biến), CRAM-MD5, LOGIN PLAIN.

Các hộp kiểm "Enable alert from" cho phép bạn chọn bật/tắt các cảnh báo được tạo bởi Automatic analysis, Commentary/Interpretation và các chỉ báo tùy chỉnh.

Cửa sổ đầu ra cảnh báo hiện có thêm một cột hiển thị nguồn cảnh báo - nếu đây là Automatic Analysis, Commentary hoặc một trong các chỉ báo tùy chỉnh của bạn. Điều này giúp bạn dễ dàng tìm ra phần nào của AmiBroker tạo cảnh báo.

Tính năng mới trong AmiBroker 5.30 - hỗ trợ SSL (kết nối an toàn) được sử dụng bởi GMail chẳng hạn. Để bật hỗ trợ SSL, bạn cần thực hiện theo các bước sau:
1. Tải xuống và chạy tiện ích bổ sung SSL từ http://www.amibroker.com/bin/SSLAddOn.exe
2. Cấu hình (Tools->Preferences->Alerts) với SSL được bật như minh họa bên dưới



Hàm AlertIF

Hàm AlertIF tương tự như WriteIF. Nhưng thay vì chỉ viết văn bản vào cửa sổ đầu ra (bình luận/diễn giải), nó cho phép:

• hướng văn bản tùy chỉnh đến cửa sổ "đầu ra cảnh báo",
• tạo âm thanh (chỉ bằng tiếng bíp máy tính hoặc từ tệp .WAV)
• gửi email
• khởi chạy bất kỳ ứng dụng bên ngoài nào

Cú pháp như sau:

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

1. BOOLEAN_EXPRESSION là biểu thức mà nếu đánh giá là True (giá trị khác không) sẽ kích hoạt cảnh báo. Nếu đánh giá là False (giá trị không) thì không có cảnh báo nào được kích hoạt. Xin lưu ý rằng chỉ xem xét các thanh gần đây nhất của lookback.

2. Chuỗi lệnh xác định hành động được thực hiện khi cảnh báo được kích hoạt. Nếu chuỗi lệnh trống, văn bản cảnh báo sẽ chỉ được hiển thị trong cửa sổ đầu ra Cảnh báo (Cửa sổ->Đầu ra Cảnh báo). Các giá trị được hỗ trợ khác của chuỗi lệnh là:

SOUND the-path-to-the-WAV-file

EMAIL
EXEC the-path-to-the-file-or-URL <các đối số tùy chọn>

Lệnh SOUND phát tệp WAV một lần.

Lệnh EMAIL gửi email đến tài khoản được xác định trong cài đặt (Công cụ->Tùy chọn->Email). Định dạng của email như sau:

Chủ đề: Cảnh báo type_name (loại) Ticker theo Ngày/Giờ Nội dung: văn bản

Lệnh EXEC khởi chạy ứng dụng bên ngoài hoặc tệp hoặc URL được chỉ định sau lệnh EXEC. <các đối số tùy chọn> được đính kèm sau tên tệp và văn bản được đính kèm ở cuối

3. Văn bản xác định văn bản sẽ được in trong cửa sổ đầu ra hoặc được gửi qua email hoặc được thêm làm đối số cho ứng dụng được chỉ định bởi lệnh EXEC

4. Loại xác định loại cảnh báo. Các loại được xác định trước là 0 - mặc định, 1 - mua, 2 - bán, 3 - bán khống, 4 - che phủ. Bạn có thể chỉ định các giá trị cao hơn và chúng sẽ có tên là "other". Kiểu là quan trọng. XEM NHẬN XÉT BÊN DƯỚI.

5. Cờ kiểm soát hành vi của hàm AlertIF. Trường này là sự kết hợp (tổng) của các giá trị sau:
(1 - hiển thị văn bản trong cửa sổ đầu ra, 2 - phát ra tiếng bíp (qua loa máy tính), 4 - không hiển thị các cảnh báo lặp lại có cùng kiểu, 8 - không hiển thị các cảnh báo lặp lại có cùng ngày/giờ) Theo mặc định, tất cả các tùy chọn này đều được BẬT.

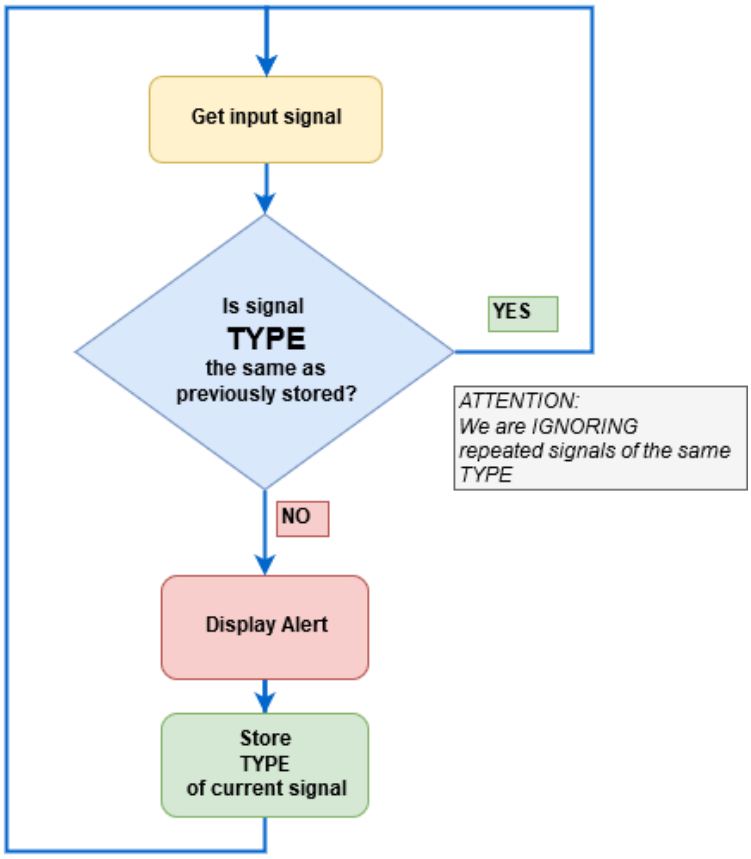
6. tham số lookback kiểm soát số lượng thanh gần đây được kiểm tra Ví dụ:
Buy = Cross(MACD(), Signal()); Sell = Cross(Signal(), MACD()); Short = Sell;
Cover = Buy;

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);
Lưu ý Lệnh EXEC sử dụng hàm ShellExecute và cho phép không chỉ các tệp EXE mà cả URL nữa.

Logic nội bộ

Hàm Alertif triển khai logic nội bộ dưới dạng máy trạng thái hữu hạn ngăn chặn các tín hiệu lặp lại cùng loại xảy ra. Trạng thái (tức là LOẠI cảnh báo cuối cùng) được lưu trữ trên cơ sở PER-SYMBOL. Vì vậy, mỗi ký hiệu có "cảnh báo cuối cùng" được đính kèm. Đối với logic theo ký hiệu, hãy xem sơ đồ bên dưới:



Ghi chú

1. Xin lưu ý một lần nữa rằng theo mặc định, hàm AlertIf không tạo ra các tín hiệu lặp lại khi cùng một lần quét được chạy nhiều lần. Trong quá trình thử nghiệm, bạn có thể muốn nhận được các tín hiệu lặp lại trong các lần quét tiếp theo. Để làm như vậy, bạn nên thay đổi cờ mặc định thành 1 + 2:

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

2. Nếu bạn chỉ muốn tạo cảnh báo trên thanh COMPLETED, bạn có thể cần thêm mã này:

barcomplete = BarIndex() < LastValue(BarIndex()); AlertIF(barcomplete AND condition, "", "Text", 1);

3. Có những hàm đơn giản hơn thực hiện các hành động như gửi email, phát âm thanh hoặc thực thi chương trình bên ngoài mà không cần tham gia vào logic nội bộ phức tạp:

• Nếu bạn chỉ cần gửi email, hãy sử dụng: SendEmail
• Nếu bạn cần phát âm thanh, hãy sử dụng PlaySound
• Nếu bạn cần thực thi chương trình, hãy sử dụng ShellExecute

Sử dụng cửa sổ giải thích

Lưu ý: Vui lòng đọc bài viết Cách viết bình luận biểu đồ của riêng bạn trước khi tiếp tục.

Cửa sổ giải thích (Cửa sổ->Giải thích) hiển thị các bình luận nhạy cảm với biểu đồ. Để thêm một bình luận, chỉ cần sử dụng Trình chỉnh sửa công thức và thêm mã bình luận sau mã cho chỉ báo. Xin lưu ý rằng để có hiệu suất tốt nhất, bạn nên sử dụng câu lệnh có điều kiện đảm bảo rằng mã giải thích chỉ được thực thi ở chế độ "bình luận".

```
if( Status("action") == actionCommentary )
{
// các câu lệnh printf ở đây....
}
```

Ví dụ:

```
Plot( Close, "Price", -1, 64 );


Plot( SAR( Prefs( 50 ), Prefs( 51 ) ), "SAR",-17, 8+16 );


if( Status("action") == actionCommentary )
{
printf("Parabolic SAR cung cấp các điểm thoát tuyệt vời. \n"); printf("Bạn nên Đóng các vị thế mua khi giá giảm xuống dưới\n"); printf("SAR VÀ Đóng các vị thế bán khi giá tăng trên SAR.\n");
printf( WriteIf( Graph1 > Đóng, "SAR ở trên mức đóng", "SAR ở dưới mức đóng" ) );
}
```

Hỗ trợ nhiều khung thời gian trong AFL

QUAN TRỌNG: Các hàm TimeFrame KHÔNG nhằm mục đích thay thế cài đặt Periodicity. Để chuyển đổi Periodicity/khoảng thời gian, bạn chỉ nên sử dụng cài đặt Periodicity. Các hàm TimeFrame CHỈ dành cho các công thức TRỘN nhiều khoảng thời gian khác nhau cùng một lúc trong một công thức.

Phiên bản 4.41 mang đến khả năng sử dụng nhiều khung thời gian (khoảng thời gian thanh) trong một công thức duy nhất. Các hàm khung thời gian có thể được chia thành 3 nhóm chức năng:

1. chuyển đổi khung thời gian của các mảng O, H, L, C, V, OI, Avg tích hợp: TimeFrameSet, TimeFrameRestore
2. nén/mở rộng các mảng đơn lẻ thành/từ khoảng thời gian đã chỉ định: TimeFrameCompress, TimeFrameExpand
3. truy cập ngay vào các mảng giá/khối lượng trong khung thời gian khác: TimeFrameGetPrice

Nhóm đầu tiên được sử dụng khi công thức của bạn cần thực hiện một số phép tính trên các chỉ báo trong khung thời gian khác với khung thời gian hiện tại đã chọn. Ví dụ: nếu bạn cần tính trung bình động 13 thanh trên dữ liệu 5 phút và trung bình mũ 9 thanh từ dữ liệu hàng giờ trong khi khoảng thời gian hiện tại là 1 phút, bạn sẽ viết:

```
TimeFrameSet( in5Minute); // chuyển sang khung 5 phút

/* MA hiện hoạt động trên dữ liệu 5 phút, ma5_13 giữ MA 13 thanh nén theo thời gian của các thanh 5 phút */

ma5_13 = MA( C, 13);

TimeFrameRestore(); // khôi phục khung thời gian về TimeFrameSet( inHourly); // chuyển sang giờ
mah_9 = EMA( C, 9); // trung bình động 9 thanh từ dữ liệu hàng giờ TimeFrameRestore(); // khôi phục khung thời gian về gốc
Plot( Close, "Price", colorWhite, styleCandle);

// vẽ trung bình mở rộng

Plot( TimeFrameExpand( ma5_13, in5Minute), "trung bình động 13 thanh từ các thanh 5 phút", colorRed);
Plot( TimeFrameExpand( mah_9, inHourly), "trung bình động 9 thanh từ các thanh hàng giờ",
colorRed);
```

TimeFrameSet(interval) - thay thế các mảng giá/khối lượng tích hợp hiện tại: open, high, low, close, volume, openint, avg bằng các thanh nén thời gian của khoảng thời gian đã chỉ định sau khi bạn chuyển sang khung thời gian khác, tất cả các phép tính và chỉ báo tích hợp

đều hoạt động trên khung thời gian đã chọn. Để quay lại khoảng thời gian ban đầu, hãy gọi hàm TimeFrameRestore(). Nếu bạn muốn gọi lại TimeFrameSet với khoảng thời gian khác, trước tiên bạn phải khôi phục khung thời gian gốc bằng TimeFrameRestore(). Interval là khoảng thời gian tính bằng giây. Ví dụ: 60 là thanh một phút. Bạn nên sử dụng hằng số thuận tiện cho các khoảng thời gian phổ biến: in1Minute, in5Minute, in15Minute, inHourly, inDaily, inWeekly, inMonthly.

Với phiên bản 4.70, bạn cũng có thể chỉ định khoảng thời gian N-tick. Điều này được thực hiện bằng cách truyền giá trị ÂM làm khoảng thời gian. Ví dụ -5 sẽ nén thanh 5 tích tắc và -133 sẽ nén 133 tích tắc. Xin lưu ý rằng việc sử dụng khoảng thời gian N-tick chỉ hoạt động nếu cơ sở dữ liệu của bạn sử dụng khoảng thời gian cơ sở Tick được đặt trong hộp thoại File -> Database Settings.

TimeFrameSet(-133); // chuyển sang khoảng thời gian 133-tick

QUAN TRỌNG: TimeFrameSet() KHÔNG tương đương với thiết lập chu kỳ trong Analysis Settings. Công dụng duy nhất của các hàm khung thời gian là khi bạn muốn có các quy tắc giao dịch dựa trên NHIỀU khung thời gian cùng một lúc. Xem chi tiết trong "Cách thức hoạt động bên trong" bên dưới.

TimeFrameRestore() - khôi phục các mảng giá được thay thế bằng SetTimeFrame. Lưu ý rằng chỉ các biến tích hợp OHLC, V, OI và Avg mới được khôi phục về khung thời gian gốc khi bạn gọi TimeFrameRestore(). Tất cả các biến khác được tạo khi ở trong khung thời gian khác vẫn được nén. Để giải nén chúng về khoảng thời gian gốc, bạn phải sử dụng TimeFrameExpand.

Khi bạn chuyển đổi khung thời gian bằng TimeFrameSet, tất cả các hàm AFL đều hoạt động trên khung thời gian này cho đến khi bạn chuyển lại khung thời gian về khoảng thời gian ban đầu bằng TimeFrameRestore hoặc đặt lại thành khoảng thời gian khác bằng TimeFrameSet. Tốt nhất là LUÔN gọi TimeFrameRestore khi bạn hoàn tất quá trình xử lý trong các khung thời gian khác.

Khi khung thời gian được chuyển sang khoảng thời gian khác ngoài khoảng thời gian ban đầu, kết quả của tất cả các hàm được gọi vì TimeFrameSet cũng được nén theo thời gian. Nếu bạn muốn hiển thị chúng trong khung thời gian ban đầu, bạn sẽ cần phải 'mở rộng' chúng như mô tả sau. Các biến được tạo và gán trước khi gọi TimeFrameSet() vẫn giữ nguyên trong khung thời gian chúng được tạo. Hành vi này cho phép kết hợp không giới hạn các khung thời gian khác nhau trong một công thức.

LƯU Ý rằng bạn chỉ có thể nén dữ liệu từ khoảng thời gian ngắn hơn sang khoảng thời gian dài hơn. Vì vậy, khi làm việc với dữ liệu 1 phút, bạn có thể nén thành dữ liệu 2, 3, 4, 5, 6, N phút. Nhưng khi làm việc với dữ liệu 15 phút, bạn không thể có được thanh dữ liệu 1 phút. Tương tự như vậy, nếu bạn chỉ có dữ liệu EOD, bạn không thể truy cập vào khung thời gian trong ngày.

Nhóm thứ hai: TimeFrameCompress/TimeFrameExpand cho phép nén và mở rộng các mảng đơn lẻ sang/từ các khung thời gian khác nhau. Đặc biệt đáng đề cập là TimeFrameExpand được sử dụng để giải nén các biến mảng được tạo trong các khung thời gian khác nhau. Giải nén là bắt buộc để hiển thị đúng mảng được tạo trong các khung thời gian khác nhau. Ví dụ, nếu bạn muốn hiển thị đường trung bình động hàng tuần, thì phải

'mở rộng' để dữ liệu của một thanh hàng tuần bao gồm năm thanh hàng ngày (Thứ Hai-Thứ Sáu) của tuần tương ứng.

TimeFrameExpand(array, interval, mode = expandLast) - mở rộng mảng nén thời gian từ khung thời gian 'interval' sang khung thời gian cơ sở ('interval' phải khớp với giá trị được sử dụng trong TimeFrameCompress hoặc TimeFrameSet)
Các chế độ khả dụng:

expandLast - giá trị nén được mở rộng bắt đầu từ thanh cuối cùng trong khoảng thời gian nhất định (ví dụ giá đóng/cao/thấp hàng tuần khả dụng trên thanh của Thứ Sáu)
expandFirst - giá trị nén được mở rộng bắt đầu từ thanh đầu tiên trong khoảng thời gian nhất định (ví dụ giá mở hàng tuần khả dụng trên thanh của Thứ Hai)
expandPoint - mảng kết quả chỉ nhận được các giá trị không rỗng cho thanh cuối cùng trong khoảng thời gian nhất định (tất cả các thanh còn lại đều là Null (trống)).
Lưu ý: expandFirst được sử dụng trên giá khác với giá mở có thể xem xét trong tương lai. Ví dụ, nếu bạn tạo chuỗi HIGH hàng tuần, hãy mở rộng chuỗi này thành khoảng thời gian hàng ngày bằng cách sử dụng expandFirst để biết được vào MONDAY, giá trị cao nhất trong toàn bộ tuần là bao nhiêu.

QUAN TRỌNG: TimeFrameExpand LÀ YÊU CẦU đối với bất kỳ công thức nào sử dụng hàm TimeFrame*. Nếu bạn không mở rộng dữ liệu nén theo thời gian, bạn sẽ có dấu thời gian không chính xác (xem mô tả bên dưới trong "Cách thức hoạt động").

TimeFrameCompress được cung cấp để hoàn thiện và có thể sử dụng khi bạn muốn nén một mảng mà không ảnh hưởng đến các mảng OHLC, V tích hợp sẵn. Nếu bạn gọi TimeFrameCompress, nó sẽ không ảnh hưởng đến kết quả của các hàm khác.

```
wc = TimeFrameCompress( Close, inWeekly );

/* bây giờ khung thời gian vẫn không thay đổi (ví dụ như hàng ngày) và MA của chúng ta sẽ hoạt động trên dữ liệu hàng ngày */

dailyma = MA( C, 14 );

/* nhưng nếu chúng ta gọi MA trên mảng đã nén, nó sẽ cung cấp MA từ khung thời gian khác
*/

weeklyma = MA( wc, 14 ); // lưu ý rằng đối số là mảng nén theo thời gian Plot( dailyma, "DailyMA", colorRed );
weeklyma = TimeFrameExpand( weeklyma, inWeekly ); // mở rộng để hiển thị Plot( weeklyma, "WeeklyMA", colorBlue );
```

Trong công thức này, khung thời gian vẫn giữ nguyên thiết lập ban đầu, chúng tôi chỉ nén một mảng.

TimeFrameCompress(array, interval, mode = compressLast)
- nén một mảng thành một khoảng thời gian nhất định bằng cách sử dụng các chế độ nén có sẵn: compressLast - giá trị cuối cùng (đóng) của mảng trong khoảng thời gian

compressOpen - giá trị mở của mảng trong khoảng thời gian compressHigh - giá trị cao nhất của mảng trong khoảng thời gian compressLow - giá trị thấp nhất của mảng trong khoảng thời gian compressVolume - tổng các giá trị của mảng trong khoảng thời gian

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

Nhóm thứ ba chỉ bao gồm một hàm hữu ích: TimeFrameGetPrice cho phép tham chiếu giá và khối lượng từ các khung thời gian khác mà không cần chuyển đổi /nén/mở rộng khung thời gian. Chỉ cần một lệnh gọi hàm để lấy giá từ khung thời gian cao hơn. Nó cũng cho phép tham chiếu không chỉ các thanh hiện tại mà cả các thanh trước đó từ các khung thời gian khác nhau.

TimeFrameGetPrice(pricefield, interval, shift = 0, mode = expandFirst);
- tham chiếu các trường OHLCV từ các khung thời gian khác. Điều này hoạt động ngay lập tức mà không cần phải gọi TimeFrameSet.
Trường giá là một trong những trường sau: "O", "H", "L", "C", "V", "I" (lãi suất mở). Interval là khoảng thời gian thanh tính bằng giây. shift cho phép tham chiếu dữ liệu quá khứ (giá trị âm) và tương lai (giá trị dương) trong khung thời gian cao hơn. Ví dụ -1 cung cấp dữ liệu của thanh trước đó (giống như trong hàm Ref nhưng điều này hoạt động trong khung thời gian cao hơn).

Ví dụ:

TimeFrameGetPrice("O", inWeekly, -1) // cung cấp cho bạn giá mở cửa tuần trước
TimeFrameGetPrice("C", inWeekly, -3) // cung cấp cho bạn giá đóng cửa tuần trước 3 tuần trước
TimeFrameGetPrice("H", inWeekly, -2) // cung cấp cho bạn giá cao tuần trước 2 tuần trước
TimeFrameGetPrice("O", inWeekly, 0) // cung cấp cho bạn giá mở cửa tuần này.
TimeFrameGetPrice("H", inDaily, -1) // cung cấp giá cao trong ngày trước khi làm việc trên dữ liệu trong ngày
Shift hoạt động như trong hàm Ref() nhưng được áp dụng cho khung thời gian nén.

Lưu ý các hàm này hoạt động giống như 3 hàm lồng nhau này
TimeFrameExpand(Ref(TimeFrameCompress(array, interval, compress(tùy thuộc vào trường được sử dụng)), shift), interval, expandFirst)
do đó nếu shift = 0 dữ liệu nén có thể nhìn vào tương lai (mức cao hàng tuần có thể được biết vào thứ hai). Nếu bạn
muốn viết một hệ thống giao dịch bằng hàm này, vui lòng đảm bảo tham chiếu dữ liệu QUÁ KHỨ bằng cách sử dụng giá trị shift âm.

Điểm khác biệt duy nhất là TimeFrameGetPrice nhanh hơn 2 lần so với Expand/Compress lồng nhau.

Lưu ý về hiệu suất của các hàm TimeFrame:

a) Các phép đo được thực hiện trên Athlon 1,46 GHz, 18500 thanh hàng ngày được nén thành khung thời gian hàng tuần

TimeFrameGetPrice("C", inWeekly, 0) - 0,0098 giây (9,8 mili giây) TimeFrameSet(inWeekly) - 0,012 giây (12 mili giây) TimeFrameRestore( ) - 0,006 giây (6 mili giây) TimeFrameCompress(Close, inWeekly, compressLast); - 0,0097 giây (9,7 mili giây) TimeFrameExpand(array, inWeekly, expandLast); - 0,0098 giây (9,8 mili giây)
b) Các phép đo được thực hiện trên Athlon 1,46 GHz, 1000 thanh hàng ngày được nén thành khung thời gian hàng tuầntất cả các hàm đều dưới 0,0007 giây (0,7 mili giây)

Nó hoạt động như thế nào bên trong?

Các hàm khung thời gian không thay đổi BarCount - chúng chỉ nén các mảng để bạn có N-bar đầu tiên được điền bằng các giá trị NULL và sau đó - phần cuối cùng của mảng chứa các giá trị nén thời gian thực tế.

Đây là lý do tại sao việc mở rộng dữ liệu trở lại khung ban đầu bằng TimeFrameExpand là điều cần thiết.

Khám phá đơn giản sau đây cho thấy những gì xảy ra sau khi bạn chuyển sang khung thời gian cao hơn. Chạy Khám phá trên ký hiệu hiện tại, tất cả các báo giá, chu kỳ được đặt thành hàng ngày và bạn sẽ thấy cột "weekly close compress" chứa các giá trị trống ở đầu và dữ liệu nén hàng tuần ở cuối mảng.

Filter = 1;

AddColumn(Close, "Daily close");

TimeFrameSet(inWeekly);

AddColumn(wc = Close, "weekly close compress"); TimeFrameRestore();

AddColumn(TimeFrameExpand(wc, inWeekly), "weekly close extended");

VÍ DỤ

VÍ DỤ 1: Vẽ MACD hàng tuần và mũi tên chéo từ dữ liệu hàng ngày

TimeFrameSet(inWeekly);

m = MACD(12, 26); // MACD từ dữ liệu TUẦN TimeFrameRestore();

m1 = TimeFrameExpand(m, inWeekly); Plot(m1, "Weekly MACD", colorRed);

PlotShapes(Cross(m1, 0) * shapeUpArrow, colorGreen); PlotShapes(Cross(0, m1) * shapeDownArrow, colorGreen);

VÍ DỤ 2: biểu đồ nến hàng tuần chồng lên biểu đồ giá hàng ngày theo đường

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);

VÍ DỤ 3: Hệ thống màn hình ba đơn giản

/* chuyển sang khung thời gian hàng tuần */ TimeFrameSet(inWeekly);
whist = MACD(12, 26) - Signal(12, 26, 9);
wtrend = ROC(whist, 1); // xu hướng hàng tuần - thay đổi một tuần của biểu đồ histogram macd hàng tuần
TimeFrameRestore();

/* mở rộng MACD được tính toán thành hàng ngày để chúng ta có thể sử dụng nó với các tín hiệu hàng ngày */ wtrend = TimeFrameExpand(wtrend, inWeekly);

/* elder ray */
bullpower= Cao - EMA(Đóng,13); bearpower= Thấp - EMA(Đóng,13);

Mua = wtrend > 0 /* Màn hình thứ nhất: xu hướng hàng tuần tích cực */
VÀ
bearpower < 0 VÀ bearpower > Tham chiếu(bearpower, -1) /* Màn hình thứ hai bear power âm nhưng đang tăng */
VÀ
H > Tham chiếu(H, -1); /* Màn hình thứ 3, nếu giá đạt mức cao mới */

BuyPrice = Tham chiếu(H, -1); // mức dừng mua;

Sell = 0; // chỉ thoát khi dừng
ApplyStop(stopTypeProfit, stopModePercent, 30, True); ApplyStop(stopTypeTrailing, stopModePercent, 20, True);

Sử dụng hiệu quả đa luồng

AmiBroker 5.50 hỗ trợ đầy đủ đa luồng (thực thi song song trên tất cả các lõi CPU) trong cả biểu đồ và cửa sổ Phân tích mới. Điều này giúp tăng đáng kể tốc độ hoạt động và cải thiện khả năng phản hồi của ứng dụng vì các luồng thực thi AFL của công nhân không chặn giao diện người dùng. Ví dụ, trên Intel i7 4 lõi có thể chạy tới 8 luồng, nó có thể chạy nhanh hơn tới 8 lần so với cửa sổ Phân tích cũ. Tốc độ tăng chính xác phụ thuộc vào độ phức tạp của công thức (càng phức tạp thì càng có thể tăng tốc), lượng dữ liệu được xử lý (truy cập RAM có thể không nhanh bằng CPU do đó hạn chế khả năng tăng tốc).

Chương này mô tả cách tránh những cạm bẫy có thể ảnh hưởng đến hiệu suất đa luồng.

Hiểu cách triển khai đa luồng

Trước tiên, điều quan trọng là phải hiểu một quy tắc đơn giản - trong AmiBroker, một luồng có thể chạy một thao tác trên dữ liệu của một ký hiệu:

1 thao tác * 1 ký hiệu = 1 luồng

Thao tác này hiển thị một khung biểu đồ, quét, khám phá, kiểm tra ngược, tối ưu hóa. Hậu quả như sau: khung biểu đồ đơn luôn sử dụng một luồng. Ngoài ra, một kiểm tra ngược hoặc tối ưu hóa đơn chạy trên một ký hiệu chỉ sử dụng một luồng.

Nhưng một biểu đồ bao gồm 3 khung sử dụng 3 luồng, mặc dù tất cả chúng đều hoạt động trên cùng một ký hiệu. Vì vậy, chúng ta cũng có thể viết:

N thao tác * 1 ký hiệu = N luồng

Chúng ta cũng có thể chạy một thao tác (như quét/khám phá/kiểm tra ngược/tối ưu hóa) trên nhiều ký hiệu, sau đó

1 thao tác * N ký hiệu = N luồng

Tất nhiên, bạn cũng có thể chạy nhiều cửa sổ Phân tích, mỗi cửa sổ chạy nhiều ký hiệu hoặc chạy nhiều biểu đồ trên nhiều ký hiệu, sau đó

P thao tác * N ký hiệu = (P * N) luồng

Điều quan trọng nữa là phải hiểu rằng một số thao tác không chỉ bao gồm phần thực thi AFL mà còn có một số xử lý bổ sung và/hoặc công việc giao diện người dùng. Trong những trường hợp như vậy, chỉ có thể thực hiện thực thi AFL với nhiều luồng. Điều này có hậu quả đối với chế độ Kiểm tra ngược cá nhân sẽ được mô tả chi tiết hơn.

Lưu ý: Trong phiên bản 5.70 có một ngoại lệ đối với quy tắc này: tối ưu hóa cá nhân đa luồng mới, cho phép chạy tối ưu hóa ký hiệu đơn bằng nhiều luồng.

Giới hạn

Số luồng thực sự được khởi chạy phụ thuộc vào CPU của bạn và phiên bản AmiBroker bạn đang sử dụng. Phiên bản Tiêu chuẩn có giới hạn 2 (hai) luồng cho mỗi cửa sổ Phân tích. Phiên bản Chuyên nghiệp có giới hạn 32 luồng cho mỗi cửa sổ Phân tích. Ngoài giới hạn này, AmiBroker sẽ phát hiện số lượng bộ xử lý logic được Windows báo cáo (ví dụ: một CPU Intel i7 920 duy nhất được nhận dạng là 8 bộ xử lý logic (4 lõi x 2 siêu phân luồng)) và sẽ không chạy nhiều luồng hơn cho mỗi cửa sổ Phân tích duy nhất so với số lượng bộ xử lý logic.

Những lỗi sai phổ biến

Sau đây là những lĩnh vực lập trình AFL cần được chú ý nếu bạn muốn viết công thức AFL thân thiện với đa luồng:

1. Tránh sử dụng OLE / CreateObject
2. Giảm thiểu việc sử dụng AddToComposite / Foreign
3. Sử dụng hiệu quả và đúng các biến tĩnh
4. Triển khai tiền xử lý / khởi tạo trong cửa sổ Phân tích
5. Truy cập ~~~Ký hiệu Equity

Nói chung, công thức AFL chỉ có thể chạy ở tốc độ tối đa nếu không truy cập bất kỳ tài nguyên dùng chung nào. Bất kỳ nỗ lực nào truy cập tài nguyên dùng chung đều có thể dẫn đến việc thực thi công thức phải chờ phần semaphore/critical bảo vệ tài nguyên dùng chung khỏi việc sửa đổi đồng thời.

1. Tránh sử dụng OLE / CreateObject

AmiBroker hỗ trợ đầy đủ việc gọi các đối tượng OLE từ cấp công thức AFL và vẫn an toàn khi sử dụng, nhưng có những lý do kỹ thuật để phản đối việc sử dụng OLE. Lý do quan trọng nhất là OLE chậm, đặc biệt là khi được gọi không phải từ luồng "chủ sở hữu".

OLE được Microsoft phát triển vào những năm 1990 trong thời đại 16-bit, đây là công nghệ cũ và nó ngăn chặn hiệu quả các luồng chạy ở tốc độ tối đa vì tất cả các lệnh gọi OLE phải được phục vụ bởi một và chỉ một luồng giao diện người dùng. Để biết thêm chi tiết, hãy xem bài viết này: http://blogs.msdn.com/b/oldnewthing/archive/2008/04/24/8420242.aspx

Vì lý do này, nếu có thể, bạn nên tránh sử dụng OLE / CreateObject trong các công thức của mình.

Nếu bạn không làm như vậy, hiệu suất sẽ bị ảnh hưởng. Bất kỳ lệnh gọi nào đến OLE từ luồng công nhân đều khiến thông báo được đăng lên cửa sổ ẩn OLE và chờ luồng UI ứng dụng chính xử lý yêu cầu. Nếu nhiều luồng thực hiện tương tự, hiệu suất sẽ dễ dàng giảm xuống mức luồng đơn, vì tất cả các lệnh gọi OLE đều được xử lý bởi luồng UI chính.

Không chỉ vậy. Các luồng đang chờ OLE có thể dễ dàng bị bế tắc khi máy chủ OLE đang bận với một số công việc khác. AmiBroker chứa một số mã được cấp bằng sáng chế công nghệ cao kiểm tra tình trạng bế tắc OLE như vậy và có thể mở khóa từ đó, nhưng có thể

mất tới 10 giây để mở khóa. Thậm chí còn tệ hơn. Các lệnh gọi OLE được thực hiện từ luồng không phải UI phải chịu chi phí nhắn tin và sắp xếp và có thể chậm hơn tới 30 lần so với khi chúng được gọi từ cùng một luồng UI chính của quy trình. Để tránh tất cả những rắc rối đó, hãy tránh sử dụng OLE nếu có thể.

Ví dụ, thay vì sử dụng OLE để thực hiện RefreshAll như thế này:

AB = CreateObject("Broker.Application"); // TRÁNH ĐIỀU NÀY AB.RefreshAll(); // TRÁNH ĐIỀU NÀY

Sử dụng hàm RequestTimedRefresh gốc của AmiBroker, nhanh hơn gấp bội và không gây ra bất kỳ sự cố nào. Nếu bạn muốn làm mới UI sau khi Quét/Phân tích/Kiểm tra ngược, hãy sử dụng SetOption("RefreshWhenCompleted", True)

Hãy nhớ rằng trong hầu hết các trường hợp, việc làm mới hoàn toàn tự động (ví dụ sau khi AddtoComposite) và không yêu cầu bất kỳ mã hóa bổ sung nào.

Nếu bạn sử dụng OLE để đọc các thiết lập bộ lọc Phân tích (chẳng hạn như số danh sách theo dõi), như sau:

AB = CreateObject("Broker.Application"); // TRÁNH ĐIỀU NÀY AA = AB.Analysis; // TRÁNH ĐIỀU NÀY
wlnum = AA.Filter(0, "watchlist"); // TRÁNH ĐIỀU NÀY

bạn nên thay thế các lệnh gọi OLE bằng lệnh gọi đơn giản, gốc đến GetOption cho phép đọc các thiết lập bộ lọc công thức phân tích theo cách thân thiện với đa luồng. Ví dụ để đọc số danh sách theo dõi Filter Include, hãy sử dụng:

wlnum = GetOption("FilterIncludeWatchlist"); // CÁCH THÍCH HỢP

Để biết thêm thông tin về các trường thiết lập bộ lọc được hỗ trợ, hãy xem trang tham chiếu hàm GetOption.

Ngoài ra, lưu ý rằng đối tượng OLE AB.Analysis luôn tham chiếu đến cửa sổ phân tích tự động CŨ. Điều này có tác dụng phụ là khởi chạy/hiển thị phân tích tự động cũ bất cứ khi nào bạn sử dụng AB.Analysis trong mã của mình. Như đã giải thích ở trên, tất cả các lệnh gọi đến OLE phải được xóa khỏi các công thức của bạn nếu bạn muốn chạy trong cửa sổ Phân tích đa luồng mới. Chỉ được phép truy cập Phân tích mới qua OLE từ các chương trình/tập lệnh bên ngoài. Để truy cập Phân tích mới từ chương trình bên ngoài, bạn cần sử dụng các đối tượng AnalysisDocs/AnalysisDoc như được mô tả trong tài liệu giao diện OLE Automation.

2. Giảm thiểu việc sử dụng AddToComposite / Foreign

Bất kỳ quyền truy cập nào vào ký hiệu khác ngoài ký hiệu "hiện tại" từ cấp công thức đều liên quan đến khóa toàn cục (phần quan trọng) và do đó có thể ảnh hưởng đến hiệu suất. Vì

lý do này, nên giảm việc sử dụng các hàm AddToComposite/Foreign và sử dụng các biến tĩnh bất cứ khi nào có thể

3. Sử dụng hiệu quả và đúng các biến tĩnh

Quyền truy cập vào các biến tĩnh nhanh, an toàn cho luồng và nguyên tử ở cấp lệnh gọi StaticVarSet/StaticVarGet duy nhất. Điều đó có nghĩa là nó đọc/ghi toàn bộ mảng theo cách nguyên tử, do đó không có luồng nào khác sẽ đọc/ghi mảng đó ở giữa luồng khác đang cập nhật nó.

Tuy nhiên, bạn phải cẩn thận nếu bạn viết nhiều biến tĩnh cùng một lúc. Nói chung, khi bạn viết các biến tĩnh như một phần của quét/khám phá/kiểm tra ngược, tối ưu hóa Phân tích nhiều ký hiệu, bạn nên thực hiện việc viết (StaticVarSet) ở bước đầu tiên bằng cách sử dụng Status("stocknum")==0 như mô tả bên dưới. Đây là cách thực hiện được khuyến nghị:

```
if( Status("stocknum") == 0 )
{
// thực hiện tất cả các thao tác ghi/khởi tạo biến tĩnh tại đây
}
```

Thực hiện tất cả các thao tác khởi tạo/ghi vào biến tĩnh theo cách đó sẽ mang lại hiệu suất tốt nhất và các lần đọc tiếp theo (StaticVarGet) hoàn toàn an toàn và nhanh chóng. Bạn nên tránh làm mọi thứ trở nên phức tạp khi có thể tuân theo quy tắc đơn giản và hiệu quả của một trình ghi - nhiều trình đọc. Miễn là chỉ có một luồng ghi và nhiều luồng chỉ đọc các biến tĩnh, bạn sẽ an toàn và không cần phải lo lắng về đồng bộ hóa.

Chỉ dành cho những người viết công thức nâng cao:
Nếu vì lý do nào đó, bạn cần viết nhiều biến tĩnh được chia sẻ và truy cập từ nhiều luồng cùng một lúc và khi bạn phải đảm bảo rằng tất cả các bản cập nhật đều là nguyên tử, thì bạn cần bảo vệ các vùng công thức của mình cập nhật nhiều biến tĩnh bằng một semaphore hoặc phần quan trọng. Để có hiệu suất tốt nhất, bạn nên nhóm tất cả các lệnh đọc/ghi vào một phần như sau:

```
if( _TryEnterCS( "mysemaphore" ) ) // xem hàm StaticVarCompareExchange để triển khai
{
// bây giờ bạn đang ở trong phần quan trọng
// thực hiện tất cả các lệnh ghi/đọc biến tĩnh tại đây - không có luồng nào khác can thiệp vào đây
_LeaveCS();
}
else
{
_TRACE("Không thể nhập CS");
}
```

Việc triển khai cả semaphore và phần quan trọng trong AFL được thể hiện trong các ví dụ về hàm StaticVarCompareExchange.

4. Triển khai xử lý trước/khởi tạo trong cửa sổ Phân tích

Đôi khi cần phải thực hiện một số khởi tạo hoặc một số tính toán tốn thời gian trước khi hoàn thành tất cả các công việc khác. Để cho phép xử lý mà không có luồng khác can thiệp vào kết quả, bạn có thể sử dụng mệnh đề if sau:

```
if( Status("stocknum") == 0 )
{
// mã khởi tạo / tiền xử lý
}
```

AmiBroker phát hiện ra câu lệnh như vậy và chỉ chạy ký hiệu đầu tiên trong một luồng, đợi hoàn tất và chỉ sau khi hoàn tất, nó mới khởi chạy tất cả các luồng khác. Điều này cho phép những thứ như thiết lập các biến tĩnh để sử dụng trong quá trình xử lý tiếp theo, v.v. Lưu ý: câu lệnh trên KHÔNG được đặt bên trong #include.

5. Truy cập ký hiệu ~~~Equity

Sử dụng Foreign("~~~Equity", "C") chỉ có ý nghĩa để hiển thị biểu đồ về vốn chủ sở hữu của backtest đã hoàn tất. Điều quan trọng là phải hiểu rằng cửa sổ Phân tích mới hỗ trợ nhiều phiên bản và do đó, nó không thể sử dụng bất kỳ ký hiệu vốn chủ sở hữu nào được chia sẻ, vì nếu có, nhiều backtest đang chạy sẽ gây nhiễu lẫn nhau. Vì vậy, New Analysis có phiên bản cục bộ, riêng tư của tất cả dữ liệu vốn chủ sở hữu được sử dụng trong quá trình kiểm tra ngược và chỉ SAU KHI quá trình kiểm tra ngược hoàn tất, nó mới sao chép dữ liệu vốn chủ sở hữu đã sẵn sàng sử dụng vào biểu tượng ~~~Equity. Điều này có nghĩa là nếu bạn gọi Foreign("~~~Equity", "C") từ bên trong công thức hiện đang được kiểm tra ngược, bạn sẽ nhận được vốn chủ sở hữu kiểm tra ngược trước đó, không phải vốn chủ sở hữu hiện tại.

Để truy cập vốn chủ sở hữu hiện tại, bạn cần sử dụng giao diện backtester tùy chỉnh. Giao diện này có thuộc tính "Equity" trong đối tượng backtester chứa vốn chủ sở hữu tài khoản hiện tại. Nếu bạn cần vốn chủ sở hữu dưới dạng một mảng, có hai lựa chọn, hoặc thu thập giá trị theo cách này:

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

if( Status("action") == actionPortfolio )
{
bo = GetBacktesterObject(); bo.PreProcess(); // Khởi tạo backtester

PortEquity = Null; // sẽ giữ nguyên giá trị vốn chủ sở hữu danh mục đầu tư

for(bar=0; bar < BarCount; bar++)
{
bo.ProcessTradeSignals( bar);

// lưu trữ giá trị vốn chủ sở hữu hiện tại vào phần tử mảng PortEquity[ i ] = bo.Equity;
```

```
}

bo.PostProcess(); // Hoàn tất backtester

// TẠI THỜI ĐIỂM NÀY, PortEquity của BẠN chứa MẢNG giá trị vốn chủ sở hữu

}
```

Hoặc bạn có thể sử dụng thuộc tính EquityArray được thêm vào đối tượng Backtester trong v5.50.1

```
if( Status("action") == actionPortfolio )
{
bo = GetBacktesterObject(); bo.Backtest();
AddToComposite( bo.EquityArray, // lấy mảng vốn chủ sở hữu danh mục đầu tư trong một lệnh gọi "~~~MY_EQUITY_COPY", "X",
atcFlagDeleteValues | atcFlagEnableInPortfolio );
}
```

Xin lưu ý rằng các giá trị được điền trong quá trình kiểm tra ngược và tất cả các giá trị chỉ hợp lệ sau khi kiểm tra ngược hoàn tất (như trong ví dụ trên). Nếu bạn gọi nó ở giữa quá trình kiểm tra ngược, nó sẽ chỉ chứa giá trị vốn chủ sở hữu lên đến thanh nhất định. Tránh lạm dụng chức năng này và nó tốn kém về RAM/CPU (tuy nhiên, nó ít tốn kém hơn Foreign).

Cả hai cách được trình bày đều sẽ truy cập bản sao cục bộ, hiện tại của giá trị vốn chủ sở hữu trong Phân tích mới (không giống như Foreign truy cập các giá trị ký hiệu toàn cục từ kiểm tra ngược trước đó)

Các hoạt động ký hiệu đơn chạy trong một luồng

Như đã giải thích ở đầu bài viết, bất kỳ hoạt động nào như quét, khám phá, kiểm tra ngược, tối ưu hóa hoặc kiểm tra tiến trình được thực hiện trên một ký hiệu chỉ có thể sử dụng một luồng. Vì lý do đó, hầu như không có lợi thế về tốc độ so với việc chạy cùng một mã trong các phiên bản cũ của AmiBroker.

Cập nhật kể từ 5.70: Phiên bản này có chức năng "Tối ưu hóa riêng lẻ" mới cho phép chạy tối ưu hóa ký hiệu đơn bằng nhiều luồng, mặc dù có một số hạn chế: chỉ hỗ trợ tối ưu hóa toàn diện và không hỗ trợ trình kiểm tra ngược tùy chỉnh. Có hai lý do: a) công cụ tối ưu hóa thông minh cần kết quả của bước trước để quyết định chọn kết hợp tham số nào cho bước tiếp theo; b) giai đoạn thứ hai của backtest sẽ trao đổi với UI và OLE (backtester tùy chỉnh) và do đó không thể chạy từ luồng không phải UI (xem chi tiết bên dưới).

Backtest cá nhân chỉ có thể chạy trong một luồng

Điều quan trọng nhất cần hiểu là backtest cá nhân là backtest cấp danh mục đầu tư được chạy trên CHỈ MỘT ký hiệu. Ngay cả khi bạn chạy nó trên danh sách theo dõi, nó vẫn thực hiện mọi thứ theo trình tự, backtest đơn lẻ trên một

ký hiệu cùng một lúc, sau đó chuyển sang ký hiệu tiếp theo trong danh sách theo dõi. Lý do tại sao lại như vậy được mô tả bên dưới.

Cả backtest cấp danh mục đầu tư và backtest cá nhân đều bao gồm hai giai đoạn giống nhau
I. chạy công thức của bạn và thu thập tín hiệu
II. backtest thực tế có thể liên quan đến lần chạy thứ hai của công thức của bạn (backtester tùy chỉnh)

Giai đoạn I chạy công thức trên mỗi ký hiệu trong danh sách và có thể chạy đa luồng (nếu có nhiều hơn một ký hiệu trong danh sách).

Giai đoạn II xử lý các tín hiệu thu thập được trong giai đoạn I, tạo báo cáo và hiển thị kết quả chỉ được thực hiện một lần cho mỗi lần kiểm tra ngược.
Không thể đa luồng vì:
a) nó giao tiếp với Giao diện người dùng (UI)
b) nó sử dụng OLE/COM để cho phép bạn chạy trình kiểm tra ngược tùy chỉnh.

Cả OLE và UI + quyền truy cập đều không thể thực hiện từ luồng công nhân (không phải giao diện người dùng). Tệ hơn nữa, OLE/UI + đa luồng đồng nghĩa với cái chết, hãy xem: http://blogs.msdn.com/b/oldnewthing/archive/2008/04/24/8420242.aspx

Thông thường, trong trường hợp danh mục đầu tư nhiều ký hiệu, Giai đoạn I mất 95% thời gian cần thiết để chạy kiểm tra ngược danh mục đầu tư, vì vậy khi bạn chạy Giai đoạn I trong nhiều luồng, bạn sẽ có khả năng mở rộng rất tốt vì chỉ có 5% là không đa luồng.

Vì backtest cá nhân chạy trên MỘT ký hiệu nên pha duy nhất có thể chạy trong nhiều luồng, tức là pha 1 - chỉ bao gồm một lần chạy và do đó được chạy trong một luồng.

Để có thể chạy Phase II từ nhiều luồng, bạn sẽ KHÔNG thể giao tiếp với UI và KHÔNG thể sử dụng COM/OLE (không có backtester tùy chỉnh).

Điều đó khiến cho Backtest cá nhân KHÔNG THỂ nhanh hơn trong Automatic Analysis cũ.

Tính toán và kỳ vọng hợp lý

Một số người dùng sống trong thế giới tưởng tượng và nghĩ rằng họ có thể ném dữ liệu 100GB và dữ liệu sẽ được xử lý nhanh vì "họ có phần cứng mới nhất". Điều này hoàn toàn sai. Những gì bạn sẽ nhận được là sự cố. Mặc dù Windows 64 bit xóa bỏ rào cản không gian địa chỉ ảo 2GB cho mỗi ứng dụng, nhưng không đúng khi nói rằng không còn giới hạn nào nữa.
Thật không may, ngay cả những người có nền tảng kỹ thuật cũng quên tính toán cơ bản và có một số kỳ vọng không hợp lý. Điều đầu tiên và quan trọng nhất mà mọi người bỏ qua là

sự khác biệt lớn giữa tốc độ truy cập được tạo ra bởi kích thước dữ liệu. Thuật ngữ "Bộ nhớ truy cập ngẫu nhiên" trong quá khứ (như vào năm 1990) có nghĩa là việc truy cập dữ liệu mất cùng một khoảng thời gian, bất kể vị trí. Điều đó KHÔNG CÒN đúng nữa. Có sự khác biệt lớn về tốc độ truy cập tùy thuộc vào vị trí dữ liệu. Ví dụ, Intel i7 920, cấu hình ba kênh truy cập dữ liệu được lưu trong bộ nhớ đệm L1 với tốc độ 52 GB/giây, dữ liệu được lưu trong bộ nhớ đệm L2 là 30 GB/giây (chậm hơn 2 lần!), dữ liệu được lưu trong bộ nhớ đệm L3 là 24 GB/giây và RAM thông thường là 11 GB/giây. Điều đó có nghĩa là truy cập dữ liệu được lưu trong bộ nhớ đệm nhanh hơn 5 lần so với truy cập RAM. Mọi thứ trở nên kịch tính hơn nữa nếu bạn hết RAM và hệ thống phải chuyển sang đĩa. Với hầu hết các ổ SSD hiện đại, chúng ta chỉ nói về 200 MB/giây (0,2 GB/giây). Chậm hơn RAM hai bậc (100 lần) và chậm hơn bộ nhớ đệm ba bậc. Giả sử độ trễ bằng không (tìm kiếm). Trong thế giới thực, truy cập đĩa có thể chậm hơn RAM 10000 lần.
Bây giờ hãy tự giúp mình và tính toán. Chia 100GB cho tốc độ đĩa SSD 0,2GB/giây. Bạn sẽ nhận được gì? 500 giây - gần mười phút chỉ để đọc dữ liệu. Bây giờ bạn có biết rằng nếu ứng dụng không xử lý tin nhắn chỉ trong 1 giây được Windows coi là "không phản hồi" không? Điều đó có nghĩa là gì? Điều đó có nghĩa là ngay cả trong thế giới 64 bit, bất kỳ ứng dụng Windows nào cũng sẽ gặp sự cố khi xử lý các tập dữ liệu vượt quá 5GB chỉ vì tốc độ đọc đĩa thô trong trường hợp tốt nhất không vượt quá 200MB/giây (thường tệ hơn nhiều). Cố gắng kiểm tra ngược lượng dữ liệu vô lý như vậy trên PC cao cấp sẽ chỉ dẫn đến sự cố, vì sẽ hết thời gian chờ, Windows sẽ gặp khó khăn khi xử lý tin nhắn và bạn sẽ tràn bộ đệm hệ thống. Và điều đó không liên quan gì đến phần mềm. Đây chỉ là bài học toán học tàn khốc mà một số người đã quên. Quy tắc đầu tiên và quan trọng nhất để có tốc độ nhanh hơn là giới hạn kích thước dữ liệu của bạn, để nó ít nhất vừa với RAM.

Chức năng xếp hạng

Xếp hạng là mối quan hệ giữa một tập hợp các mục sao cho, đối với bất kỳ hai mục nào, mục đầu tiên sẽ được 'xếp hạng cao hơn', 'xếp hạng thấp hơn' hoặc 'xếp hạng bằng' mục thứ hai. Cách đơn giản nhất để có được thứ hạng là sắp xếp các mục theo 'giá trị' hoặc 'điểm'. Ví dụ, bạn có thể lấy tỷ lệ thay đổi 100 thanh cho các ký hiệu - đó sẽ là 'điểm' hoặc 'giá trị' của mục. Sau đó, sắp xếp các kết quả theo nó để bạn sẽ có được danh sách ký hiệu trong đó ký hiệu đầu tiên có hiệu suất tốt nhất (tỷ lệ thay đổi cao nhất) và ký hiệu cuối cùng có hiệu suất kém nhất.
AmiBroker cho phép người dùng thực hiện/sử dụng ba loại xếp hạng khác nhau

1. sử dụng xếp hạng tín hiệu vào lệnh giao dịch (mua/bán khống) để quyết định lệnh nào được ưu tiên hơn các lệnh khác trong quá trình kiểm tra/tối ưu hóa danh mục đầu tư
2. hiển thị nhiều xếp hạng trong các bảng được tạo bằng Exploration
3. tạo xếp hạng số để sử dụng sau này (chức năng mục đích chung)

Loại xếp hạng đầu tiên được thực hiện tự động nếu công thức hệ thống giao dịch của bạn xác định biến PositionScore. Bạn có thể sử dụng biến PositionScore để quyết định lệnh giao dịch nào nên được nhập nếu có nhiều tín hiệu vào lệnh trên các chứng khoán khác nhau hơn số lượng tối đa cho phép của các vị thế mở hoặc quỹ khả dụng. Trong trường hợp đó, AmiBroker sẽ sử dụng giá trị tuyệt đối của biến PositionScore để quyết định lệnh giao dịch nào được ưu tiên. Để biết thông tin chi tiết về chức năng xếp hạng trong quá trình kiểm tra ngược, hãy xem hướng dẫn Portfolio Backtester.

Loại xếp hạng thứ hai chỉ đơn giản là gán một số (xếp hạng) cho dòng đầu ra của lệnh khám phá. Cột xếp hạng được thêm vào đầu ra của lệnh khám phá chỉ bằng cách gọi hàm AddRankColumn sau khi thực hiện sắp xếp bằng hàm SetSortColumns. Bạn có thể gọi SetSortColumns nhiều lần và bạn có thể gọi AddRankColumn nhiều lần để đạt được nhiều thứ hạng khác nhau dựa trên nhiều cột. Xem ví dụ bên dưới:

```
Filter = 1;
AddColumn( Close, "Close" ); AddColumn( Volume, "BI" ); AddSummaryRows( 31 + 32, 1.5 );

AddRankColumn(); // không cần sắp xếp trước AddRankColumn chỉ cần thêm số dòng
SetSortColumns( -4 );
AddRankColumn(); // xếp hạng theo cột thứ 4 (giảm dần) SetSortColumns( -3 );
AddRankColumn(); // xếp hạng theo cột thứ 3 (tăng dần)
```

Loại xếp hạng thứ ba là xếp hạng theo từng thanh, mục đích chung được thực hiện bằng cách sử dụng các biến tĩnh. Loại này ngốn nhiều tài nguyên nhất (tốn nhiều tài nguyên tính toán) nhưng cũng cung cấp nhiều khả năng nhất.

Nhìn chung, quy trình này bao gồm việc tạo các biến tĩnh có giá trị để sử dụng cho việc sắp xếp/xếp hạng, tức là "điểm" và sau đó gọi một hàm đặc biệt (StaticVarGenerateRanks) để tạo ra một tập hợp các biến tĩnh mới chứa thứ hạng đã tính toán.

LƯU Ý: Hàm này KHÔNG nhằm mục đích thay thế thứ hạng tích hợp của bakctester thông qua PositionScore. Ngược lại: bất cứ khi nào có thể, bạn nên sử dụng PositionScore vì đây

là cách nhanh hơn nhiều và ít tốn bộ nhớ hơn để thực hiện các bài kiểm tra ngược với thứ hạng.

StaticVarGenerateRanks thường được dùng cho các tác vụ KHÁC ngoài việc kiểm tra ngược như khám phá hoặc chỉ báo có thể yêu cầu chức năng xếp hạng, nhưng tất nhiên nó cũng có thể được dùng để kiểm tra ngược khi/nơi mà riêng PositionScore không cho phép triển khai những gì bạn cần trong hệ thống giao dịch của mình.

CẢNH BÁO: hàm này tốn nhiều bộ nhớ và tính toán. Mất khoảng 20ms cho mỗi 15K thanh và 7 ký hiệu. Hãy thử gọi hàm này chỉ một lần cho mỗi lần quét/khám phá/kiểm tra ngược bằng cách sử dụng if(Status("stocknum")==0) hoặc tốt hơn nữa, hãy sử dụng lệnh quét riêng chỉ một lần để tính toán trước thứ hạng và sử dụng sau (như quét tạo hợp thành). Nếu bạn không làm như vậy và gọi StaticVarGenerateRanks cho mọi ký hiệu, hiệu suất sẽ giảm đáng kể vì hàm này không chỉ cần nhiều thời gian để tính toán mà còn phải khóa quyền truy cập vào bộ nhớ dùng chung được các biến tĩnh sử dụng để các luồng khác cố gắng truy cập các biến tĩnh sẽ phải đợi cho đến khi hàm này hoàn tất.

Hàm StaticVarGenerateRanks

StaticVarGenarateRanks("outputprefix", "inputprefix", topranks, tiemode) là một thành phần cốt lõi của hệ thống xếp hạng mục đích chung. Nó có 4 tham số: "outputprefix" - tiền tố được thêm vào các biến tĩnh đầu ra giữ thứ hạng, "inputprefix" là tiền tố của các biến tĩnh giữ điểm (đầu vào), topranks - xác định số lượng ký hiệu xếp hạng trên/dưới nên được đưa vào bộ xếp hạng được tạo và tiemode xác định cách giải quyết các mối quan hệ (xếp hạng bằng nhau).

"inputprefix" là tiền tố xác định tên của các biến tĩnh sẽ được sử dụng làm đầu vào để xếp hạng. AmiBroker sẽ tìm kiếm tất cả các biến tĩnh bắt đầu bằng tiền tố đó và giả sử phần còn lại của tên biến là ký hiệu cổ phiếu. Giả sử bạn muốn xếp hạng cổ phiếu theo ROC (tỷ lệ thay đổi). Tất cả những gì bạn cần làm là lưu trữ các giá trị vào các biến tĩnh. Giả sử chúng ta sẽ sử dụng các tên biến tĩnh như "ItemScoreAPPL", "ItemScoreMSFT", v.v.

Để điền các biến tĩnh đầu vào, bạn có thể sử dụng vòng lặp này:

```
for ( i = 0; ( sym = StrExtract( symlist, i ) ) != ""; i++ )
{
SetForeign( sym ); Value = ROC( C, 10 );
RestorePriceArrays();
StaticVarSet( "ItemScore" + sym, Value );
}
```

Bây giờ bạn đã sẵn sàng để thực hiện sắp xếp/xếp hạng. Có hai chế độ, chế độ xếp hạng thông thường và chế độ Xếp hạng trên cùng/dưới cùng. Chế độ xếp hạng thông thường được thực hiện khi đối số toprank được đặt thành 0.

```
StaticVarGenerateRanks( "rank", "ItemScore", 0, 1224 );
```

Trong trường hợp này, lệnh gọi StaticVarGenerateRanks sẽ tạo ra một tập hợp các biến tĩnh bắt đầu bằng tiền tố được xác định bởi đối số thứ 2, mỗi biến giữ thứ hạng của một ký hiệu cụ thể, vì vậy trong trường hợp này RankItemScoreMSFT sẽ giữ thứ hạng của MSFT, RankItemScoreAAPL sẽ giữ thứ hạng của AAPL. Lưu ý rằng trong AmiBroker, số hạng bắt đầu từ MỘT.

Đối số thứ ba (topranks) bằng không ở chế độ xếp hạng thông thường. Đối số thứ tư (tiemode) xác định cách xếp hạng các cặp. Các chế độ được hỗ trợ là 1234 và 1224. Ở chế độ 1224, các cặp được đánh số với thứ hạng bằng nhau.

Mã ví dụ cho chế độ xếp hạng thông thường (mọi thứ được thực hiện đều được thực hiện trong một lần, có thể được sử dụng trong chỉ báo):

```
symlist = "C,CAT,DD,GE,IBM,INTC,MSFT";

// xóa các biến tĩnh StaticVarRemove( "ItemScore*" );

// điền vào các mảng tĩnh đầu vào

for ( i = 0; ( sym = StrExtract( symlist, i ) ) != ""; i++ )
{
SetForeign( sym ); Value = ROC( C, 10 );
RestorePriceArrays();
StaticVarSet( "ItemScore" + sym, Value );

}

// thực hiện xếp hạng
StaticVarGenerateRanks( "rank", "ItemScore", 0, 1224 ); // chế độ xếp hạng bình thường

// đọc xếp hạng
for ( i = 0; ( sym = StrExtract( symlist, i ) ) != ""; i++ )
{
Plot( StaticVarGet( "RankItemScore" + sym ), sym, colorCustom10 + i );
}
```

Chế độ xếp hạng trên/dưới (tạo ra các bảng xếp hạng trên/dưới chứa các chỉ mục cho các giá trị xếp hạng trên. Khi topranks > 0 các giá trị xếp hạng trên được sử dụng, khi topranks < 0 thì các giá trị xếp hạng dưới được sử dụng. Các giá trị được lưu trữ trong các biến có định dạng:
OutputprefixInputprefixN trong đó N là số 1, 2, 3 biểu thị thứ hạng trên/dưới. Giả sử tham số OutputPrefix là "Top" và tham số Inputprefix là ROC. Trong trường hợp đó, biến TopROC1 sẽ giữ chỉ mục của giá trị được xếp hạng cao nhất. TopROC2 sẽ giữ giá trị được xếp hạng cao thứ hai, v.v.
Hàm StaticVarGenerateRanks sử dụng đánh số thứ hạng bắt đầu từ MỘT. Ở chế độ xếp hạng cao nhất, StaticVarGenerateRanks cũng sẽ chuẩn bị biến tĩnh chứa danh sách các tên biến được phân tách bằng dấu phẩy có thể được sử dụng để tìm ra chỉ mục nào tham chiếu

đến ký hiệu nào. Vì vậy, nếu TopROC1 giữ 1, bạn sẽ tra cứu chuỗi con đầu tiên trong biến TopROCSymbols để tìm ra biến (ký hiệu) nào được xếp hạng tại top.
Ngoài ra StaticVarGetRankedSymbols cung cấp phương pháp dễ sử dụng để lấy danh sách các ký hiệu được xếp hạng được phân tách bằng dấu phẩy cho ngày giờ cụ thể.

Mã ví dụ cho chế độ xếp hạng hàng đầu:

```
symlist = "C,CAT,DD,GE,IBM,INTC,MSFT";

// xóa các biến tĩnh StaticVarRemove( "ItemScore*" );

// điền các mảng tĩnh đầu vào

for ( i = 0; ( sym = StrExtract( symlist, i ) ) != ""; i++ )
{
SetForeign( sym ); Value = ROC( C, 10 );
RestorePriceArrays();
StaticVarSet( "ItemScore" + sym, Value );

}

// thực hiện xếp hạng
StaticVarGenerateRanks( "hạng", "ItemScore", 0, 1224 ); // chế độ xếp hạng bình thường
StaticVarGenerateRanks( "top", "ItemScore", 3, 1224 ); // chế độ top-N
StaticVarGenerateRanks( "bot", "ItemScore", -3, 1224 ); // chế độ bottom-N
// đọc xếp hạng
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 ) ;
```

Cách sử dụng StaticVarGenerateRanks trong cửa sổ Phân tích

Vì xếp hạng là quy trình ngốn nhiều tài nguyên, nên chỉ nên thực hiện một lần cho mỗi lần chạy Phân tích, không phải cho mọi ký hiệu. Bạn có thể thực hiện bằng cách chạy công thức tạo xếp hạng riêng một lần bằng tay trước khi chạy Phân tích hoặc sử dụng câu lệnh Status("stocknum") == 0 để đảm bảo rằng quy trình xếp hạng chỉ được thực hiện cho ký hiệu đầu tiên trong danh sách theo dõi đang phân tích.

Sau đây là một ví dụ về mã để khám phá danh sách theo dõi hiện đang hoạt động hoặc danh sách tất cả các ký hiệu và thực hiện xếp hạng

```
if ( GetOption( "ApplyTo" ) == 2 )
{

}
else

wlnum = GetOption( "FilterIncludeWatchlist" );
List = CategoryGetSymbols( categoryWatchlist, wlnum ) ;

if ( GetOption( "ApplyTo" ) == 0 )
{

}
else
{

" );
}

List = CategoryGetSymbols( categoryAll, 0 );

Error( "Công thức hoạt động tốt nếu cài đặt ApplyTo của bạn là 'Filter' hoặc 'All'

if ( Status("stocknum") == 0 ) // TẠO XẾP HẠNG KHI CHÚNG TA ĐANG Ở KÝ HIỆU ĐẦU TIÊN

{
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( "giá trị" + symbol ); rank = StaticVarGet( "giá trị rankvalues" + symbol
);

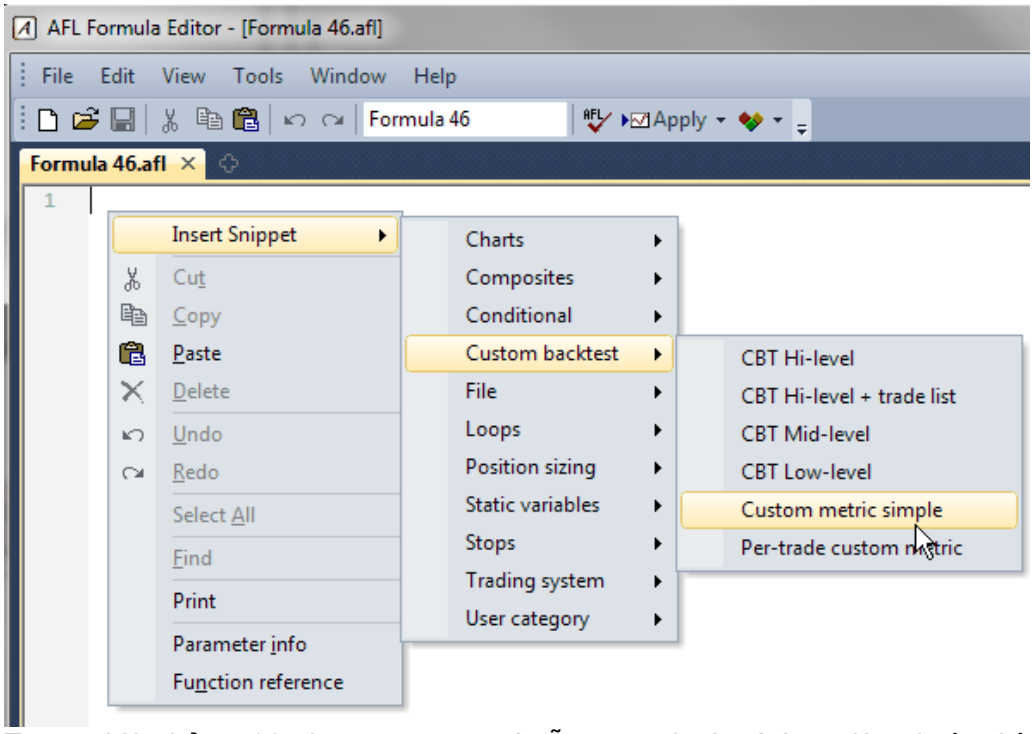

AddColumn( giá trị, "giá trị" ); AddColumn( thứ hạng, "thứ hạng" );


Filter = 1;
SetSortColumns( 2, 4 );
```

Sử dụng đoạn mã AFL

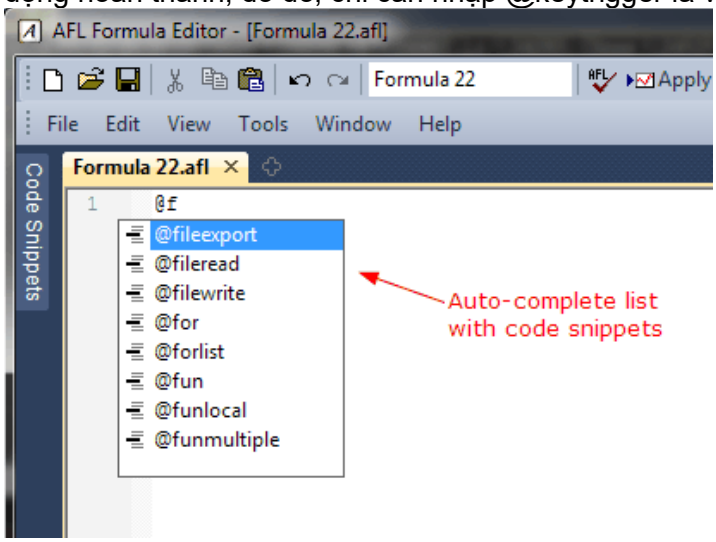
Đoạn mã là một đoạn mã AFL nhỏ có thể tái sử dụng. Có thể chèn đoạn mã bằng cách

• nhấp chuột phải vào cửa sổ trình soạn thảo AFL và chọn menu "Chèn đoạn mã" hoặc
• kéo đoạn mã từ cửa sổ Đoạn mã hoặc
• nhập lệnh kích hoạt bàn phím (như @for) trong trình soạn thảo



Trong phiên bản 5.90, đoạn mã cũng có sẵn trong danh sách tự động hoàn thành trong Trình soạn thảo AFL. Chỉ cần nhập @ cộng với chữ cái đầu tiên của lệnh kích hoạt phím đoạn mã và danh sách tự động hoàn thành sẽ hiển thị cho bạn danh sách các đoạn mã khả dụng có lệnh kích hoạt bàn phím được xác định bắt đầu bằng chữ cái đó.

Việc thay thế lệnh kích hoạt bàn phím vẫn hoạt động ngay cả khi không kích hoạt lệnh tự động hoàn thành, do đó, chỉ cần nhập @keytrigger là văn bản đoạn mã sẽ được thay thế.

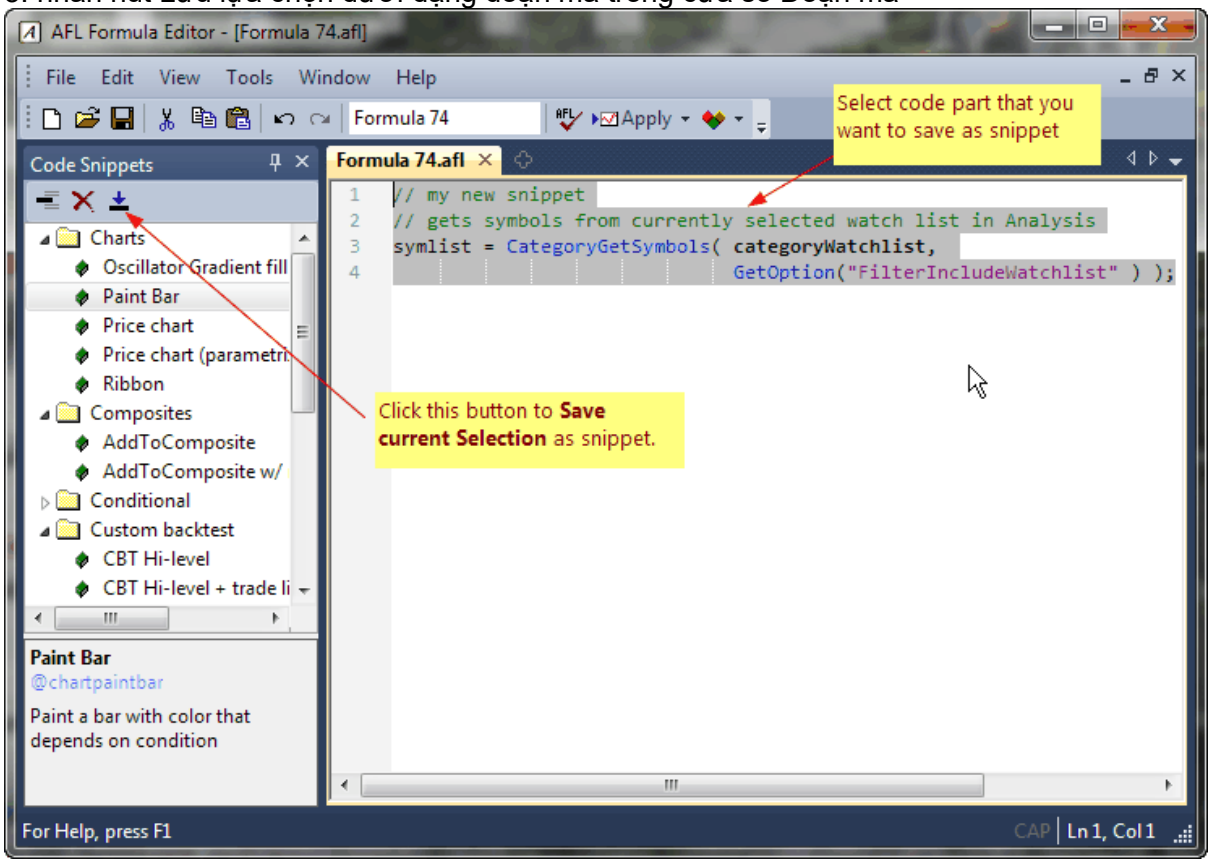


ĐỊNH NGHĨA CÁC ĐOẠN MỚI CỦA RIÊNG BẠN

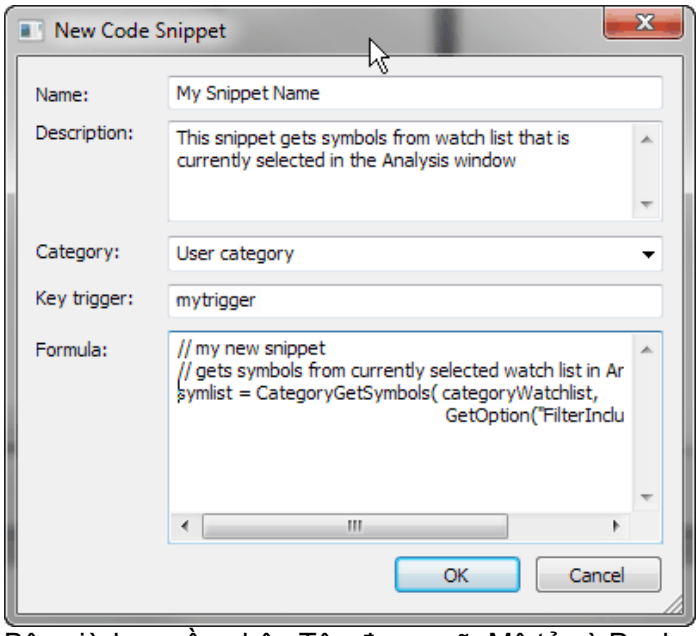
Bạn có thể thêm các đoạn mã của riêng mình khá dễ dàng bằng cách sử dụng cửa sổ Đoạn mã mới. Cửa sổ Đoạn mã có sẵn trong trình soạn thảo AFL mới. Có thể hiển thị/ẩn đoạn mã bằng menu Window.

Để tạo đoạn mã của riêng bạn, hãy làm như sau:

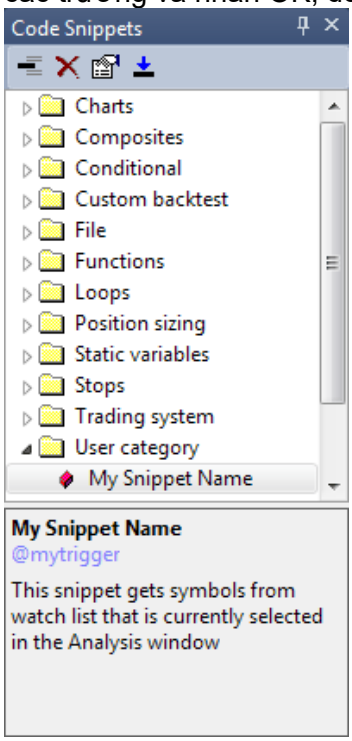
1. nhập mã bạn muốn
2. chọn (đánh dấu) mã bạn muốn đặt trong đoạn mã
3. nhấn nút Lưu lựa chọn dưới dạng đoạn mã trong cửa sổ Đoạn mã



Nếu bạn thực hiện các bước trên, hộp thoại sau sẽ xuất hiện:



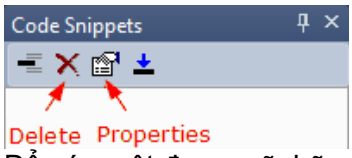
Bây giờ bạn cần nhập Tên đoạn mã, Mô tả và Danh mục. Danh mục có thể được chọn từ các mục đã tồn tại (sử dụng hộp thả xuống) hoặc có thể nhập tên danh mục mới vào trường danh mục. Trường kích hoạt khóa là tùy chọn và chứa kích hoạt tự động hoàn thành đoạn mã (được mô tả ở trên). Trường Công thức là chính mã đoạn mã. Sau khi bạn nhập tất cả các trường và nhấn OK, đoạn mã mới của bạn sẽ xuất hiện trong danh sách.



Từ đó trở đi, bạn có thể sử dụng đoạn mã của riêng mình theo cùng cách như các đoạn mã hiện có. Có lẽ phương pháp thuận tiện nhất là sử dụng kéo-thả từ danh sách vào trình soạn thảo AFL.

Như bạn có thể thấy, các đoạn mã do người dùng xác định được đánh dấu bằng hộp màu đỏ trong danh sách Đoạn mã. Chỉ có thể ghi đè và/hoặc xóa các đoạn mã do người dùng xác định.

Để chỉnh sửa đoạn mã do người dùng xác định hiện có, bạn có thể làm theo các bước trên và đặt tên hiện có. AmiBroker sẽ hỏi bạn có muốn ghi đè đoạn mã hiện có không hoặc bạn chỉ cần nhấp vào nút Thuộc tính và chỉnh sửa trực tiếp đoạn mã mà không cần chèn lại.



Để xóa một đoạn mã, hãy chọn đoạn mã bạn muốn xóa khỏi danh sách và nhấn nút Xóa (X) trong cửa sổ Đoạn mã.

THÔNG TIN KỸ THUẬT (chỉ dành cho người dùng nâng cao)

Có hai tệp nằm trong thư mục AmiBroker chứa các đoạn mã:
CodeSnippets.xml - đây là các đoạn mã được cung cấp cùng với bản cài đặt AmiBroker (và có thể được thay thế trong các bản cài đặt tiếp theo, vì vậy đừng sửa đổi nó!)
UserSnippets.xml - đây là các đoạn mã do người dùng định nghĩa. Tệp này KHÔNG có trong bản cài đặt và người dùng có thể tự tạo tệp này.

Sơ đồ XML cho tệp đoạn mã rất đơn giản (như bên dưới). Chức năng kích hoạt khóa CHƯA được triển khai,

tuy nhiên, các trường Keytrigger nên được đưa vào định nghĩa để sử dụng trong tương lai. Nó sẽ hoạt động giống như 'tự động hoàn thành' để bạn nhập phím tắt vào, nó sẽ mở rộng thành công thức.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<AmiBroker-CodeSnippets CompactMode="0">

<Snippet>
<Name>Đoạn trích đầu tiên</Name>
<Description>Mô tả của đoạn trích</Description>
<Category>Danh mục người dùng</Category>
<KeyTrigger>?trigger1</KeyTrigger>
<Formula>
<![CDATA[

// chính công thức

]]>
</Formula>
</Snippet>

<Snippet>
<Name>Đoạn trích thứ hai</Name>
<Description>Mô tả của đoạn trích</Description>
```

```
<Category>Danh mục người dùng</Category>
<KeyTrigger>?trigger2</KeyTrigger>
<Formula>
<![CDATA[


// công thức chính nó
]]>
</Formula>
</Snippet>
</AmiBroker-CodeSnippets>
```

Cách sử dụng trình gỡ lỗi trực quan AFL

Cơ bản

Để chạy công thức trong trình gỡ lỗi, chỉ cần chọn menu Debug->Go (phím F5) trong Trình chỉnh sửa công thức hoặc nhấp vào nút thanh công cụ được đánh dấu bên dưới là Bắt đầu/Tiếp tục.

Trong hầu hết các trường hợp, mã sẽ thực thi trong nháy mắt nên bạn thậm chí sẽ không thấy nó có hoạt động hay không. Bạn có thể dừng thực thi đủ lâu để xem điều gì đang xảy ra bằng cách đặt điểm dừng rồi tiến lên phía trước.

Điểm dừng

Để đặt điểm dừng, hãy di chuyển con trỏ đến dòng mà bạn muốn dừng thực thi và chọn Gỡ lỗi->Menu Chèn/Xóa Điểm dừng hoặc nhấn nút vòng tròn màu đỏ trên thanh công cụ hoặc nhấn phím F9. Điểm dừng sẽ được hiển thị dưới dạng vòng tròn màu đỏ ở lề trái.

Có thể thêm/xóa điểm dừng bất kỳ lúc nào, trước khi bạn bắt đầu gỡ lỗi và trong khi bạn đang gỡ lỗi. Chúng có thể được thêm/xóa khi trình gỡ lỗi dừng tại điểm dừng và trong khi mã đang chạy.

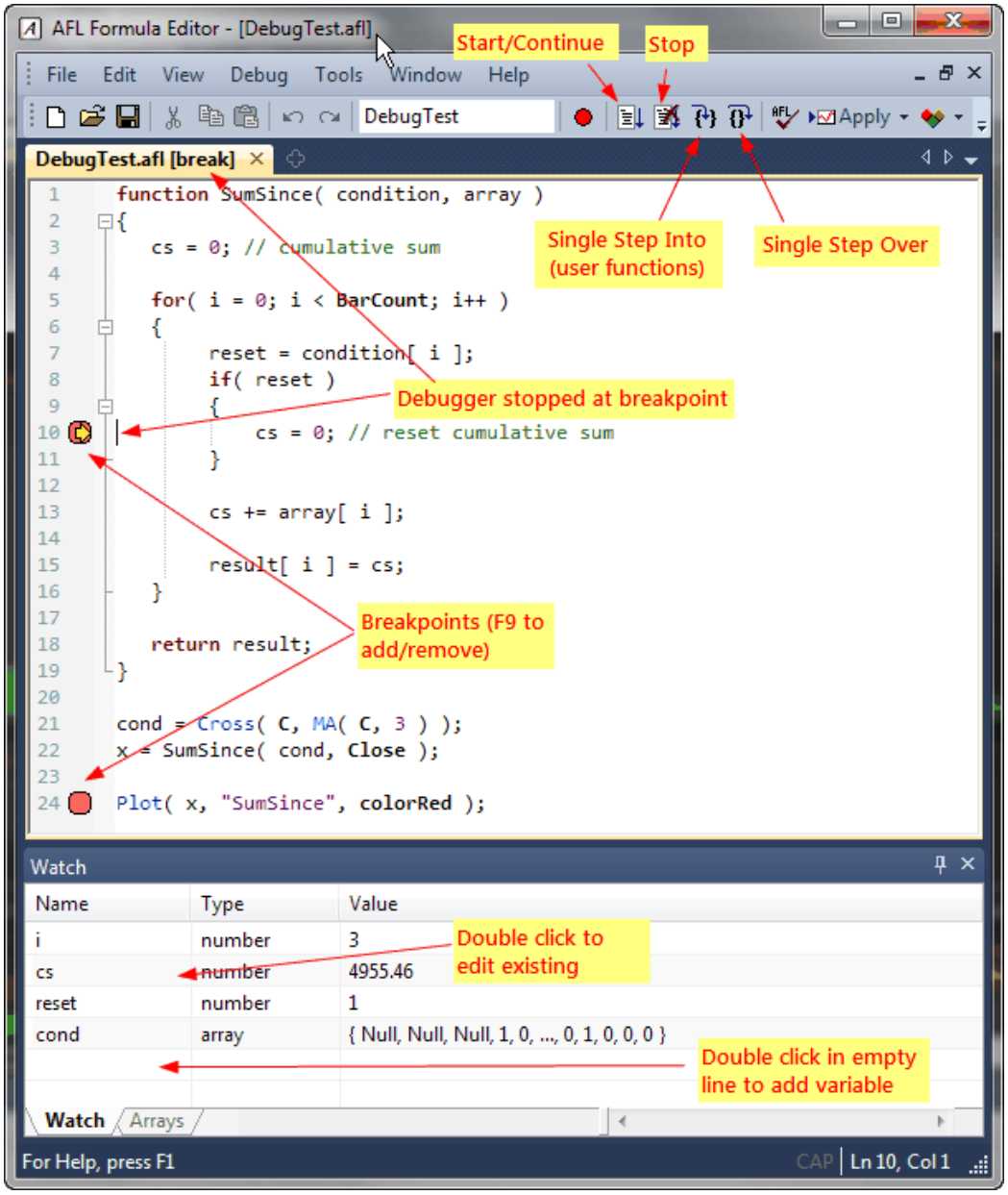
Bây giờ, hãy bắt đầu gỡ lỗi lại (Debug->Go). Quá trình thực thi sẽ dừng tại dòng được đánh dấu bằng dấu điểm dừng, trước khi mã của dòng này thực sự được thực thi. Một mũi tên màu vàng hiển thị vị trí thực thi hiện tại như trong hình bên dưới.

Bước đơn

Bạn có thể khiến mã thực thi bằng cách bước trước (Debug->Step Over, phím F10). Bạn cũng có thể nhấp vào tùy chọn Debug->Step Into (phím F11). Sự khác biệt giữa Step Into và Step Over là khi bạn cố gắng thực hiện từng bước một lệnh gọi đến hàm do người dùng định nghĩa.

Nếu dòng chứa lệnh gọi đến hàm do người dùng định nghĩa:

• Step Into thực thi chính lệnh gọi đó và dừng ở dòng mã đầu tiên bên trong hàm
• Step Over thực thi hàm được gọi như một bước mà không dừng bên trong hàm



Kiểm tra nội dung của các biến

Khi trình gỡ lỗi dừng tại điểm dừng hoặc dừng bằng cách thực hiện từng bước một, bạn có thể kiểm tra nội dung của các biến ở bất kỳ đâu trong mã. Để kiểm tra nhanh nội dung của bất kỳ biến nào, chỉ cần di chuột qua biến đó. Sau chưa đầy một giây, một chú giải công cụ sẽ xuất hiện hiển thị loại và giá trị của biến.

Cửa sổ Watch

Nếu bạn muốn theo dõi nhiều biến một cách dễ dàng, bạn có thể sử dụng cửa sổ Watch. Để mở cửa sổ Watch, hãy sử dụng menu Window->Watch.

Để thêm một biến mới vào cửa sổ Watch, hãy nhấp đúp vào hàng trống và nhập tên biến. Bạn cũng có thể chỉ cần chọn biến trong trình chỉnh sửa và kéo nó qua cửa sổ Watch.

Để chỉnh sửa biến hiện có trong cửa sổ Watch, hãy nhấp đúp vào tên của biến đó và nhập tên biến.

Để xóa biến khỏi cửa sổ Watch, hãy nhấp vào tên một lần (chọn tên đó) và nhấn phím DELETE.

Ngoài các biến, bạn có thể sử dụng toàn bộ biểu thức trong cửa sổ Watch. Bạn có thể kéo-thả biểu thức từ trình chỉnh sửa vào cửa sổ Watch hoặc nhập biểu thức sau khi nhấp đúp như bạn làm với các biến. Trình đánh giá biểu thức được cửa sổ Watch sử dụng hỗ trợ các mục sau:

• tất cả các toán tử số học như +, -, *, /
• dấu ngoặc đơn như (High+Low)/2
• biểu thức chỉ số mảng với các biến và biểu thức là chỉ số mảng như Close[i + 2]
• ma trận và biểu thức ma trận với chỉ số động, như mat[i + 1][j]
• kết hợp nhiều loại biến khác nhau như nối chuỗi và giá trị số như thế này "Giá trị là = " + Close[i + 2]

Lưu ý rằng chỉ các định danh có thể được sử dụng trong cửa sổ Watch mới là định danh biến. Bạn không thể sử dụng định danh hàm, do đó bạn không thể gọi hàm từ cửa sổ Watch.

Các giá trị được nhập vào cửa sổ Watch sẽ được theo dõi để thay đổi và nếu giá trị thay đổi giữa hai bước gỡ lỗi, giá trị đã cho sẽ được tô sáng bằng nền màu vàng. Các giá trị số (vô hướng) sẽ hiển thị màu xanh lá cây khi chúng tăng hoặc màu đỏ nếu chúng giảm.

Tab Mảng

Tab mảng trong cửa sổ Watch hiển thị đầu ra mảng giống như khám phá để kiểm tra chi tiết nội dung mảng (20 mảng đầu tiên từ cửa sổ watch được báo cáo). Điều này rất tiện lợi vì mảng thường lớn và việc có chúng ở định dạng bảng cùng với số thanh và dấu ngày/giờ tương ứng giúp dễ dàng xem nội dung của chúng hơn nhiều. Ngoài ra, mỗi phần tử của mỗi mảng được theo dõi để thay đổi và thay đổi mục riêng lẻ được đánh dấu và tự động cuộn xảy ra để mục thay đổi đầu tiên luôn hiển thị trong tab 'Mảng'.

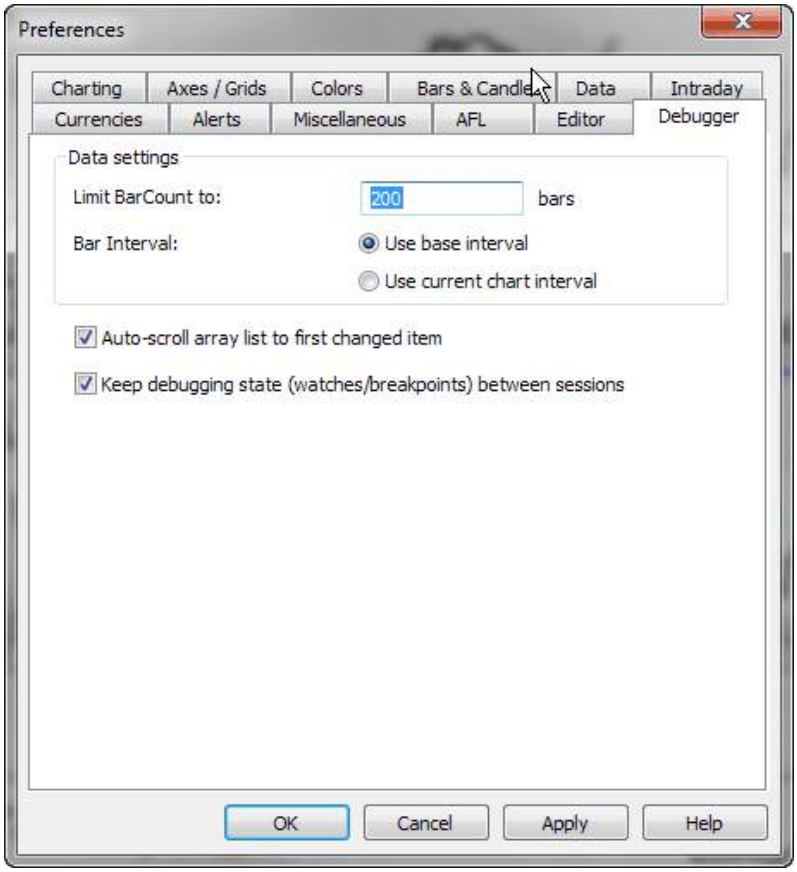
Cửa sổ đầu ra

Khi bạn chạy mã trong trình gỡ lỗi, bạn có thể sử dụng cửa sổ Đầu ra (mở bằng menu Cửa sổ->Đầu ra) để hiển thị tất cả các văn bản được in bằng hàm printf(). Khi công thức hoàn tất quá trình thực thi, văn bản "Phiên gỡ lỗi đã kết thúc" sẽ được hiển thị trong cửa sổ Đầu ra.

Tùy chọn trình gỡ lỗi

Cửa sổ Công cụ->Tùy chọn hiện có một trang chuyên dụng với các cài đặt Trình gỡ lỗi.

Các thiết lập như sau:



• Giới hạn BarCount thành - xác định số lượng thanh tối đa trong mảng (BarCount) được sử dụng trong quá trình gỡ lỗi (mặc định là 200 thanh)
• Khoảng thời gian thanh - quyết định xem trình gỡ lỗi sử dụng Khoảng thời gian cơ sở hay Khoảng thời gian biểu đồ hiện tại. Mặc định là "Sử dụng khoảng thời gian cơ sở" vì không thể mở biểu đồ nào cả!
• Tự động cuộn đến mục thay đổi đầu tiên - khi tùy chọn này BẬT, danh sách "Mảng" trong cửa sổ Theo dõi sẽ tự động cuộn đến mục mảng đầu tiên đã thay đổi (do đó, bạn không cần phải định vị các phần tử đã thay đổi theo cách thủ công)
• Giữ trạng thái gỡ lỗi - khi tùy chọn này BẬT, trình soạn thảo AFL sẽ lưu trạng thái gỡ lỗi (điểm dừng và theo dõi) và dấu trang trong tệp .dbg cùng với công thức khi đóng trình soạn thảo và khôi phục trạng thái khi công thức được mở lại. Lưu ý rằng tệp .dbg sẽ tự động bị AmiBroker xóa khi không có điểm dừng, không có dấu trang và không có dấu trang.

Phím tắt

Các phím tắt sau đây có sẵn trong chế độ gỡ lỗi:

• F5 - Bắt đầu/Tiếp tục gỡ lỗi
• F9 - Chuyển đổi Điểm dừng
• F10 - Bước qua
• F11 - Bước vào

Mẹo & thủ thuật

Nếu bạn muốn dừng thực thi khi điều kiện nhất định được đáp ứng, hãy đặt điểm dừng bên trong câu lệnh if như sau:

```
if( reset > 0 AND param > 4 )
{
cs = 0; // PUT BREAKPOINT IN THIS LINE và nó sẽ chỉ dừng khi điều kiện được đáp ứng
}
```

Điểm dừng hiện hoạt động với:

• các câu lệnh thông thường (kết thúc bằng dấu chấm phẩy). Đối với các câu lệnh nhiều dòng, hãy đặt điểm ngắt ở dòng đầu tiên của câu lệnh
• vòng lặp for
• vòng lặp while
• vòng lặp do-while (bạn cần đặt điểm ngắt tại vị trí mệnh đề 'while', nó sẽ không ngắt ở dòng 'do' vì về cơ bản nó không phải là lệnh, nếu bạn muốn ngắt ở đầu
của do, chỉ cần đặt điểm ngắt ở câu lệnh đầu tiên bên trong { block }
• câu lệnh if
• câu lệnh return
• câu lệnh switch/case
• câu lệnh break

Điểm ngắt mà bạn đặt trên các dòng khác sẽ không kích hoạt. Trình soạn thảo AFL sẽ không cho phép đặt điểm ngắt trên dòng trống hoặc dòng bắt đầu bằng // chú thích hoặc dấu ngoặc nhọn duy nhất

Bạn cũng cần lưu ý rằng điểm ngắt được đặt bên trong các câu lệnh có điều kiện như if() chỉ được kích hoạt khi điều kiện nhất định được đáp ứng. Vì vậy, ví dụ nếu bạn có mã như bên dưới, mã bên trong các câu lệnh 'if' sẽ không thực thi trừ khi nó thực sự được chạy trong biểu đồ (chỉ báo) hoặc trong thử nghiệm ngược danh mục đầu tư trong Phân tích mới window.

```
if( Status("action") == actionIndicator )
{
// breakpoint được đặt ở đây sẽ không kích hoạt trong trình gỡ lỗi vì điều kiện trên không được đáp ứng
}

if( Status("action") == actionPortfolio )
{
// breakpoint được đặt ở đây sẽ không kích hoạt trong trình gỡ lỗi vì điều kiện trên không được đáp ứng
}
```

AmiBroker thực thi mã trong nhiều ngữ cảnh khác nhau được chỉ định bằng trạng thái "action". Một số phần của mã có thể được thực thi có điều kiện trong một số ngữ cảnh nhất định bằng cách sử dụng các câu lệnh như được hiển thị ở trên. Ví dụ, điều này cho phép chỉ gọi Plot() trong biểu đồ và chỉ truy cập đối tượng kiểm tra ngược khi đối tượng đó khả dụng.

Kể từ phiên bản 6.10, trình gỡ lỗi chạy mã với ngữ cảnh actionBacktest. Để gỡ lỗi mã được chạy có điều kiện trong các ngữ cảnh khác, bạn có thể tạm thời tắt kiểm tra có điều kiện, nhưng xin lưu ý rằng bạn vẫn không thể truy cập một số đối tượng nhất định vì chúng đơn giản là không khả dụng. Cụ thể, GetBacktesterObject() khi được gọi ở bất kỳ đâu bên ngoài giai đoạn thứ hai của kiểm tra ngược danh mục đầu tư trong Phân tích mới sẽ trả về đối tượng rỗng (Null). Trong tương lai, hạn chế này có thể sẽ được gỡ bỏ.

Sử dụng các điều khiển GUI trên biểu đồ

Cơ bản

AmiBroker 6.30 cung cấp khả năng tạo và sử dụng các điều khiển GUI do người dùng xác định như nút, hộp kiểm, nút radio, trường chỉnh sửa, bộ chọn ngày giờ, thanh trượt, v.v. có thể được đặt trên biểu đồ và tương tác.

Tạo các điều khiển GUI

Có thể tạo các điều khiển GUI trên biểu đồ bằng cách gọi đến các hàm thích hợp như: GuiButton, GuiEdit, GuiToggle, GuiRadio, GuiDateTime, GuiSlider.

Tất cả các hàm tạo UI (GuiButton, GuiEdit, GuiToggle, GuiRadio, GuiDateTime, GuiSlider) đều lấy ID điều khiển làm tham số. ID điều khiển là một mã định danh duy nhất được sử dụng để phân biệt giữa các điều khiển và để thao tác điều khiển đã cho. ID điều khiển phải là số nguyên lớn hơn không. Như mã bên dưới cho thấy, bạn nên chỉ định một số tên có ý nghĩa cho ID điều khiển và sử dụng chúng thay vì các số thông thường.

Ví dụ sau đây cho thấy cách tạo nút:

```
// ID điều khiển idMyButton = 2;

GuiButton( "MyButton", idMyButton, 10, 20, 100, 20, notifyClicked);
```

Mô hình hoạt động của các điều khiển GUI trên biểu đồ là các điều khiển có thể bị hủy bất kỳ lúc nào khi người dùng chuyển sang trang biểu đồ khác. Vì lý do này, công thức cần có khả năng tạo lại các điều khiển UI nếu cần. Điều này được thực hiện chỉ bằng cách gọi các hàm tạo mỗi khi công thức được chạy. Các hàm tạo đủ thông minh để không tạo ra các bản sao.

Nếu điều khiển có ID đã cho đã thoát, nó sẽ được kiểm tra xem có giống nhau không. Nếu tìm thấy điều khiển đã tồn tại có cùng ID và cùng loại, không có gì được thực hiện và hàm tạo GUI trả về giá trị guiExisting. Nếu điều khiển tồn tại nhưng có loại khác, nó sẽ bị hủy và điều khiển có loại mới sẽ được tạo. Nếu điều khiển có ID đã cho không tồn tại, nó sẽ được tạo. Trong những trường hợp như vậy, hàm tạo sẽ trả về guiNew. Mã bên dưới cho biết cách kiểm tra giá trị trả về:

```
// ID điều khiển idMyButton = 2;

rc = GuiButton( "MyButton", idMyButton, 10, 20, 100, 20, notifyClicked );

if( rc == guiNew ) _TRACE("Control vừa tạo");
if( rc == guiExisting ) _TRACE("Control đã tồn tại");
```

Thực ra không cần thiết cho các nút không yêu cầu khởi tạo, nhưng hữu ích khi bạn muốn thiết lập một số giá trị ban đầu cho các điều khiển chỉ khi chúng được tạo. Ví dụ sau đây cho thấy cách đặt văn bản ban đầu cho trường chỉnh sửa:

```
// ID điều khiển idEdit = 1;

function CreateGUI()
{
rc = GuiEdit( idEdit, 10, 20, 100, 20, notifyEditChange );

if( rc == guiNew ) GuiSetText("Văn bản ban đầu", idEdit );

}

CreateGUI();
```

Xử lý sự kiện/thông báo UI

Các điều khiển GUI được tạo ra để cho phép người dùng tương tác với chúng. Khi điều này xảy ra (ví dụ khi người dùng nhấp vào nút), một sự kiện được tạo ra và công thức được thực thi lại. Một số điều khiển có thể kích hoạt sự kiện trong nhiều trường hợp, ví dụ khi nhấp vào nút, chỉnh sửa văn bản, tăng hoặc giảm tiêu điểm, v.v. Để quyết định khi nào điều khiển được cung cấp kích hoạt sự kiện hay không, chúng ta sử dụng tham số thông báo trong các hàm tạo Gui. Ví dụ:

```
GuiButton( "MyButton", idMyButton, 10, 20, 100, 20, notifyClicked );
```

sẽ tạo một nút sẽ kích hoạt sự kiện khi được nhấp vào. Chúng ta có thể kết hợp nhiều cờ khác nhau bằng cách sử dụng | (toán tử OR nhị phân) nếu chúng ta muốn được thông báo về nhiều sự kiện khác nhau bằng điều khiển được cung cấp, ví dụ, mã bên dưới sẽ tạo một điều khiển chỉnh sửa sẽ thông báo cho chúng ta khi nội dung của nó thay đổi và khi trường chỉnh sửa mất tiêu điểm:

```
GuiEdit( idEdit, 10, 20, 100, 20, notifyEditChange | notifyKillFocus );
```

Có nhiều thông báo có thể có bao gồm:

• notifyClicked - khi nút/chuyển đổi/hộp kiểm/radio được nhấp
• notifySetFocus - khi điều khiển được lấy tiêu điểm đầu vào (thông qua chuột hoặc bàn phím)
• notifyKillFocus - khi điều khiển mất tiêu điểm đầu vào (thông qua chuột hoặc bàn phím)
• notifyHitReturn - khi phím ENTER/RETURN được nhấn trong trường chỉnh sửa
• notifyEditChange - khi giá trị điều khiển thay đổi (văn bản bên trong trường chỉnh sửa, ngày bên trong ngày/giờ, thanh trượt được di chuyển đến vị trí mới)
• notifySelChange - khi lựa chọn thay đổi trong điều khiển (ví dụ: phạm vi văn bản đã chọn trong trường chỉnh sửa)

• notifyMouseEnter - khi chuột di chuột qua điều khiển
• notifyMouseLeave - khi chuột rời khỏi vùng điều khiển

Khi một sự kiện kích hoạt, công thức của chúng tôi sẽ được thực thi lại. Nếu nhiều sự kiện xảy ra, chúng sẽ được đưa vào hàng đợi. Để xử lý các sự kiện UI như vậy, công thức sẽ gọi hàm GuiGetEvent() để tìm xem có bất kỳ sự kiện nào trong hàng đợi hay không và xử lý chúng một cách thích hợp.

Hàm GuiGetEvent(num, what = 0) có hai tham số:

Tham số num định nghĩa chỉ số thông báo trong hàng đợi. 0 là lần đầu tiên được nhận (hoặc "cũ nhất") kể từ lần thực thi cuối cùng. Tham số thứ hai, what, định nghĩa giá trị nào được trả về bởi hàm GuiGetEvent:

• what = 0 - ID của điều khiển đã nhận sự kiện (số)
• what = 1 - mã thông báo (số)
• what = 2 - ID, mã và mô tả thông báo dưới dạng văn bản (chuỗi)

Thông thường sẽ không có (khi thực thi công thức không được kích hoạt bởi sự kiện UI) hoặc một sự kiện trong hàng đợi nhưng lưu ý rằng có thể có nhiều hơn một thông báo sự kiện trong hàng đợi nếu công thức của bạn xử lý chậm. Để truy xuất tất cả, hãy sử dụng tham số "num" tăng dần miễn là GuiGetEvent không trả về số không (ở chế độ what = 0, = 1) hoặc chuỗi rỗng (what = 2).

```
// đọc tất cả các sự kiện đang chờ
for( i = 0; id = GuiGetEvent( i ); i++ )
{
code = GuiGetEvent( i, 1 ); text = GuiGetEvent( i, 2 );
}
```

Thực hành tốt là có một hàm riêng xử lý các sự kiện như được hiển thị bên dưới:

```
idMyFirstButton = 1;

idMySecondButton = 2;

function CreateGUI()
{
GuiButton( "enable", idMyFirstButton, 10, 60, 100, 30, notifyClicked );

GuiButton( "disable", idMySecondButton, 110, 60, 100, 30, notifyClicked);
}

function HandleEvents()
{
for ( n = 0; id = GuiGetEvent( n, 0); n++) // lấy id của sự kiện
```

```
{
code = GuiGetEvent( n, 1);


switch ( id )
{
case idMyFirstButton:
// làm gì đó
break;


case idMySecondButton:
// làm gì đó khác
break;


default:
break;
}
}
}
```

```
CreateGUI();


HandleEvents();
```

Thao tác các điều khiển UI

Có thể thao tác các điều khiển theo một số cách:

Tất cả các điều khiển có thể được ẩn bằng cách sử dụng và hiển thị lại bằng cách sử dụng hàm GuiSetVisible():

```
GuiSetVisible( id, False); // ẩn điều khiển GuiSetVisible( id, True); // hiển thị điều khiển
```

Tất cả các điều khiển có thể bị vô hiệu hóa (để chúng không phản ứng với đầu vào của người dùng và trở thành 'màu xám') và được bật lại bằng hàm GuiEnable():

```
GuiEnable( id, False ); // vô hiệu hóa điều khiển GuiEnable( id, True ); // bật điều khiển
```

Bạn có thể đặt phông chữ cho tất cả các điều khiển GUI bằng lệnh gọi GuiSetFont:

```
GuiSetFont("Tahoma", 13 );
```

Theo mặc định, các nút (giống như tất cả các điều khiển khác) sử dụng màu hệ thống. Nhưng các nút có thể có màu tùy chỉnh riêng được xác định bằng GuiSetColors (các điều khiển khác chỉ sử dụng màu hệ thống).

```
GuiButton( "First", idFirst, 0, 50, 200, 30, 7 );
GuiButton( "Second", idSecond, 200, 50, 200, 30, 7 );


GuiButton( "Third", idThird, 400, 50, 200, 30, 7);
// hai nút đầu tiên sẽ có văn bản màu đỏ, đường viền và nền đen GuiSetColors( idFirst,
idSecond, 2, colorRed, colorBlack, colorRed);
// và văn bản màu xanh lá cây thứ ba, đường viền và nền xanh lam
GuiSetColors( idThird, idThird, 2, colorYellow, colorBlue, colorYellow);
```

Trong ví dụ đơn giản ở trên, chúng ta chỉ thiết lập màu bình thường của các nút, nhưng chúng cũng có màu riêng cho các trạng thái khác nhau (đã chọn, di chuột qua) và chúng cũng có thể được thiết lập.

Tất cả các điều khiển có thể cập nhật văn bản của chúng thông qua GuiSetText và được truy xuất bằng GuiGetText.

```
// ID điều khiển idEdit = 1;


function CreateGUI()
{
rc = GuiEdit( idEdit, 10, 20, 100, 20, notifyEditChange);


if( rc == guiNew ) GuiSetText("Initial text", idEdit );


}
```

Nếu bạn thích giá trị số của văn bản trong control hơn, bạn có thể sử dụng cặp GuiSetValue()/GuiGetValue() (đặc biệt hữu ích cho thanh trượt để lấy vị trí của chúng). Thanh trượt cũng có thể có phạm vi min-max được xác định bằng GuiSetRange() như trong ví dụ bên dưới.

```
idSlider = 1;


if ( GuiSlider( idSlider, 10, 30, 200, 30, notifyEditChange ) == guiNew )
{
// giá trị khởi tạo khi tạo control GuiSetValue( idSlider, 5 );


GuiSetRange( idSlider, 1, 100, 0.1, 100 );
}
```

Duy trì trạng thái của các điều khiển

Các điều khiển cơ bản như nút nhấn không có trạng thái 'vĩnh viễn'. Chúng chỉ được sử dụng để kích hoạt một số hành động khi người dùng nhấn nút. Nhưng một số điều khiển có trạng thái của chúng (thông tin bổ sung được lưu trữ trong chúng). Ví dụ: văn bản được

nhập vào trường chỉnh sửa, vị trí của thanh trượt, ngày được chọn trong điều khiển ngày giờ, trạng thái hộp kiểm, v.v.

Mỗi điều khiển (bao gồm cả điều khiển trên biểu đồ) trong HĐH Windows là đối tượng "cửa sổ" thực tế. Các điều khiển (đối tượng cửa sổ nói chung) tự duy trì trạng thái của chúng miễn là chúng "sống", tức là miễn là chúng tồn tại dưới dạng "đối tượng cửa sổ", nghĩa là miễn là khung biểu đồ đã cho được hiển thị.

Vì vậy, miễn là bạn không quan tâm đến việc duy trì trạng thái khi biểu đồ không được hiển thị, bạn không cần phải làm gì cả. Nếu không, hãy làm theo các bước dưới đây.

Vấn đề là các điều khiển GUI trên biểu đồ được tạo động bởi mã của bạn khi công thức được thực thi. Chúng có thể tự động bị hủy nếu bạn đóng cửa sổ biểu đồ hoặc nếu bạn chuyển sang trang tính khác. Sau đó, chúng có thể được tạo lại vào lần thực thi công thức tiếp theo.

Thực tế này có một hậu quả - nếu bạn muốn giữ trạng thái điều khiển của mình giữa "destory" và "recreation", bạn phải tự lưu trữ trạng thái. Điều này có thể được thực hiện bằng cách sử dụng ví dụ như các biến tĩnh (hoặc các biến tĩnh vĩnh viễn nếu bạn muốn giữ trạng thái giữa các lần chạy AmiBroker).

Như đã thảo luận trước đó, các hàm tạo điều khiển như GuiButton, GuiEdit, GuiDateTime, GuiCheckbox, GuiRadio, GuiSlider, v.v., tất cả đều trả về giá trị guiNew nếu điều khiển mới được tạo hoặc guiExisting nếu điều khiển đã cho (đối tượng cửa sổ) đã tồn tại. Điều này cho phép chúng ta biết khi nào cần khôi phục trạng thái đã lưu trước đó. Ví dụ, giả sử rằng chúng ta đã lưu văn bản điều khiển chỉnh sửa trong biến tĩnh "mySavedText", chúng ta có thể khôi phục trạng thái bằng cách sử dụng mã như bên dưới:

```
// ID điều khiển idEdit = 1;

function CreateGUI()
{
rc = GuiEdit( idEdit, 10, 20, 100, 20, notifyEditChange );

if( rc == guiNew ) GuiSetText( StaticVarGetText( "mySavedText" ) , idEdit );

}
```

Bây giờ chúng ta cũng có thể muốn lưu trữ văn bản vào biến tĩnh bất cứ khi nào nó thay đổi:

```
function HandleEvents()
{
for( n = 0; id = GuiGetEvent( n, 0 ); n++ )
{
switch( id )
{
case idEdit:
```

```
StaticVarSetText( "mySavedText", GuiGetText( idEdit ), True /*

làm cho nó tồn tại */ );
break;
/// phần còn lại của mã xử lý sự kiện ....
}
}
}
```

Khi chúng ta lưu trữ giá trị của văn bản đã chỉnh sửa trên mỗi thay đổi vào biến tĩnh tồn tại, nó sẽ được bảo toàn giữa các lần chạy AmiBroker và được khôi phục bất cứ khi nào điều khiển chỉnh sửa được (tái) tạo lại.

Kỹ thuật tương tự cũng có thể được sử dụng cho các điều khiển khác.

Ví dụ

Mã ví dụ bên dưới cho thấy nhiều kỹ thuật được mô tả ở trên, bao gồm thiết lập giá trị ban đầu khi tạo, sửa đổi thuộc tính điều khiển, bật/tắt điều khiển, hiển thị/ẩn điều khiển, xử lý sự kiện GUI.

```
// định danh điều khiển idSlider = 1;
idEnable = 2;
idDisable = 3;
idShow = 4;
idHide = 5;

function CreateGUI()
{
if ( GuiSlider( idSlider, 10, 30, 200, 30, notifyEditChange ) == guiNew )
{
// giá trị khởi tạo khi tạo điều khiển GuiSetValue( idSlider, 5 );

GuiSetRange( idSlider, 1, 100, 0.1, 100 );
}

GuiButton( "enable", idEnable, 10, 60, 100, 30, notifyClicked );

GuiButton( "vô hiệu hóa", idDisable, 110, 60, 100, 30, notifyClicked);
GuiButton( "hiển thị", idShow, 10, 90, 100, 30, notifyClicked);
GuiButton( "ẩn", idHide, 110, 90, 100, 30, notifyClicked);
}

hàm HandleEvents()
{
for( n = 0; id = GuiGetEvent( n, 0 ); n++ )
{
```

```
switch ( id )
{
trường hợp idEnable:
GuiEnable( idSlider, True );
break;


trường hợp idDisable:
GuiEnable( idSlider, False );


break;


trường hợp idShow:
GuiSetVisible( idSlider, True );
break;


case idHide:
GuiSetVisible( idSlider, False);
break;
}
}
}


CreateGUI();
HandleEvents();


Title = "Value = " + GuiGetValue( idSlider);
```

Lưu ý

Vui lòng cân nhắc với các hàm Gui* và lưu ý đến giới hạn của Windows. Vì tất cả các điều khiển trong Windows (nút, hộp chỉnh sửa, v.v.) đều là các đối tượng Window thực tế nên chúng phải tuân theo giới hạn của Windows. Và có giới hạn là 10000 cửa sổ MỖI QUY TRÌNH. Vì vậy, đừng cố tạo hàng nghìn nút (như một nút cho mỗi thanh dữ liệu) vì trước tiên bạn sẽ thấy hiệu suất giảm đáng kể và sau đó bạn sẽ đạt đến giới hạn và gặp phải sự cố (sập)
https://blogs.msdn.microsoft.com/oldnewthing/20070718-00/?p=25963

Thực hành tốt nhất là giữ số lượng dưới 100-200. Nếu bạn cần nhiều hơn, hãy cân nhắc sử dụng đồ họa cấp thấp thay thế.

Video hướng dẫn (trực tuyến)

Sau đây là các video hướng dẫn trình bày nhiều lĩnh vực khác nhau của AmiBroker.

Biểu đồ

• Cách tùy chỉnh biểu đồ giá CustomizePrice.mp4

• Cách sử dụng biểu đồ kéo thả dragdrop1.mp4

• Cách hiển thị biểu đồ của hai ký hiệu cùng lúc TwoSymbols.mp4

• So sánh hai ký hiệu trên một biểu đồ: TwoSymbolsOneChart.mp4

• Cách hiển thị Khối lượng theo Giá: VolumeAtPrice.mp4

• Cách tạo chỉ báo tùy chỉnh: CustomIndicator.mp4

Dữ liệu

• Cách tải xuống dữ liệu miễn phí từ thị trường Hoa Kỳ bằng AmiQuote amiquote3.mp4

• Cách thiết lập AmiBroker với dữ liệu thời gian thực miễn phí từ Interactive Brokers ib.mp4

• Cách cấu hình với dữ liệu IQFeed IQFeed.mp4

• Cách cấu hình với dữ liệu eSignal esignal.mp4

• Cách tải xuống dữ liệu forex miễn phí: forex.mp4

• Cách cấu hình với Quotes Plus: qp3.mp4

• Cách cấu hình với TC2000/TCNet tc2k.mp4

Giao diện người dùng

• Cách sử dụng cửa sổ docking DockingWindows.mp4

• Cách thêm định nghĩa khoảng thời gian trong ngày tùy chỉnh CustomTimeInterval.mp4

• Cách thêm Report Explorer làm công cụ/nút tùy chỉnh: RepExInToolbar.mp4

• Cách tùy chỉnh giao diện người dùng uicustomize.mp4

• Cách thay đổi kích thước hộp nhập ký hiệu uiresizeticker.mp4

• Cách sử dụng lớp layers.mp4

• Cách sử dụng layouts layouts.mp4

Phân tích

• Cách chạy backtest hệ thống (ĐÃ CẬP NHẬT) BackTest.mp4

• Cách tạo khám phá của riêng bạn (4.80) exploration.mp4

• Cách tối ưu hóa các tham số của hệ thống giao dịch (4.80) optimize.mp4

Để biết thêm hướng dẫn bằng video, vui lòng kiểm tra:
https://www.amibroker.com/support.html

Hướng dẫn tham khảo Amibroker

- Tham khảo giao diện người dùng AmiBroker
- Tham khảo công cụ nhập ASCII
- Mô hình đối tượng tự động hoá OLE của AmiBroker
- Mô hình đối tượng tự động hoá OLE của AmiQuote

Windows trong AmiBroker

Phần này mô tả chức năng của các cửa sổ trong AmiBroker.

Tất cả các cửa sổ này đều hoạt động không đồng bộ, tức là bạn có thể mở bao nhiêu cửa sổ tùy thích và làm việc với tất cả chúng cùng một lúc.

Chức năng vẽ biểu đồ

- Cửa sổ pane biểu đồ (Chart window pane)
- Cửa sổ dữ liệu (Data window)
- Cửa sổ thông số (Parameters window)
- Công cụ vẽ nghiên cứu (Study drawing tools)
- Cửa sổ thuộc tính dòng nghiên cứu (Line study properties window)
- Cửa sổ thuộc tính hộp văn bản (Text box properties window)
- Trình soạn thảo công thức (Formula editor)
- Cửa sổ bản đồ rủi ro-lợi nhuận (Risk-yield map window)
- Cửa sổ đặt lệnh (Place order window)

Cài đặt

- Cài đặt cơ sở dữ liệu / Cài đặt trong ngày (Database settings / Intraday settings)
- Sở thích (Preferences)
- Cửa sổ tuỳ chỉnh công cụ (Customize tools window)

Ký hiệu/Dữ liệu

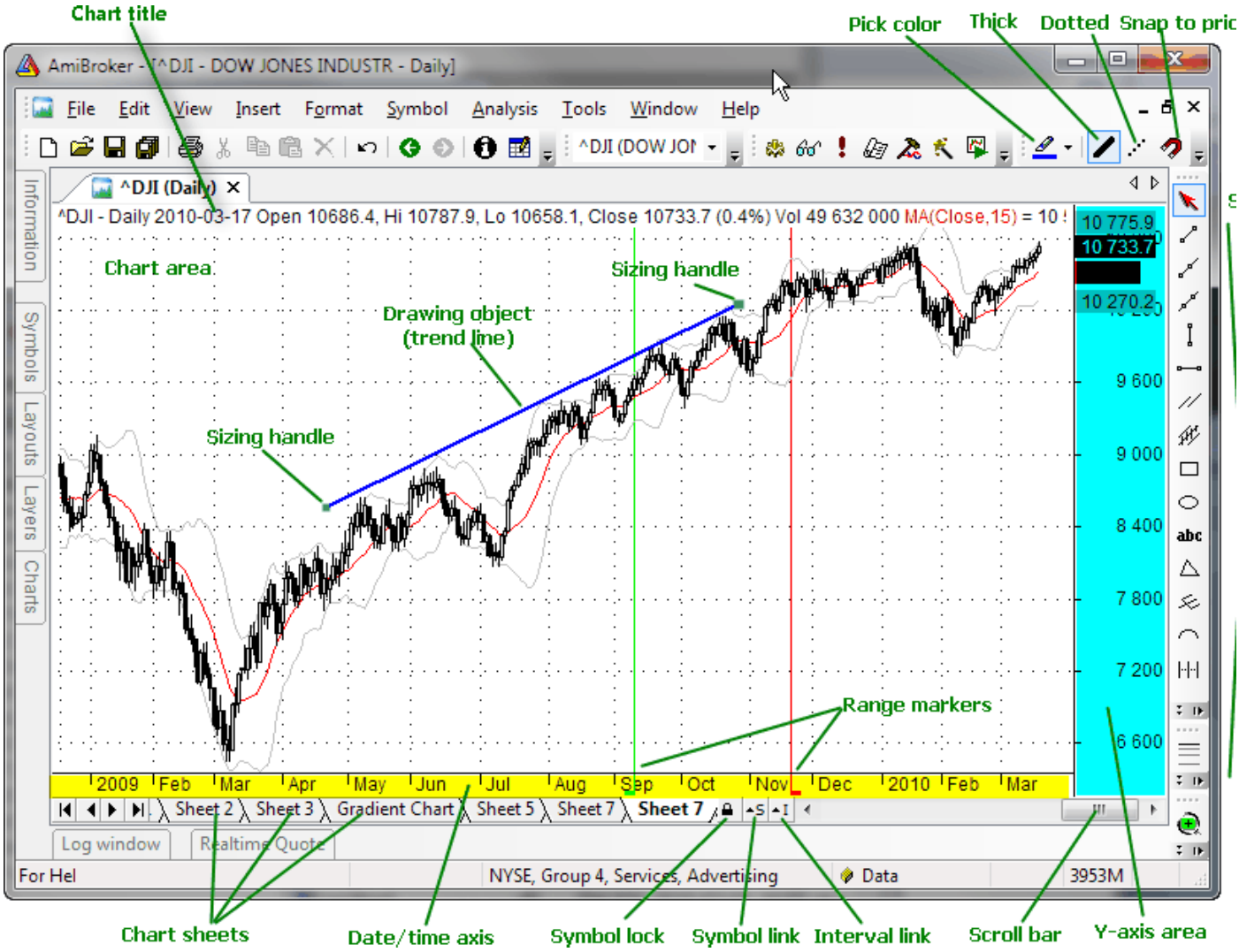
- Cây ký hiệu (Symbol tree)
- Cửa sổ thông tin (Information window)
- Cửa sổ ghi chú (Notepad window)
- Cửa sổ biên tập Trích dẫn (Quote Editor window)
- Cửa sổ tìm kiếm ký hiệu (Symbol finder window)
- Cửa sổ tài chính (Finance window)
- Chế độ xem hồ sơ (Profile view)
- Cửa sổ tổ chức công việc (Assignment organizer window)
- Cửa sổ máy tính tổng hợp (Composite calculator window)
- Cửa sổ danh mục (Categories window)
- Cửa sổ thuật sĩ nhập (Import Wizard window)
- Cửa sổ nhập khẩu Metastock (Metastock importer window)
- Cửa sổ báo giá thời gian thực (Real-time Quote window)
- Cửa sổ cảnh báo dễ dàng (Easy Alerts window)
- Cửa sổ thời gian/giao dịch (Time/Sales window)
- Cửa sổ phát lại thanh (Bar Replay window)

Phân tích/Công cụ

- Trình soạn thảo công thức (Formula editor)
- Cửa sổ đoạn mã (Code Snippets window)
- Cửa sổ xem nhanh (Quick review window)
- Cửa sổ phân tích (Analysis window)

- Cửa sổ cài đặt bộ lọc (Filter settings window)
- Cửa sổ cài đặt kiểm thử hệ thống (System test settings window)
- Cửa sổ biểu đồ hoa hồng (Commission schedule window)
- Cửa sổ báo cáo kiểm thử hệ thống (System test report window)
- Cửa sổ bình luận (Commentary window)
- Cửa sổ plugins (Plugins window)
- Trình hướng dẫn bảo dưỡng chỉ số (Indicator Maintenance wizard)
- Cửa sổ nhật ký (Log window)
- Trình giám sát hiệu suất (Performance Monitor)

Cửa sổ biểu đồ



Cửa sổ này hiển thị biểu đồ của các chỉ báo kỹ thuật khác nhau.

Ở dưới cùng của biểu đồ, bạn có thể thấy trục X; tùy thuộc vào cài đặt trong cửa sổ tham số, nó có thể hiển thị ngày hoặc không. Dưới đó, bạn có thể thấy thanh cuộn và điều khiển tab của các trang biểu đồ. Thanh cuộn có thể được sử dụng để hiển thị các báo giá trong quá khứ, trong khi tab trang cho phép bạn xem các trang biểu đồ khác nhau.

Bên phải, bạn sẽ thấy khu vực trục Y (được đánh dấu bằng màu xanh) hiển thị thang giá trị và nhãn giá trị. Nhãn giá trị là các ô màu hiển thị chính xác "giá trị cuối cùng" của các đồ thị. "Giá trị cuối cùng" là giá trị của chỉ báo (hoặc giá) cho thanh hiện tại được hiển thị bên phải cùng. Khu vực trục Y cũng được sử dụng để di chuyển/thay đổi kích thước biểu đồ theo chiều dọc.

Các tham số và cài đặt của biểu đồ có thể được điều chỉnh bằng cách nhấp chuột phải vào biểu đồ và chọn tùy chọn **Tham số** từ **menu ngữ cảnh của biểu đồ.**

Biểu đồ cũng có thể được cuộn, thay đổi kích thước, di chuyển, thu nhỏ, hoặc điều chỉnh kích thước - để tìm hiểu thêm, hãy đọc **Hướng dẫn: Hướng dẫn biểu đồ cơ bản.**

Cửa sổ tham số

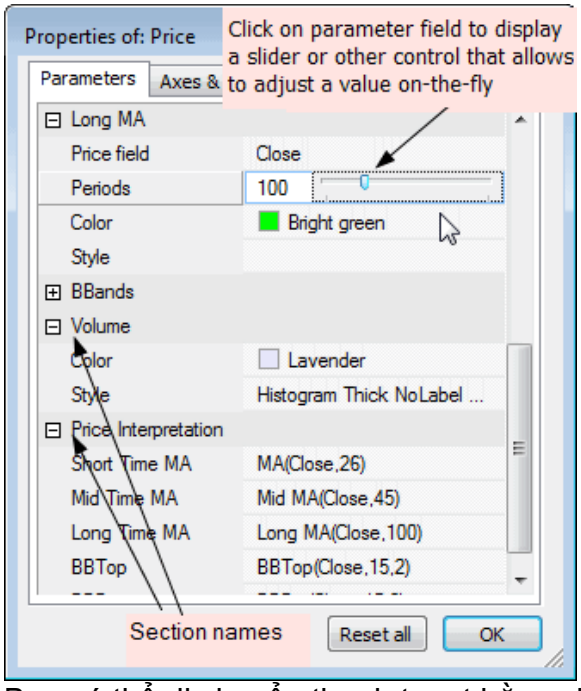
Cửa sổ này cho phép người dùng sửa đổi các tham số được chỉ định trong công thức AFL thông qua các hàm Param, ParamStr, ParamColor, ParamStyle, ParamField, ParamToggle, ParamDate, ParamTime, ParamList và cũng để điều chỉnh cài đặt trục và lưới.

Nó có thể truy cập thông qua menu ngữ cảnh biểu đồ (nhấp chuột phải lên khung biểu đồ để xem menu ngữ cảnh): chọn Tham số và một cửa sổ nhỏ với danh sách tham số sẽ xuất hiện. Để chỉnh sửa giá trị tham số, đơn giản hãy nhấp vào trường giá trị mục như được hiển thị trong hình. Sau đó tuỳ thuộc vào loại tham số, các điều khiển phù hợp sẽ xuất hiện.

Ví dụ, nếu tham số đã cho là một chuỗi thì trường văn bản sẽ xuất hiện, và nếu tham số đã cho là màu thì điều khiển chọn màu sẽ cho phép bạn thay đổi màu sắc.

Khi chỉnh sửa các tham số số, bạn có thể điều chỉnh giá trị bằng cách nhập giá trị vào trường chỉnh sửa hoặc di chuyển điều khiển trượt. Để hiển thị trường chỉnh sửa - nhấp vào chính số đó (được đánh dấu bằng màu xanh trong hình dưới đây). Để hiển thị điều khiển trượt, nhấp vào bên cạnh số (bên phải).

Nếu tham số đã cho là một số thì điều khiển trượt hoặc trường chỉnh sửa sẽ được hiển thị như trong hình dưới đây:



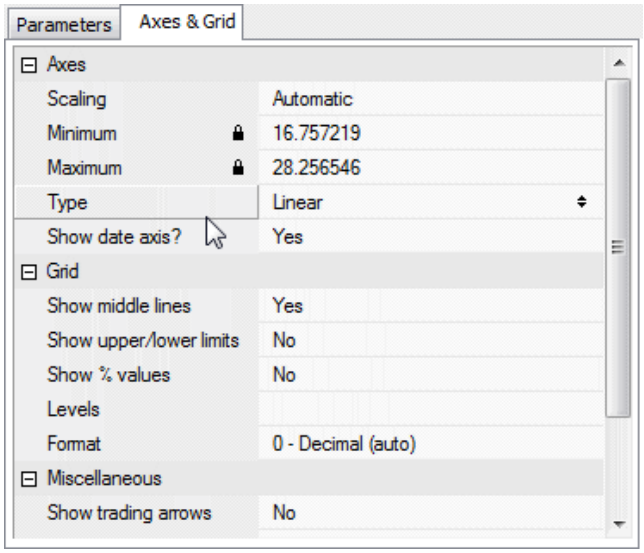
Bạn có thể di chuyển thanh trượt bằng chuột, phím mũi tên <- -> và bánh xe chuột. Khi thay đổi được thực hiện, biểu đồ dưới nền sẽ được làm mới ngay lập tức, tạo ra phản hồi tốt cho người dùng.

Tham số được nhóm thành "phần". Phần biểu thị phần của các mã được bao quanh bởi các dấu đánh dấu _SECTION_BEGIN/_SECTION_END. Để tìm hiểu thêm về điều này hãy xem Hướng dẫn: Sử dụng giao diện kéo và thả.

Bất cứ lúc nào bạn có thể nhấn nút **Đặt lại tất cả** sẽ đặt lại tất cả các tham số về giá trị mặc định.

Để biết thêm thông tin về việc sử dụng các tham số, vui lòng đọc **Hướng dẫn: Sử dụng màu, kiểu, tiêu đề và tham số trong các chỉ số và Hướng dẫn: Sử dụng giao diện kéo và thả.**

Cửa sổ Tham số cũng cho phép điều chỉnh hình dạng **trục và lưới** cũng như một số cài đặt khác cho từng biểu đồ. Các điều khiển này có sẵn ở tab thứ hai Axes & Grid như được hiển thị dưới đây:

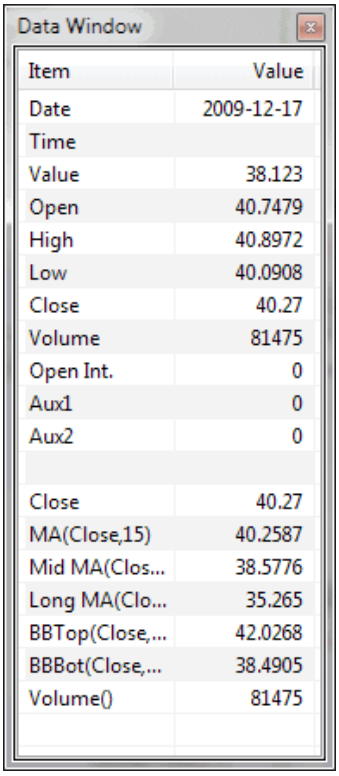


Các tùy chọn sau đây có sẵn:

- **Axes**
  - **Tự động điều chỉnh trục:**
    - **Tự động** - giá trị tối thiểu và tối đa của trục Y được xác định tự động bởi AmiBroker
    - **Tùy chỉnh** - giá trị tối thiểu và tối đa của trục Y do người dùng định rõ
  - **Tối thiểu** - giá trị tối thiểu của trục Y (thuộc tính này bị khóa nếu tự động điều chỉnh được chọn, để mở khóa chọn Tự động điều chỉnh)
  - **Tối đa** - giá trị tối đa của trục Y (thuộc tính này bị khóa nếu tự động điều chỉnh được chọn, để mở khóa chọn Tự động điều chỉnh)
  - **Loại**
    - **Tuyến tính** - sử dụng tỷ lệ trục Y tuyến tính ◊ Logarithmic - sử dụng tỷ lệ trục Y logarithmic
    - **Hiển thị trục ngày** - bật/tắt hiển thị ngày trên trục X
- **Grid**
  - **Hiển thị các đường kẻ giữa** - hiển thị các đường lưới trục Y tự động cách đều giữa giá trị tối thiểu và tối đa
  - **Hiển thị giới hạn trên/dưới** - hiển thị nhãn giá trị tối thiểu và tối đa của trục Y
  - **Hiển thị giá trị %** - hiển thị giá trị theo tỷ lệ phần trăm

    - **Các mức giới hạn** - cho phép bật các đường lưới ở một số mức phổ biến cố định như 30/70, 20/80, 10/90, -100/+100, 0
- **Miscellaneous**
    - **Hiển thị mũi tên giao dịch** - khi bật lựa chọn này, pane này sẽ hiển thị mũi tên mua/bán/bán ngắn/chốt lời được tạo ra từ các tùy chọn tương ứng sẵn sàng từ menu Phân tích tự động.
    - **Đường dẫn File** (bị khóa) - hiển thị đường dẫn đến file công thức mà biểu đồ đã cho sử dụng
    - **ID biểu đồ** (bị khóa) - hiển thị giá trị số của ID biểu đồ mà pane đã cho sử dụng. ID biểu đồ không quan trọng trừ khi bạn sử dụng hàm Study() trong công thức của bạn.

Cửa sổ Dữ liệu (Data Window)



Cửa sổ Dữ liệu có thể được hiển thị bằng cách sử dụng menu **Window->Data Window.**

Cửa sổ Dữ liệu hiển thị ngày/giờ và các giá trị mở, cao, thấp, đóng, khối lượng, mở hấp dẫn, aux1 và aux2 của bar dưới con trỏ chuột.Nó cũng hiển thị tọa độ Y của con trỏ chuột ("Value") được biểu thị dưới dạng giá tương ứng với vị trí con trỏ chuột hiện tại.

Cửa sổ Dữ liệu cũng hiển thị các giá trị của tất cả các chỉ số được xác định trong công thức. Các giá trị này tự động cập nhật khi con trỏ dừng lại trong một khoảng thời gian ngắn.

Công cụ vẽ nghiên cứu của AmiBroker

Công cụ vẽ học thuật của AmiBroker có thể truy cập từ thanh công cụ **Draw / Fibonacci & Gann:**



Các công cụ sau đây có sẵn:

• Đường xu hướng

• Tia (mới trong 4.20)

• Đường mở rộng (mới trong 4.20)

• Đường thẳng đứng

• Đường ngang

• Đường song song (mới trong 4.20)

• Kênh hồi quy: Raff, độ lệch chuẩn, lỗi chuẩn (tất cả đều mới trong 4.20)

• Nghiên cứu Fibonacci Retracement (nâng cao trong 4.20)

• Nghiên cứu Fibonacci Time Zones

• Fibonacci Extensions (mới trong 4.60)

• Fibonacci Time Extensions (mới trong 4.60)

• Fibonacci Fan

• Fibonacci arc

• Gann Square (mới trong 4.20)

• Gann Fan (mới trong 4.20)

• Công cụ hình elip

• Công cụ hình tam giác (mới trong 4.30)

• Andrews' Pitchfork (mới trong 4.30)

• Công cụ Cycles (mới trong 4.60)

• Công cụ Arrow (mới trong 4.70)

• Công cụ Zig-zag (mới trong 4.70)

• Công cụ Arc

• Rectangle

• Công cụ text box

Công cụ mặc định Chọn (mũi tên đỏ) được sử dụng để chọn đối tượng vẽ và các báo giá trên biểu đồ. Nếu bạn muốn vẽ nghiên cứu đã cho, chỉ cần bật nút thích hợp và bắt đầu vẽ trên biểu đồ bằng cách chỉ chuột vào nơi bạn muốn bắt đầu vẽ và nhấn giữ nút chuột trái. Sau đó di chuyển chuột. Dòng theo dõi nghiên cứu sẽ xuất hiện. Thả nút chuột trái khi bạn muốn kết thúc việc vẽ.Bạn cũng có thể hủy bỏ việc vẽ biểu đồ bằng cách nhấn phím ESC (Escape). Để xem hướng dẫn cho người mới bắt đầu về biểu đồ, hãy kiểm tra Hướng dẫn: Biểu đồ.

Đường xu hướng, Ray, Mở rộng, Đứng, Nằm ngang

Những công cụ này cung cấp các biến thể khác nhau của đường xu hướng cơ bản. Đường xu hướng tạo ra một đoạn thẳng, Ray tạo ra một đường xu hướng kéo dài sang bên phải, Mở rộng tạo ra đường xu hướng tự động kéo dài từ cả hai bên trái và phải. Đứng và Nằm ngang thì tự giải thích.

Mũi tên

Tương tự như Đường xu hướng nhưng kết thúc bằng một mũi tên.

Zig-zag

Vẽ một chuỗi các đường xu hướng liên kết. Để kết thúc việc vẽ, nhấn phím ESC.

Đồng song song

Công cụ này cho phép vẽ một chuỗi các đoạn đường xu hướng song song. Đầu tiên, bạn vẽ một đường xu hướng như bình thường, sau đó một đường thứ hai song song với đường đầu tiên sẽ được tạo tự động và bạn có thể di chuyển chúng bằng chuột. Khi bạn nhấp vào biểu đồ, nó sẽ được đặt ở vị trí đã cho. Sau đó, một đường song song khác xuất hiện có thể được đặt ở nơi khác. Và lặp lại. Để dừng lại, hãy nhấn phím ESC hoặc chọn công cụ "Select".

Kênh hồi quy

AmiBroker cho phép dễ dàng vẽ 3 loại kênh hồi quy:

- Kênh hồi quy Raff

- Kênh lỗi chuẩn

- Kênh độ lệch chuẩn

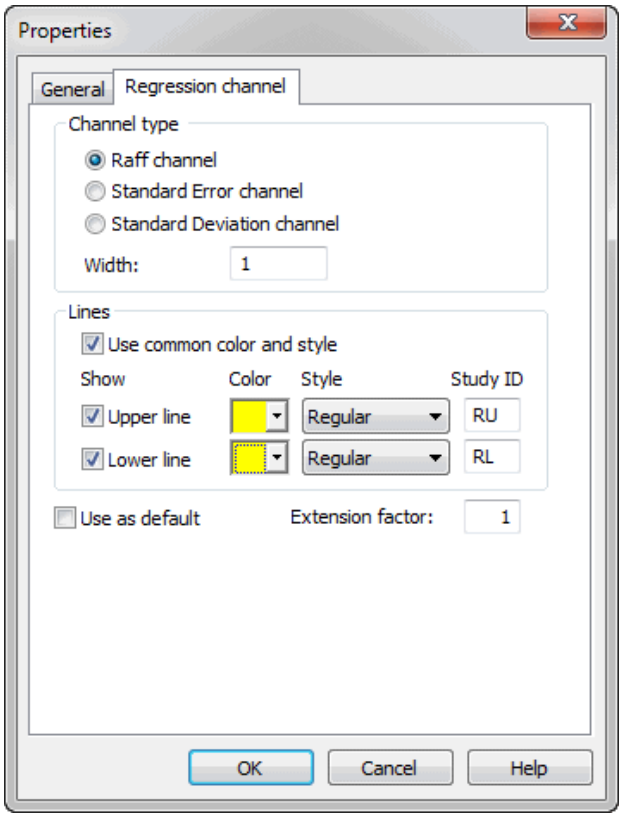
Tất cả những kênh này đều dựa trên đường xu hướng hồi quy tuyến tính.

Kênh Hồi quy được xây dựng bằng cách vẽ hai đường song song, cách đều nhau trên và dưới một đường hồi quy tuyến tính. Khoảng cách giữa các đường kênh đến đường hồi quy là khoảng cách lớn nhất mà bất kỳ một giá cao hoặc thấp nào từ đường hồi quy.

Kênh Lỗi Chuẩn được xây dựng bằng cách vẽ hai đường song song trên và dưới một đường hồi quy tuyến tính. Các đường được vẽ một số lượng lỗi chuẩn cụ thể xa đường hồi quy tuyến tính.

Kênh Độ Lệch Chuẩn được xây dựng bằng cách vẽ hai đường song song trên và dưới một đường hồi quy tuyến tính. Các đường được vẽ một số lượng lỗi chuẩn cụ thể xa đường hồi quy tuyến tính.

Bạn có thể chọn loại kênh bằng cách nhấp đúp vào nghiên cứu kênh (hoặc chọn Properties từ menu nút chuột phải)



Nếu **Use common color and style** được đánh dấu thì dòng kênh sẽ sử dụng cùng một kiểu và màu sắc với đường hồi quy (đường giữa). Nếu nó không được đánh dấu, bạn có thể thiết lập màu sắc và kiểu riêng biệt cho dòng kênh trên và dưới. Bạn cũng có thể tắt hoàn toàn dòng kênh trên và dưới bằng cách bỏ chọn **Show Upper line và Show Lower line.**

Cột "Study ID" xác định định danh nghiên cứu có thể được sử dụng trong công thức tùy chỉnh của bạn để phát hiện sự chéo lên. Bạn có thể thay đổi các ID này nếu cần thiết bằng cách chỉnh sửa đơn giản những trường này. Để biết thêm thông tin về Study ID, hãy kiểm tra Hướng dẫn sử dụng: Sử dụng nghiên cứu trong công thức AFL.

Thông tin thêm về các kênh hồi quy có sẵn từ Hướng dẫn phân tích kỹ thuật.

Công cụ vẽ Hình elip và Hình cung

Công cụ vẽ mới này được kết nối với tọa độ ngày/giá (như đường xu hướng) thay vì pixel màn hình nên chúng có thể thay đổi hình dạng hình ảnh khi hiển thị ở các yếu tố phóng to hoặc kích thước màn hình khác nhau.

Để xem các thuộc tính của những yếu tố này, bạn nên nhấp đúp vào vị trí giờ giống như đồng hồ 3, 6, 9 hoặc 12.

Hình cung Fibonacci

Công cụ vẽ mới này tạo ra các hình cung Fibonacci chuẩn được điều khiển bởi đường xu hướng được vẽ bằng kiểu chấm chấm. Để xem các thuộc tính của các hình cung, nhấp vào đường xu hướng kiểm soát.

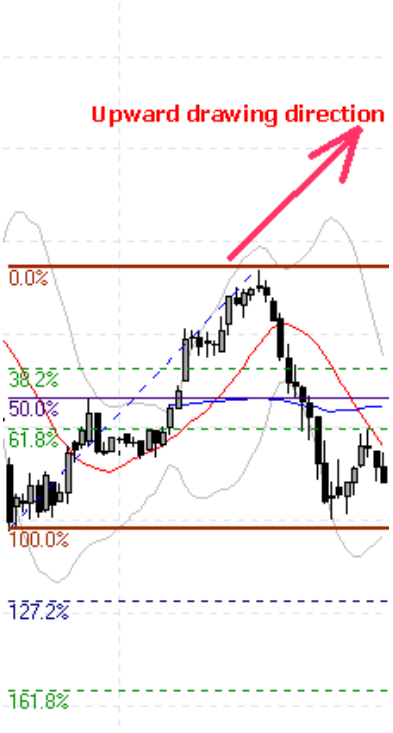
Lưu ý rằng bán kính hình cung và điểm trung tâm liên quan đến đường xu hướng kiểm soát và bởi vì hình cung Fibonacci phải hình tròn bất kể kích thước màn hình/độ phân giải và yếu tố phóng to, vị trí của các hình cung có thể di chuyển trong lĩnh vực ngày / giá.

Fibonacci Retracement

Trước tiên, hãy lưu ý rằng công cụ Fibonacci hoạt động khác nhau tùy thuộc vào hướng vẽ và cờ "hiện phần mở rộng". Xem các hình ảnh bên dưới.

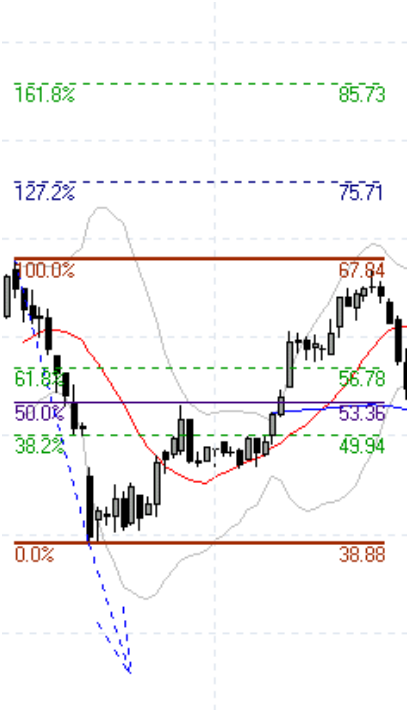


Hướng vẽ hướng lên Hiển thị phần mở rộng BẬT

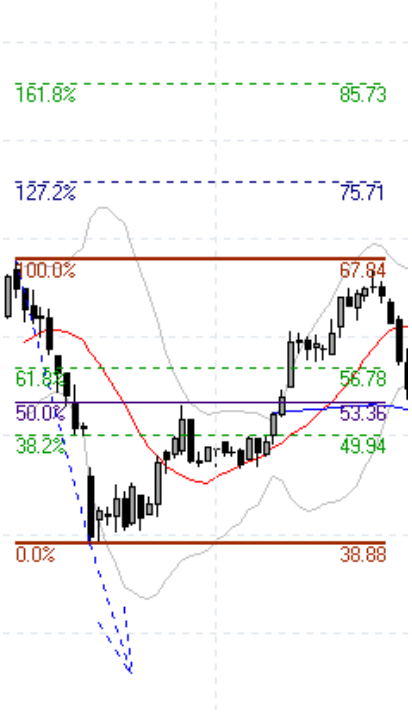


Hướng vẽ hướng lên Hiển thị phần mở rộng TẮT

Như bạn có thể thấy, công cụ này hiển thị cả các mức hồi quy (38.2, 50, 61.8) và các mức mở rộng (127.2, 161.8). Nếu hộp "hiện phần mở rộng" được tắt, công cụ chỉ hiển thị CÁC MỨC HỒI QUY. Nó hoạt động tương tự khi kiểm soát đường xu hướng được vẽ theo chiều hướng xuống.

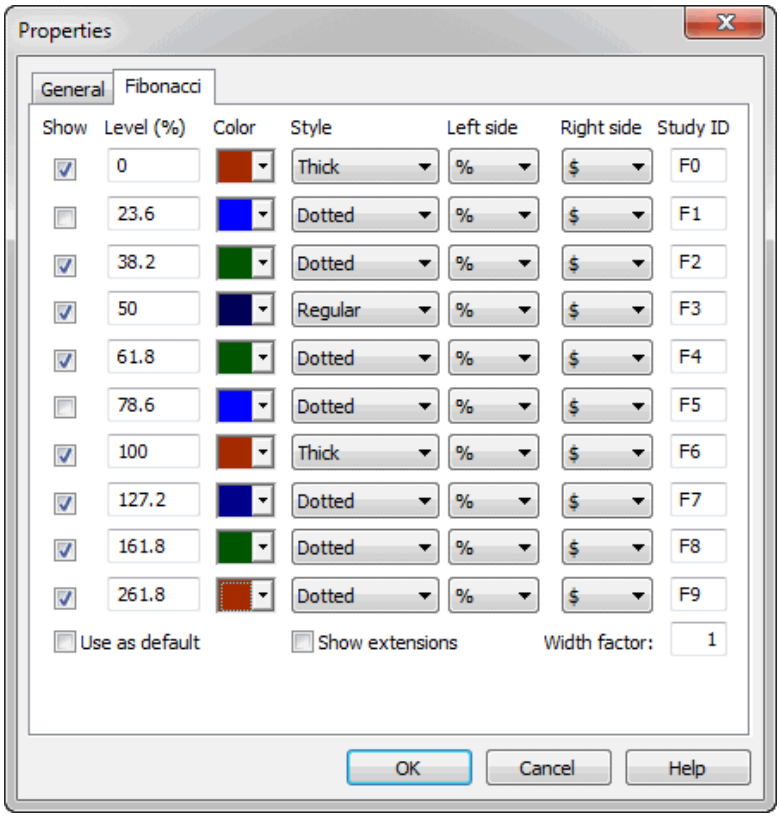


Hướng vẽ xuống Hiển thị phần mở rộng BẬT



Hướng vẽ xuống Hiển thị phần mở rộng TẮT

Bây giờ hãy tìm hiểu thêm về cửa sổ cài đặt Fibonacci:



Cột đầu tiên "Hiện" bật/tắt từng đường.

Cột thứ hai "Mức (%)" xác định mức phần trăm. 100 và 0 đại diện cho tọa độ Y thông qua các điểm bắt đầu và kết thúc của đường xu hướng điều khiển.

Cột thứ ba "Color" xác định màu của đường, cột thứ tư "Style" cho phép chọn giữa các kiểu thông thường, dày và chấm.

Cột thứ năm và thứ sáu "Left side" và "Right side" điều khiển việc hiển thị văn bản xuất hiện ở phía trái và phải của dòng mức Fibonacci. Trống - nghĩa là không có văn bản, % - nghĩa là mức phần trăm, $ - nghĩa là mức đô la (điểm).

Cột thứ bảy "Study ID" định nghĩa ID nghiên cứu có thể được sử dụng trong công thức tùy chỉnh của bạn để phát hiện crossover. Mỗi mức Fibonacci có một ID riêng mặc định là F0... F9. Bạn có thể thay đổi các ID này nếu cần bằng cách chỉnh sửa đơn giản các trường này.

Như được mô tả trong Hướng dẫn sử dụng: Hướng dẫn sử dụng: Using studies in AFL formulas

Bạn có thể dễ dàng viết công thức kiểm tra sự thâm nhập của mức Fibonacci cụ thể. Trong ví dụ này, chúng tôi sẽ phát hiện xem giá đóng cửa có giảm dưới mức F2 (mức retracement 38.2%) hay không. Công thức rất đơn giản:

```
sell = cross( study( "F2" ), close )
```

Lưu ý rằng hàm study() chấp nhận hai đối số: đối số đầu tiên là mã hai chữ cái StudyID tương ứng với một mã cho trong hộp thoại thuộc tính; đối số thứ hai là ID biểu đồ - mặc định là 1 (khi không được đưa ra hoàn toàn) và sau đó nó ánh xạ các nghiên cứu được vẽ trong khung giá chính. Đối với việc kiểm tra các nghiên cứu được vẽ ở các khung khác, bạn nên sử dụng các mã được đưa ra ở trên (trong bảng mô tả hàm study()).

Vui lòng lưu ý rằng công thức này là phổ biến - nó sẽ sử dụng mức phù hợp từ bất kỳ biểu tượng nào có đường Fibonacci.

Điều này xảy ra vì AmiBroker lưu trữ dữ liệu của tất cả các nghiên cứu đã vẽ trong cơ sở dữ liệu của nó. Khi bạn quét bằng mã ở trên, AmiBroker kiểm tra xem các mức Fibonacci có được vẽ cho ký hiệu hiện tại hay không. Nếu tìm thấy, nó xem nghiên cứu F2 là gì – ví dụ, nó nhận ra đây là một đường Fibonacci 38.2% nằm ở mức $29.06 cho một ký hiệu cụ thể. Do đó, AmiBroker thay thế nghiên cứu ("F2") bằng $29.06 (lưu ý: đây là sự đơn giản hóa – thực tế nó tạo ra một mảng đại diện cho đường xu hướng) và kiểm tra sự cắt ngang.

"**Yếu tố mở rộng**" quyết định độ dài mà các đường được mở rộng sang bên phải (theo hướng trục X). Nếu bạn nhập 2, bạn sẽ nhận được các đường mở rộng gấp đôi so với mặc định '1'. Nếu bạn nhập 0, các đường mức Fibonacci sẽ kết thúc ở vị trí mà đường xu hướng điều khiển kết thúc.

" **Sử dụng làm mặc định**" – nếu bạn đánh dấu vào hộp này và chấp nhận cài đặt bằng cách nhấn **OK**, tất cả các vẽ Fibonacci mà bạn thực hiện sau này sẽ sử dụng các cài đặt này.Khi sử dụng công cụ hộp văn bản, chỉ cần gõ văn bản vào hộp. Khi bạn muốn kết thúc, hãy nhấp ra ngoài hộp văn bản. Bạn cũng có thể hủy chỉnh sửa bằng cách nhấn phím ESC.

Mở rộng Fibonacci

Công cụ Mở rộng Fibonacci tương tự như công cụ Hồi quy Fibonacci. Công cụ Mở rộng Fibonacci yêu cầu một điểm thứ ba. Các mức mở rộng và hồi quy được vẽ từ điểm thứ ba này, nhưng dựa trên khoảng cách giữa hai điểm đầu tiên. Một cách sử dụng phổ biến của công cụ này là kết nối hai điểm đại diện cho các điểm cuối của một xu hướng lớn (hoặc sóng). Sau đó, chọn điểm thứ ba làm điểm kết thúc của một sự hồi quy của xu hướng đó. Các mức mở rộng sau đó được vẽ theo hướng của xu hướng ban đầu, từ điểm thứ ba, sử dụng khoảng cách giữa hai điểm đầu tiên làm cơ sở cho các mức mở rộng.

Công cụ vẽ và nút thanh công cụ Mở rộng Fibonacci hoạt động tương tự như công cụ vẽ Pitchfork của Andrew. Đầu tiên, nhấp vào nút Mở rộng Fibonacci trên thanh

công cụ. Sau đó, nhấp ba lần, mỗi lần là một trong các điểm được liên quan đến Mở rộng Fibonacci. Lần nhấp đầu tiên nên là điểm bắt đầu của trendline ban đầu. Lần nhấp thứ hai nên là điểm kết thúc của trendline ban đầu. Lần nhấp thứ ba nên là thanh kết thúc của kỳ rút lui.

Giống như Fibonacci Retracements, có rất nhiều linh hoạt thông qua tab cài đặt Fibonnacci có sẵn sau khi nhấp vào nghiên cứu bằng cú nhấp chuột phải và chọn "Properties" từ menu ngữ cảnh.

Mở Rộng Thời Gian Fibonacci

Công cụ Mở rộng Thời gian Fibonacci được sử dụng để chỉ định các dòng thẳng dọc tại các mức ngày/giờ được xác định là các giá trị có thể thay đổi xu hướng dựa trên phạm vi ngày/giờ trước đó của thị trường và một điểm mở rộng thứ ba.

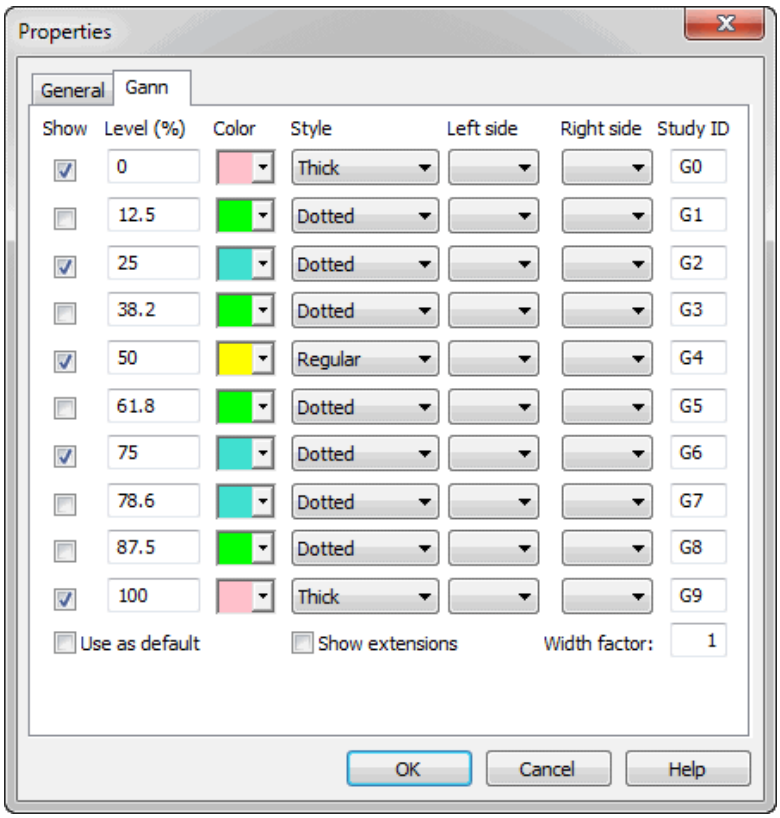
Công cụ mở rộng thời gian nên được sử dụng như sau. Đầu tiên, nhấp vào nút mở rộng thời gian Fibonacci trên thanh công cụ. Sau đó chọn điểm phạm vi đầu tiên (thông thường là đỉnh hoặc đáy chính của thị trường) bằng cách nhấp vào biểu đồ tại nơi bạn muốn phạm vi bắt đầu, sau đó di chuyển con trỏ chuột để chọn điểm phạm vi thứ hai bằng cách nhấp lại vào biểu đồ nơi bạn muốn phạm vi kết thúc. Đường mở rộng giờ sẽ được vẽ lên các thanh tương lai.

Như trong công cụ rút lui giá Fibonacci và công cụ mở rộng, bạn có quyền kiểm soát hoàn toàn các phần trăm được sử dụng trong công cụ Mở rộng Thời gian, và màu sắc của từng giá trị mở rộng thông qua hộp thoại Properties.

Gann Square Và Gann Fan

Gann Squares chỉ ra các chuyển động thời gian và giá có thể từ các điểm cao và thấp quan trọng. Để vẽ một Gann Square lên biểu đồ, di chuyển con trỏ trên biểu đồ đến điểm bắt đầu. Điểm bắt đầu thông thường là một Đỉnh hoặc Đáy quan trọng trên biểu đồ. Sau đó kéo chuột sang phải cho đến khi đạt được điểm kết thúc mong muốn. Các điểm bắt đầu và kết thúc sẽ là các góc của hình vuông. Điểm kết thúc thường nằm bên phải của các thanh biểu đồ. Hãy theo dõi các xu hướng thay đổi hướng tại các cấp độ Gann Square. Khi Gann Square được vẽ lên màn hình, góc của đường xu hướng điều khiển sẽ được hiển thị trong thanh trạng thái.

Cửa sổ thuộc tính



Cửa sổ thuộc tính được sử dụng để thay đổi các mức vuông, màu sắc, kiểu dáng, độ dày và mặc định. Nhấp vào bất kỳ mục hiển thị của Gann Square để thêm hoặc xóa các dòng. Nhấp vào hộp màu vuông để thay đổi màu dòng. Nhấp vào hộp tổ hợp kiểu để thay đổi kiểu dòng. Kiểm tra hộp Use as Default để lưu các cài đặt này như là mặc định cho tất cả các Gann Squares tiếp theo được vẽ. "Left side" và "Right side" là các cột kiểm soát hiển thị của văn bản xuất hiện ở phía trái và phía phải của các dòng Gann. Rỗng - có nghĩa là không có văn bản, % - có nghĩa là mức phần trăm, $ - có nghĩa là mức dollar (điểm). Cột "Study ID" định nghĩa định danh nghiên cứu có thể được sử dụng trong công thức tùy chỉnh của bạn để phát hiện chéo. Bạn có thể thay đổi ID này nếu cần bằng cách chỉnh sửa đơn giản các trường này. Để biết thêm thông tin về Study IDs, kiểm tra **Hướng dẫn: Sử dụng nghiên cứu trong công thức AFL**

Công cụ Tam giác

Công cụ Tam giác là tự giải thích. Vẽ một tam giác là dễ dàng: nhấp chuột trái vào điểm đầu tiên, giữ xuống và kéo đến điểm thứ hai, sau đó thả nút chuột và kéo đến điểm thứ ba và nhấp một lần. Tam giác điều khiển sẽ trở thành cái búa đinh.

Búa đinh của Andrews

Búa đinh của Andrews là một nghiên cứu sử dụng các đường xu hướng song song. Trong việc xây dựng nghiên cứu, các điểm bắt đầu được chọn. Đầu tiên là một đỉnh lớn hoặc đáy ở phía trái của biểu đồ hiển thị. Điểm bắt đầu thứ hai và thứ ba được

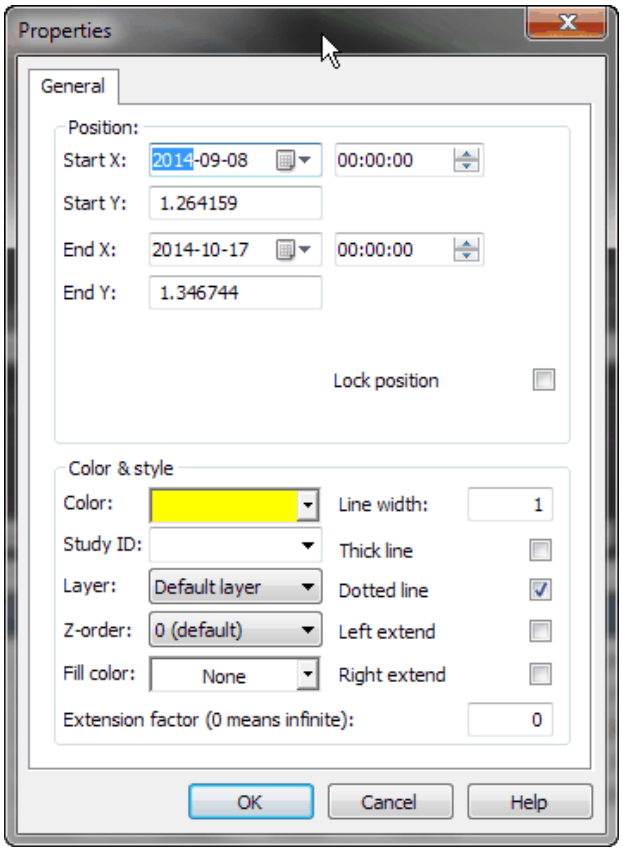
chọn là một đỉnh lớn và một đáy lớn ở phía phải của điểm đầu tiên. Sau khi tất cả các điểm bắt đầu đã được quyết định, AmiBroker vẽ một đường xu hướng từ điểm đầu tiên (nằm ở bên trái nhất) sao cho nó đi ngang qua trực tiếp giữa các điểm bên phải nhất. Đường này được gọi là tay cầm của cái búa đinh. Hai đường xu hướng thứ hai và thứ ba được vẽ bởi AmiBroker bắt đầu từ các điểm bắt đầu và song song với tay cầm. Dr. Andrews đã đề xuất rằng giá cố gắng đạt đến median line (hoặc tay cầm) khoảng 80% trong khi xu hướng giá còn hiệu lực. Điều này có nghĩa là trong khi xu hướng giá dài hạn cơ bản vẫn còn nguyên, Tiến sĩ Andrews tin rằng các xu hướng nhỏ hơn trong giá sẽ thu hút về đường trung bình, trong khi xu hướng giá lớn hơn vẫn giữ nguyên. Khi điều này không xảy ra, có thể đó là dấu hiệu cho thấy một sự đảo chiều trong xu hướng giá lớn hơn đang diễn ra hoặc cho thấy một thiên hướng mạnh mẽ hơn đang hoạt động trên thị trường. Khi giá không thể đến được đường trung bình từ cả hai phía, điều này thường biểu thị sự nhiệt tình tương đối của người mua và người bán và có thể dự đoán hướng đi lớn tiếp theo của giá. Nếu giá không đạt đến đường trung bình trong khi ở trên đường trung bình, điều này được coi là dấu hiệu tăng giá; và nếu không thể đạt đến đường trung bình từ dưới lên, điều này được coi là dấu hiệu giảm giá.

Việc sử dụng công cụ Andrews' Pitchfork tương tự như việc vẽ hình tam giác. Nhấp chuột trái vào điểm đầu tiên, giữ và kéo đến điểm thứ hai, sau đó thả nút chuột và kéo đến điểm thứ ba và nhấp chuột một lần. Hình tam giác điều khiển sẽ trở thành cái búa.

Công Cụ Cycles

Để sử dụng công cụ thời gian cycles, nhấp vào nút công cụ vẽ cycles trên thanh công cụ sau đó nhấp vào điểm đầu tiên của chu kỳ và kéo đến cuối chu kỳ. Hai điểm điều khiển này kiểm soát khoảng thời gian giữa các dòng chu trình. Khi bạn thả nút chuột, bạn sẽ nhận được một loạt các dòng song song với khoảng cách bằng nhau giữa chúng.

Cửa sổ Thuộc tính Dòng Nghiên cứu



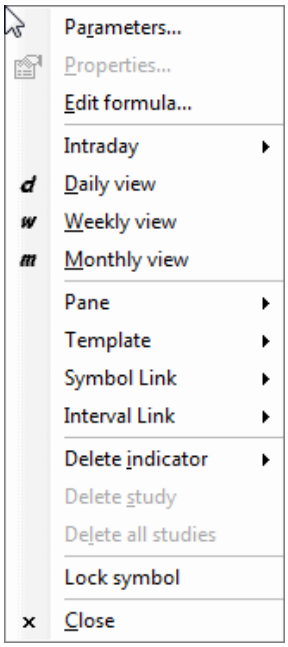
Trong cửa sổ thuộc tính nghiên cứu, bạn có thể chọn tọa độ bắt đầu và kết thúc cũng như màu sắc và kiểu dáng dòng. Bạn cũng có thể kích hoạt mở rộng tự động sang trái hoặc phải để dòng sẽ được mở rộng khi có báo giá mới.

Các trường sau đây có sẵn:

- Start X, Start Y, End X, End Y - tọa độ bắt đầu và kết thúc nghiên cứu
- Third X, Third Y - chỉ hợp lệ cho các nghiên cứu TRI-POINT như tam giác, pitchfork - tọa độ của điểm điều khiển thứ 3 của nghiên cứu
- Vị trí khóa - nếu trường này được đánh dấu, sẽ không thể thay đổi vị trí của nghiên cứu bằng cách sử dụng chuột
- Color - cho phép bạn thay đổi màu sắc của nghiên cứu
- StudyID - xác định ID nghiên cứu cho phép bạn tham chiếu đến nghiên cứu từ công thức AFL. Có thông tin chi tiết trong chương Sử dụng nghiên cứu trong công thức AFL của bạn.
- Layer - chỉ ra lớp mà nghiên cứu được đặt. Để tìm hiểu thêm về lớp, đọc Working with layers.
- Z-order - xác định thứ tự Z của dòng. Dòng, biểu đồ và đồ họa có thể được sắp xếp theo hướng "Z" bằng cách sử dụng Z-order. Tìm hiểu thêm về điều này trong Using Z-order tutorial.

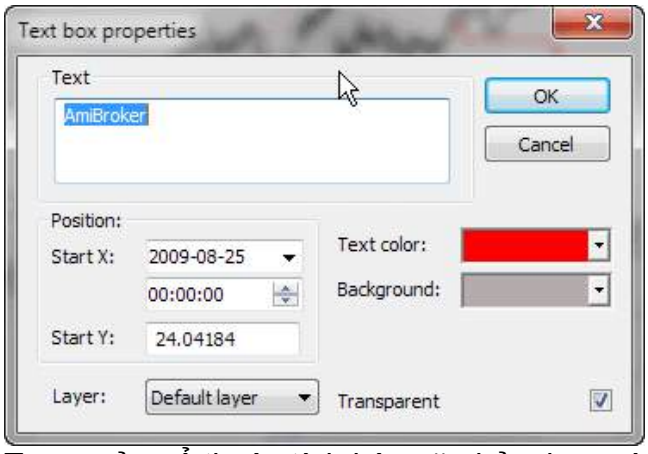
- Line width - (mới trong 5.90) xác định chiều rộng của dòng bằng pixel. Chiều rộng dòng mặc định là 1 pixel.
- Thick - gấp đôi chiều rộng của dòng. Chiều rộng được xác định bởi tham số Line width. Bật chế độ này khiến dòng rộng gấp đôi, vì vậy chiều rộng pixel thực sự sẽ là 2 * lineWidth.
- Left / Right Extend - bạn có thể chọn dòng có được mở rộng hay không Extension factor - (mới trong 5.90) - quyết định bao xa dòng được mở rộng sang trái/phải. Dòng được mở rộng theo hướng của "trục X" (tức là trục ngày/giờ). 0 (số không) có nghĩa là mở rộng vô hạn, một đơn vị đại diện khoảng cách trục X giữa điểm kết thúc và bắt đầu nghiên cứu. Giá trị phân số được cho phép. Phạm vi cho phép 0...25.5.

Cửa sổ thuộc tính Line study có thể truy cập từ menu nút chuột phải của cửa sổ biểu đồ. Khi bạn nhấp chuột phải vào một đường nghiên cứu, menu sau sẽ xuất hiện



Chỉ cần chọn Properties để hiển thị cửa sổ Line study.

Cửa sổ thuộc tính hộp văn bản



Trong cửa sổ thuộc tính hộp văn bản, bạn có thể thay đổi văn bản hiển thị trong hộp, chọn tọa độ bắt đầu cũng như màu sắc văn bản và nền, và kiểu trong suốt.

Các trường sau đây có sẵn:

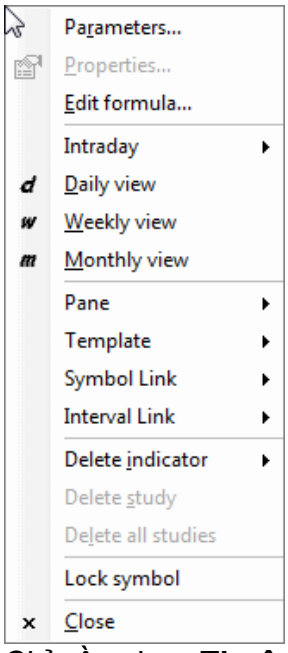
Tọa độ Bắt đầu X, Bắt đầu Y - tọa độ văn bản

Màu sắc - cho phép bạn thay đổi màu văn bản

Màu nền - cho phép bạn thay đổi màu nền

Lớp - cho biết lớp mà văn bản được đặt trên đó. Để tìm hiểu thêm về các lớp, hãy đọc phần **"Làm việc với các lớp."**

Cửa sổ thuộc tính hộp văn bản có thể truy cập từ menu chuột phải. Khi bạn nhấp chuột phải vào một hộp văn bản, menu sau sẽ xuất hiện:



Chỉ cần chọn **Thuộc tính** để hiển thị cửa sổ thuộc tính hộp văn bản.

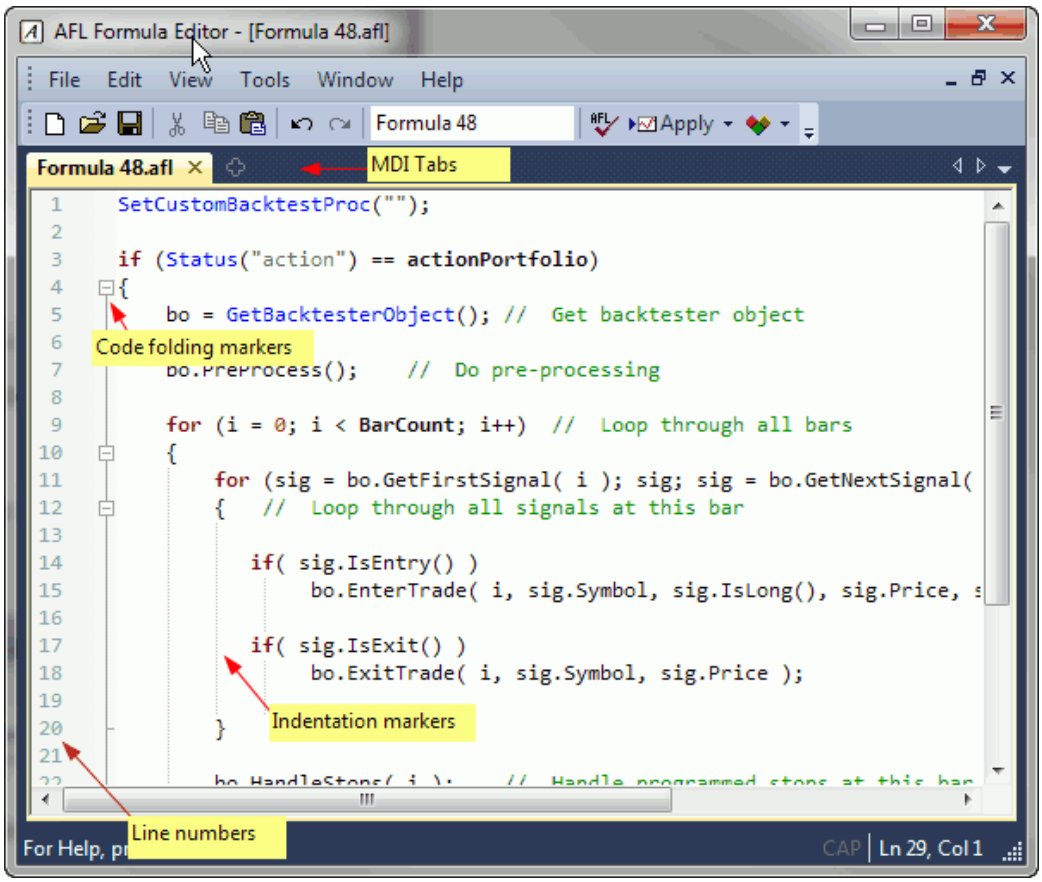
Trong cửa sổ propertites của hộp văn bản, bạn có thể thay đổi văn bản được hiển thị trong hộp, chọn tọa độ bắt đầu cũng như màu văn bản và màu nền và phong cách transparent.

Trình soạn thảo công thức

Trình soạn thảo Công thức AFL

- Trình soạn thảo công thức AFL mới có các tính năng sau:
- Đánh dấu cú pháp (đã được cải thiện trong phiên bản 5.80)
- Tự động khớp/nổi bật dấu ngoặc (MỚI trong 5.80)
- Tự động thụt lề (MỚI trong 5.80)
- Nhắc đặt thụt lề (MỚI trong 5.80)
- Tính năng tự động hoàn thiện câu lệnh ở hai chế độ (ngay lập tức (MỚI trong 5.80) và theo yêu cầu)
- Chú giải thông tin tham số
- Dòng đánh số lề và dòng margin chọn (MỚI trong 5.80)
- Gấp đoạn mã (MỚI trong 5.80)
- Báo lỗi ngay tại dòng bị lỗi (MỚI trong 5.80)
- Giao diện mới với khả năng làm việc ở cả hai chế độ MDI và chế độ khung nổi riêng, có thể di chuyển phía sau màn hình chính AmiBroker và quay trở lại (Window->Toggle Frame) (MỚI trong 5.80) hoặc giữ nguyên ở trạng thái chế độ luôn hiển thị (Window->Keep on top)
- Sao chép/Xóa khối chữ nhật (Sử dụng chuột và giữ phím Alt để đánh dấu khối chữ nhật) (Mới trong 5.80)
- Tự động viết hoa (thay đổi trường hợp)
- Không gian ảo (mới trong 5.80)
- Tính năng in ấn nâng cao (kèm đánh dấu cú pháp và tiêu đề/ chân trang)
- Đoạn mã mẫu (mới trong 5.80)
- Trình gỡ lỗi trực quan tích hợp (MỚI trong 6.10)
- Đánh dấu (MỚI trong 6.10)
- Tìm trong tệp (MỚI trong 6.10)
- Chú thích khối/Nhập khẩu comment (MỚI trong 6.10)
- Liên kết có thể nhấp trong comment (MỚI trong 6.10)

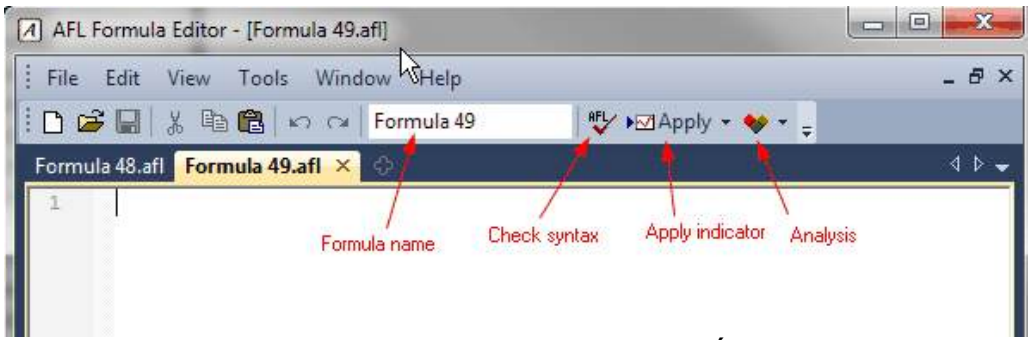
Những tính năng này đơn giản hóa việc viết công thức và cung cấp sự trợ giúp tức thì nên thời gian cần thiết để viết công thức giảm đáng kể.



Menu

Các tùy chọn trong menu Formula Editor được mô tả chi tiết trong chương **Menus: Formula Editor** của hướng dẫn.

Thanh công cụ



Thanh công cụ Trình soạn thảo Công thức cung cấp các nút sau:

- **New** - xóa nội dung trong cửa sổ formula editor
- **Open** - mở tệp công thức - Save - lưu công thức với tên hiện tại - Print - in công thức - Cut - cắt đoạn đã chọn và sao chép vào clipboard

- **Copy** - sao chép đoạn đã chọn vào clipboard
- **Paste** - dán nội dung clipboard hiện tại ở vị trí con trỏ hiện tại
- **Undo** - hoàn tác hành động gần đây (nhiều cấp độ)
- **Redo** - làm lại hành động gần đây (nhiều cấp độ)
- **Tên công thức** - một trường EDIT cho phép thay đổi tên file công thức, một khi bạn thay đổi tên ở đây và nhấn vào nút Save, công thức sẽ được lưu dưới tên mới và thay đổi sẽ được phản ánh trên thanh tiêu đề CAPTION BAR và trong thanh trạng thái STATUS BAR (Status bar hiển thị đường dẫn đầy đủ).
- **Check syntax** - kiểm tra công thức hiện tại vì lỗi
- **Apply indicator** - lưu công thức và áp dụng công thức hiện tại như một biểu đồ/chỉ số MỘT LẦN
- **Phân tích** - lưu công thức và chọn nó như công thức hiện tại trong cửa sổ Phân tích Tự động và lặp lại hoạt động Phân tích được sử dụng gần đây nhất (tức là Quét hoặc Khám phá hoặc Thử nghiệm lại hoặc Tối ưu hóa)

Sử dụng

- Việc sử dụng thông thường của Formula Editor như sau:
- mở Formula Editor - nhập công thức
- nhập tên có ý nghĩa mô tả mục đích của mã của bạn vào trường **Formula Name** (Tên công thức) rồi nhấn vào nút **Apply indicato**r (Áp dụng chỉ thị) (nếu bạn đã viết mã code chỉ thị)
  ..hoặc..
  nhấn nút **Analysis** (Phân tích) để hiển thị cửa sổ Phân tích Tự động (khi bạn đã viết mã khám phá/quét hoặc hệ thống giao dịch)

Syntax Highlighting

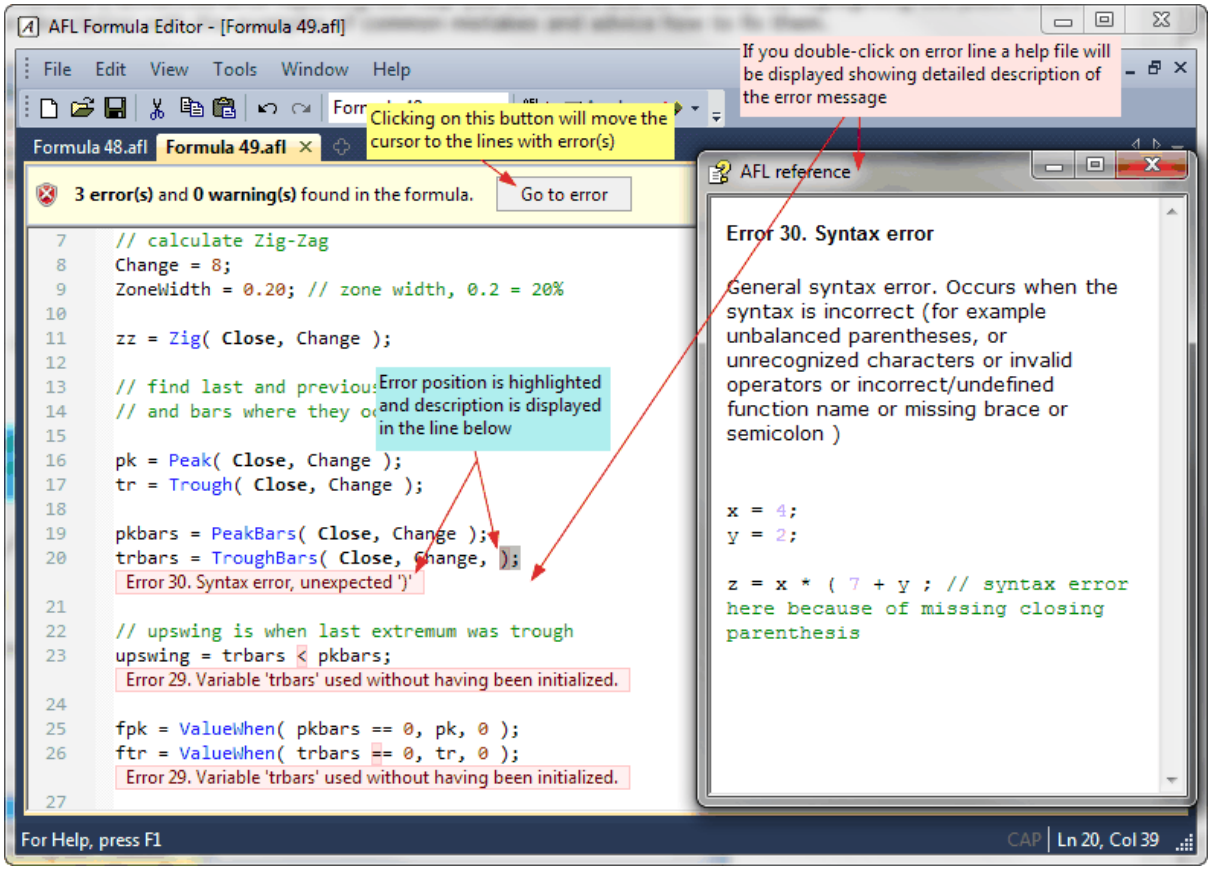
Trình biên soạn AFL của AmiBroker có tính năng làm nổi bật cú pháp có thể định rõ bởi người dùng, tự động áp dụng màu sắc và kiểu dáng do người dùng định rõ cho các yếu tố ngôn ngữ khác nhau như hàm và tên biến dự phòng, chuỗi, số, bình luận, v.v. Tính năng này đơn giản hóa quá trình soạn thảo mã. Bạn có thể chỉnh sửa lược đồ màu sắc trong cửa sổ Tùy chọn.

Enhanced Error Reporting

Khi bạn gặp lỗi trong công thức của mình, báo cáo lỗi nâng cao của AmiBroker sẽ giúp bạn xác định và sửa lỗi bằng cách làm nổi bật vị trí xảy ra lỗi và hiển thị mô tả lỗi mở rộng với các ví dụ về những sai lầm thường gặp và hướng dẫn cách khắc phục chúng. Trong phiên bản 5.80, mô tả lỗi được hiển thị trực tiếp trong mã.

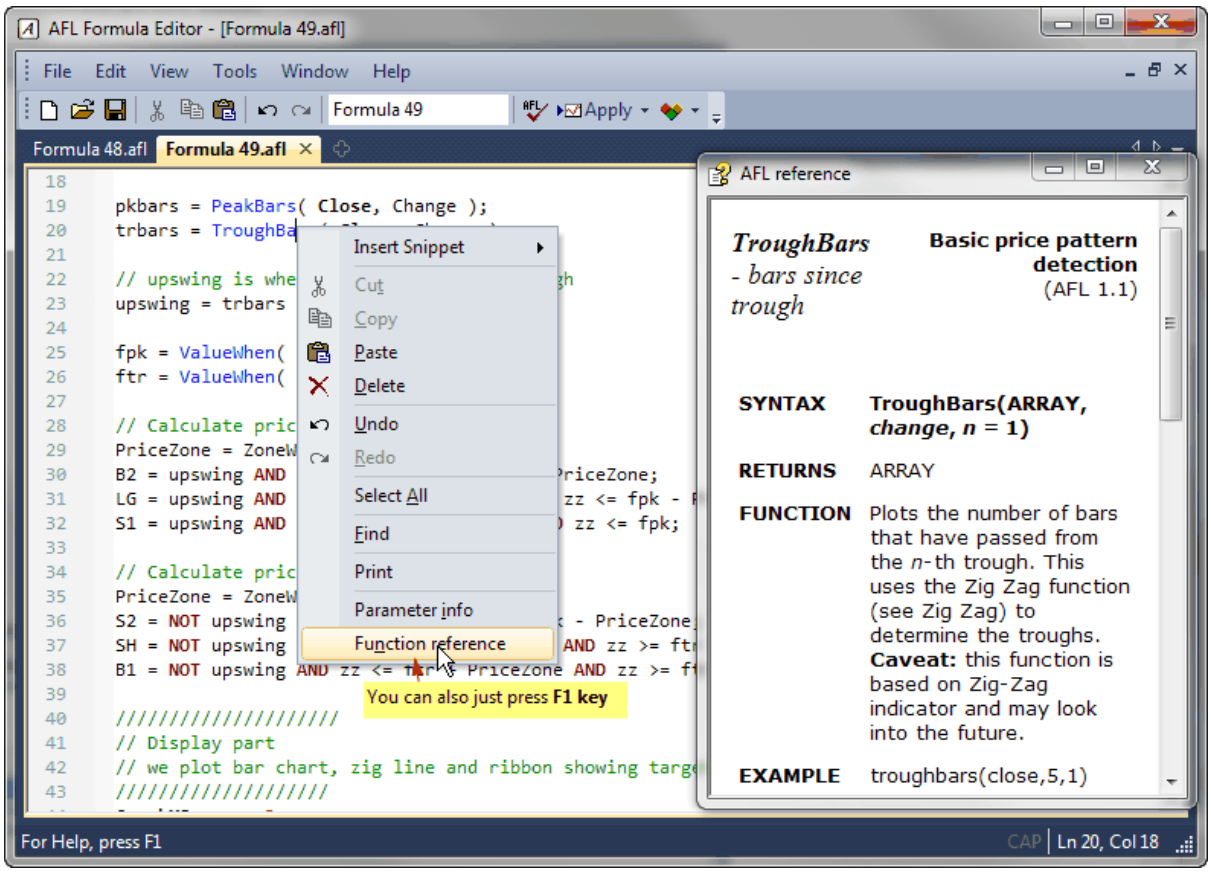
Một thanh thông báo hiển thị tổng số lỗi và/hoặc cảnh báo. Nếu bạn nhấn nút **"Đi đến lỗi"**, trình soạn thảo sẽ di chuyển con trỏ đến dòng có lỗi liên quan; nếu bạn nhấn lại, nó sẽ di chuyển đến lỗi tiếp theo và cứ tiếp tục như vậy. Nếu bạn đóng thanh thông báo bằng nút "X", tất cả các thông báo lỗi sẽ bị xóa (ẩn) khỏi màn hình.

Bạn có thể sử dụng menu **Chỉnh sửa -> Xóa Thông báo Lỗi (Ctrl+E)** để xóa thông báo lỗi riêng lẻ (trong dòng hiện tại).



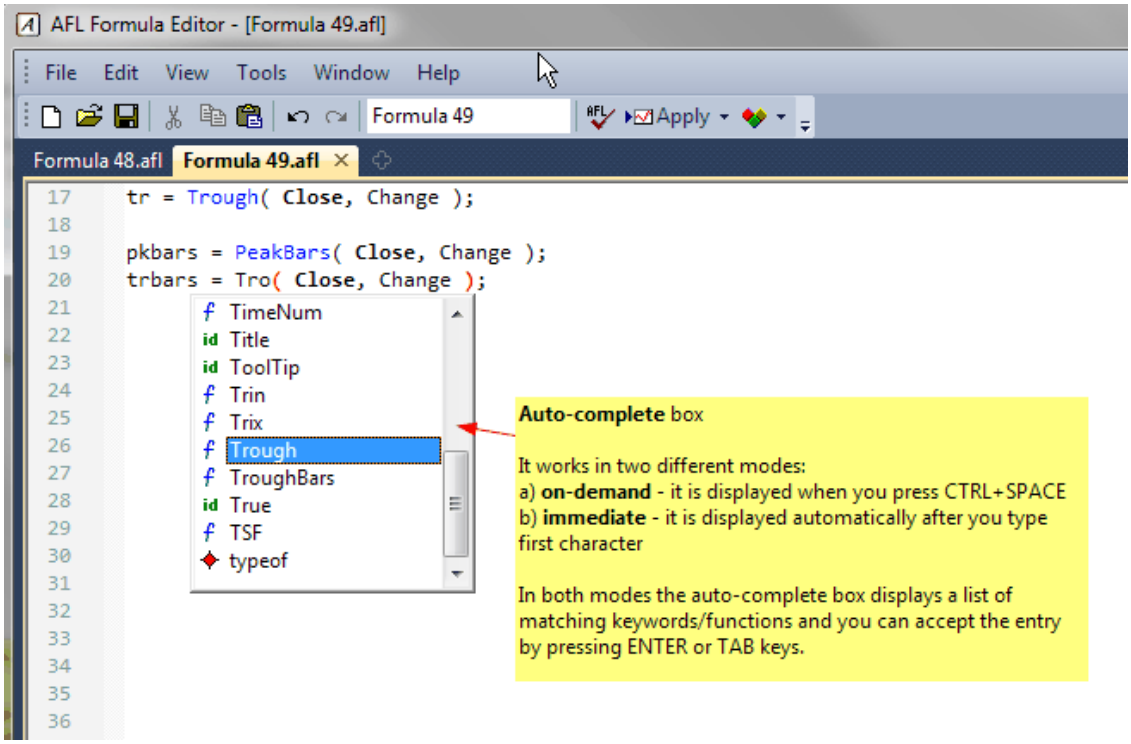
Trợ giúp Ngữ cảnh

Bạn có thể nhanh chóng hiển thị trang tham khảo chức năng AFL liên quan nếu bạn nhấn phím F1 hoặc chọn "Function reference" từ menu ngữ cảnh trong khi con trỏ đang ở trong hoặc ngay sau tên chức năng như hình dưới đây:



Tự động hoàn thành câu lệnh

Tính năng hoàn thành tự động (có sẵn khi bạn nhấn tổ hợp phím **CTRL+SPACE**) sẽ giúp bạn hoàn thành các hàm và biến dự trữ, hoặc hiển thị danh sách các ứng viên nếu những gì bạn đã gõ có nhiều hơn một kết quả phù hợp. Bạn có thể chọn mục trong danh sách bằng cách sử dụng các phím mũi tên lên/xuống hoặc bằng chuột. Để chấp nhận lựa chọn, hãy nhấn phím RETURN (ENTER). Bạn cũng có thể gõ một khoảng trắng ngay lập tức (đối với biến) hoặc dấu ngoặc mở (đối với hàm), và AmiBroker sẽ tự động hoàn thành từ đang chọn và đóng danh sách. Để bỏ danh sách, hãy nhấn phím ESC.



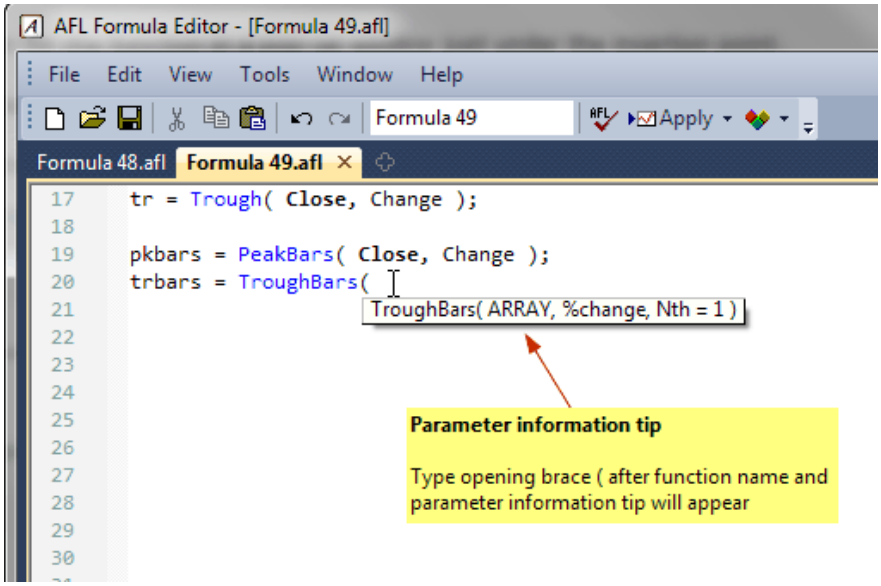
Thông tin tham số

Khi bạn gõ một hàm, bạn có thể hiển thị một Tool Tip chứa đầy đủ mẫu hàm, bao gồm cả các tham số. Tool Tip Thông tin Tham số cũng được hiển thị cho các hàm lồng nhau.Khi con trỏ của bạn nằm bên cạnh một hàm, hãy gõ dấu ngoặc mở như bạn thường làm để bao quanh danh sách tham số.

AmiBroker sẽ hiển thị khai báo đầy đủ của hàm trong một cửa sổ bật lên ngay dưới vị trí con trỏ.

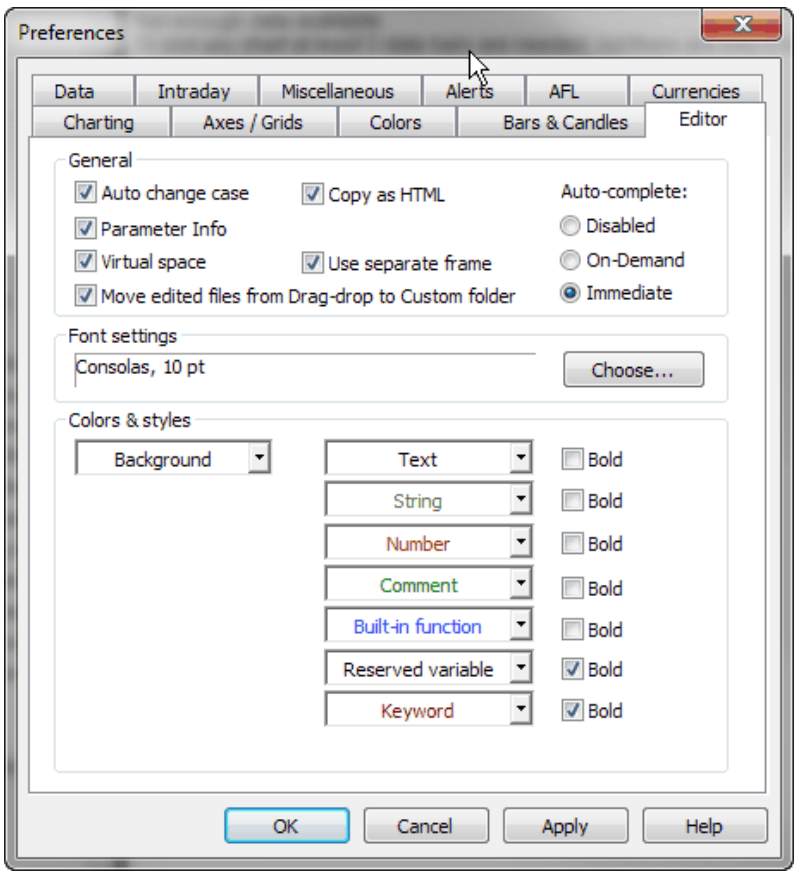
Gõ dấu ngoặc đóng sẽ làm ẩn danh sách tham số.

Bạn cũng có thể bỏ danh sách nếu nhấn phím mũi tên lên/xuống, nhấp chuột hoặc nhấn RETURN.



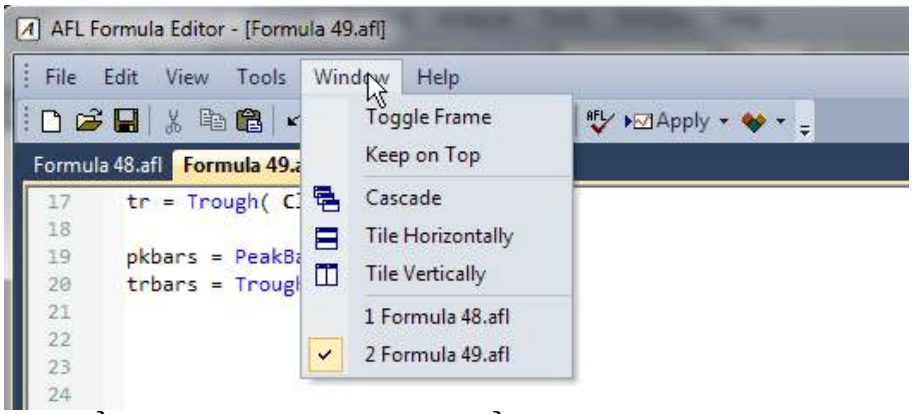
Cấu hình Trình soạn thảo

Các cài đặt của trình soạn thảo AFL có thể được thay đổi bằng cách sử dụng menu **Công cụ -> Tùy chọn**, trang **Trình soạn thảo.**



- Tự động thay đổi chữ hoa chữ thường - điều khiển việc trình soạn thảo tự động thay đổi chữ hoa của các từ khóa dự trữ (ví dụ, nếu người dùng gõ "valuewhen", nó sẽ tự động chuyển thành "ValueWhen").
- Thông tin tham số - điều khiển việc hiển thị các mẹo thông tin tham số.
- Không gian ảo - điều khiển việc có thể đặt con trỏ tự do ở bất kỳ vị trí nào sau khi kết thúc một dòng.
- Di chuyển tệp đã chỉnh sửa từ kéo-thả vào thư mục tùy chỉnh - thông thường, các công thức được tạo bằng cơ chế kéo-thả nằm trong thư mục kéo-thả ẩn. Nếu bạn muốn chỉnh sửa chúng, bạn có thể làm như vậy tại chỗ để chúng vẫn nằm trong thư mục kéo-thả (ẩn), hoặc bạn có thể chọn tự động di chuyển chúng vào thư mục 'tùy chỉnh'. Chuyển đổi này cho phép tự động di chuyển vào thư mục tùy chỉnh.
- Sao chép dưới dạng HTML - cho phép sao chép ở định dạng HTML, vì vậy mã AFL được sao chép với màu sắc. Nếu không, nó sẽ được sao chép dưới dạng văn bản thuần túy mà không có định dạng.
- Sử dụng khung riêng- nếu bật, trình soạn thảo AFL sẽ hiển thị trong một khung hoàn toàn riêng biệt, hoạt động như một ứng dụng riêng. Nếu tắt, thì trình soạn thảo AFL sẽ hiển thị như một tab MDI trong khung chính của AmiBroker (cùng với biểu đồ, cửa sổ phân tích, web, cửa sổ tài khoản, và nhiều hơn nữa). Mặc định là bật.
- Tự động hoàn thành: ở chế độ "Theo yêu cầu", danh sách tự động hoàn thành chỉ xuất hiện khi bạn nhấn Ctrl+SPACE. Ở chế độ "Ngay lập tức", danh sách tự động hoàn thành sẽ bật lên ngay khi bạn gõ ký tự đầu tiên (chữ cái) của định danh.

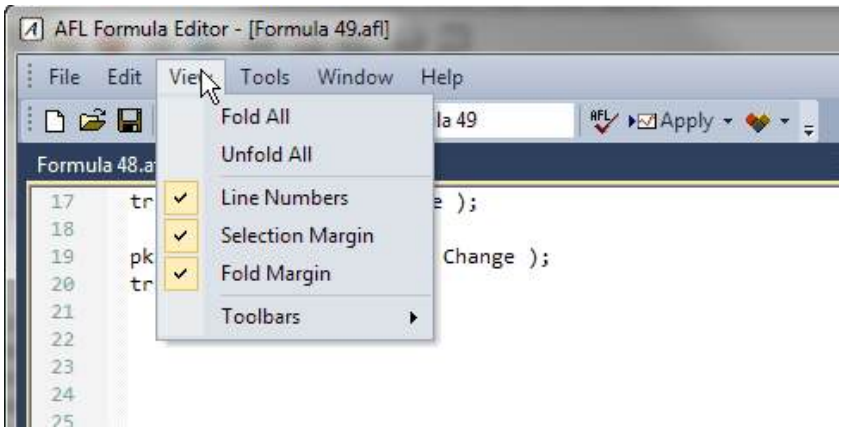
Kiểm soát cửa sổ



Cửa sổ Trình soạn thảo AFL có thể được đưa lên trên cùng hoặc ra phía sau như bất kỳ cửa sổ ứng dụng nào khác thông qua thanh tác vụ Windows. Ngoài ra, có menu **Cửa sổ -> Chuyển đổi Khung** (và phím tắt Ctrl+` , ` là phím tilde nằm ngay trên phím TAB trên hầu hết các bàn phím) cho phép nhanh chóng chuyển đổi giữa khung chính của AmiBroker và khung trình soạn thảo AFL.

Người dùng cũng có thể bật tính năng **Cửa sổ -> Giữ Ở Trên Cùng**, giữ cho cửa sổ trình soạn thảo nằm trên cùng khung chính của AmiBroker.

Khe lề

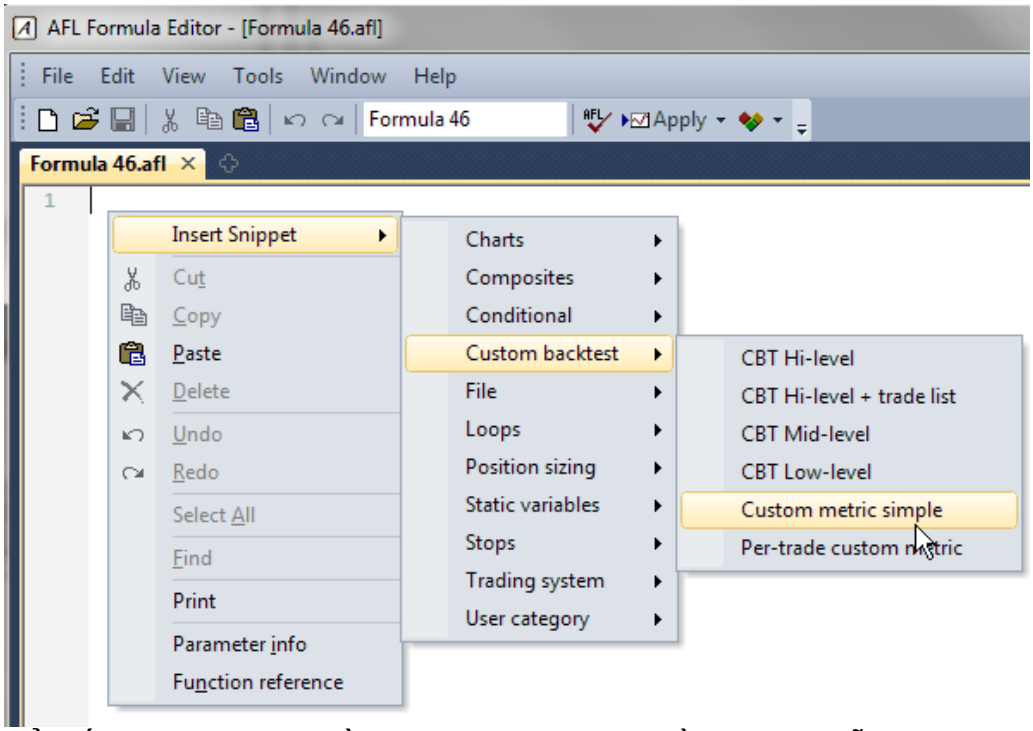


Khe lề số dòng, khe lề lựa chọn và **khe lề gập** có thể được bật/tắt bằng cách sử dụng menu Xem. Trong menu này cũng có các tùy chọn để gập/mở tất cả mã.

Mã đoạn

Mã đoạn là một đoạn mã AFL nhỏ. Nó có thể được chèn bằng cách:

- Nhấp chuột phải vào cửa sổ trình soạn thảo AFL và chọn menu "Chèn Đoạn Mã", hoặc
- Kéo một đoạn mã từ cửa sổ Đoạn Mã, hoặc
- Gõ kích hoạt bàn phím (chẳng hạn như @for) trong trình soạn thảo.



Để biết thêm thông tin về Mã đoạn, hãy xem phần Hướng dẫn: Sử dụng Mã đoạn.

Đánh dấu

Một đánh dấu là một chỉ báo cho phép bạn nhanh chóng nhảy đến dòng đã đánh dấu trong công thức.

Để đặt / xóa dấu trang, hãy sử dụng menu **Edit->Bookmark->Insert /Remove** hoặc phím **Ctrl + F2**. Khi bạn đặt dấu trang, một dấu đánh dấu màu xanh nhạt sẽ xuất hiện ở lề bên trái. Giờ đây, với một vài dấu trang đã đặt, bạn có thể di chuyển nhanh (nhảy) giữa chúng bằng phím **F2** - di chuyển đến dấu trang tiếp theo, **Shift + F2 - di chuyển** đến dấu trang trước đó. Bạn có thể xóa tất cả dấu trang bằng **Edit->Bookmark->Clear All.** Hãy lưu ý rằng dấu trang được lưu cùng với thông tin gỡ lỗi (breakpoints / watches) trong một tệp riêng với phần mở rộng .dbg.

Tìm trong các tệp

Chức năng Tìm trong các Tệp có sẵn từ menu **Edit->Find in Files** và cho phép bạn tìm kiếm tất cả các tệp trong các thư mục đã chọn cho văn bản đã chỉ định. Kết quả tìm kiếm được trình bày trong cửa sổ Output (**Window->Output menu**) như sau:

Filepath\Filename.afl(number_line): Line contents.

Bây giờ nếu bạn nhấp đúp vào dòng output đó, AmiBroker sẽ tự động mở tệp đã cho trong tab biên tập mới.

Bình luận / Bỏ Bình luận Khối

Trình chỉnh sửa AFL giờ đây cung cấp một công cụ để tự động chú thích các khối dòng đã chọn. Công cụ thêm một // (dấu gạch chéo kép) vào mỗi dòng trong khu vực đã chọn. Để sử dụng công cụ này, chọn một số dòng bằng cách sử dụng chuột hoặc con trỏ + phím SHIFT, sau đó sử dụng **Edit->Line Comment**. Tất cả các dòng trong phạm vi đã chọn sẽ được chú thích với // (dấu gạch chéo kép). Để bỏ chú thích, chọn phạm vi một lần nữa và chọn **Edit->Line Comment** một lần nữa.

Liên kết Có thể nhấp vào trong Nhận xét

Trình chỉnh sửa AFL giờ đây hỗ trợ các liên kết JavaDoc / Doxygen-style có thể nhấp vào trong doc comment. Để sử dụng các liên kết trong bình luận, bạn phải đặt lệnh @ link theo sau là đường dẫn đến tệp hoặc URL bên trong bình luận JavaDoc / Doxygen style:

- Nhận xét nhiều dòng bắt đầu bằng / như sau:

  /

  @link readme.html

  @ link http://www.amibroker.com

  */

  (lưu ý dấu hoa thị kép sau dấu gạch chéo ban đầu)

- Bình luận một dòng bắt đầu với dấu gạch chéo ba lần
- /// @link readme.html
- /// @link c:\program files\amibroker\readme.html
- /// @link http://www.amibroker.com.

Bây giờ khi bạn di chuyển chuột qua lệnh @link, bạn sẽ thấy gạch chân cho thấy nó có thể nhấp vào. Nó phản ứng với nhấp đúp (không phải nhấp đơn). Khi bạn nhấp đúp nó, tài liệu sẽ mở ra.

Lệnh @link có thể mở trang web, tệp tin địa phương (cả đường dẫn tương đối và tuyệt đối đều được hỗ trợ) bằng chương trình đã đăng ký trên Windows để mở loại tệp đã cho. Vì vậy, nếu bạn sử dụng:

/// @link something.doc

Sau đó, MS Word sẽ được sử dụng.

/// Đường dẫn @link test.xls

sẽ mở file test.xls trong Excel. Các đường dẫn tương đối ám chỉ thư mục làm việc của AmiBroker. Các file Html sẽ được mở bằng trình duyệt mặc định, các file txt

thường được mở với Notepad (hoặc bất kỳ ứng dụng nào bạn dùng). Nếu file không tồn tại, bạn sẽ nhận được thông báo lỗi.

Cửa sổ Đoạn mã

Đoạn mã là những phần nhỏ của mã AFL có thể tái sử dụng, thông tin chi tiết về cách sử dụng Đoạn mã có thể tìm thấy trong hướng dẫn này.

Cửa sổ Đoạn mã có sẵn trong trình soạn thảo AFL mới. Nó có thể được hiển thị/ẩn bằng menu Window.

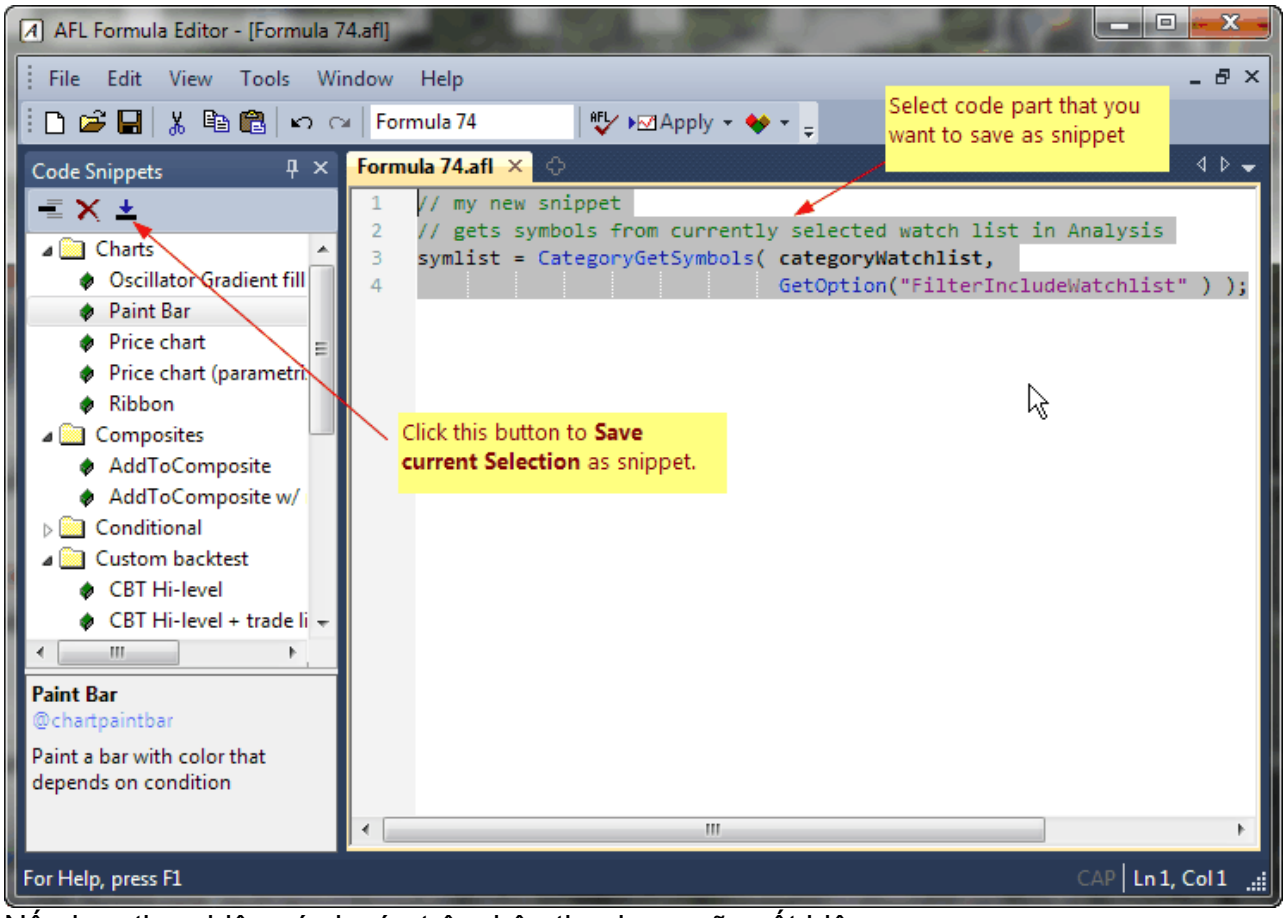
Chèn Đoạn mã

Để chèn một đoạn mã đã tồn tại, kéo và thả đoạn mã từ danh sách Đoạn mã vào Trình soạn thảo AFL, hoặc nhấp đúp vào đoạn mã.

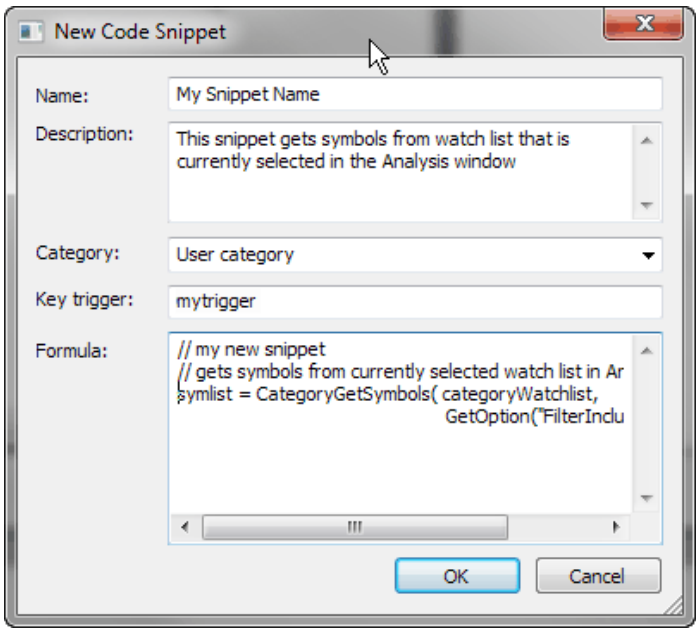
Tạo Đoạn mã của Riêng bạn

Để tạo đoạn mã của riêng bạn, làm theo các bước sau:

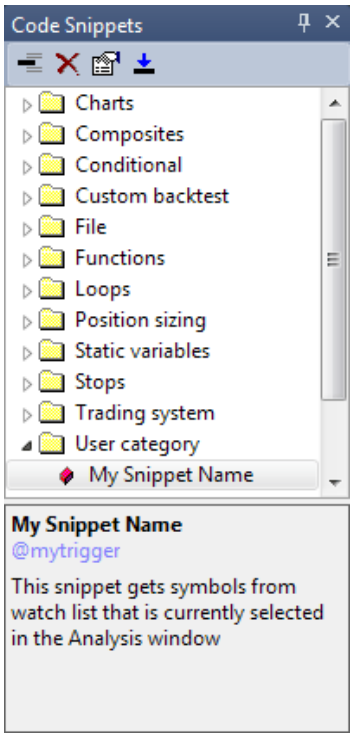
1. đánh máy mã bạn muốn
2. chọn (đánh dấu) mã bạn muốn đặt trong một đoạn mã
3. nhấn nút Lưu lựa chọn dưới dạng nút đoạn mã trong cửa sổ Đoạn mã



Nếu bạn thực hiện các bước trên, hộp thoại sau sẽ xuất hiện:



Bây giờ bạn cần nhập **Tên** của đoạn mã, **Mô tả** và **Danh mục**. **Danh mục** có thể được chọn từ các mục đã tồn tại (sử dụng hộp thả xuống), hoặc tên của danh mục mới có thể được gõ vào ô danh mục.**Ô kích hoạt bằng phím** là tùy chọn và chứa kích hoạt tự động hoàn thiện đoạn mã (được mô tả ở trên). **Ô Công thức** là mã đoạn văn bản đó. Khi bạn nhập tất cả các trường và nhấn **OK**, đoạn mã mới của bạn sẽ xuất hiện trong danh sách.

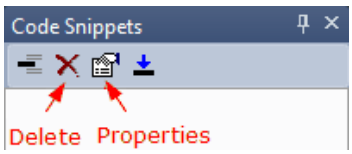


Từ đó trở đi, bạn có thể sử dụng đoạn mã của riêng mình cùng một cách với đoạn mã hiện có. Có lẽ phương pháp tiện nhất là sử dụng kéo-thả từ danh sách đến trình soạn thảo AFL.

Như bạn có thể đã nhận thấy, đoạn mã do người dùng định nghĩa được đánh dấu bằng hộp màu đỏ trong danh sách Đoạn mã. Chỉ có đoạn mã do người dùng định nghĩa mới có thể bị ghi đè và/hoặc xóa.

Chỉnh sửa Snippet

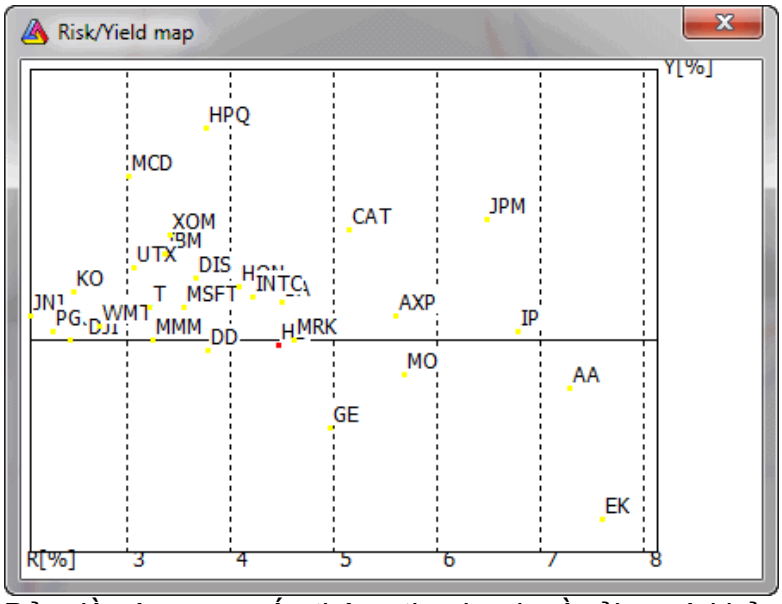
Để chỉnh sửa snippet do người dùng định nghĩa hiện có, bạn có thể tuân theo các bước trên và đưa ra tên hiện có. AmiBroker sau đó sẽ hỏi bạn nếu bạn muốn ghi đè snippet hiện có, hoặc bạn chỉ cần nhấp vào nút **Properties** và chỉnh sửa trực tiếp snippet, mà không cần thêm lại nó.



Xóa Snippet

Để xóa một snippet, chọn snippet bạn muốn xóa từ danh sách và nhấn nút **Delete** (X) ở cửa sổ Code Snippet.

Cửa sổ Bản đồ Rủi Ro - Lợi Nhuận



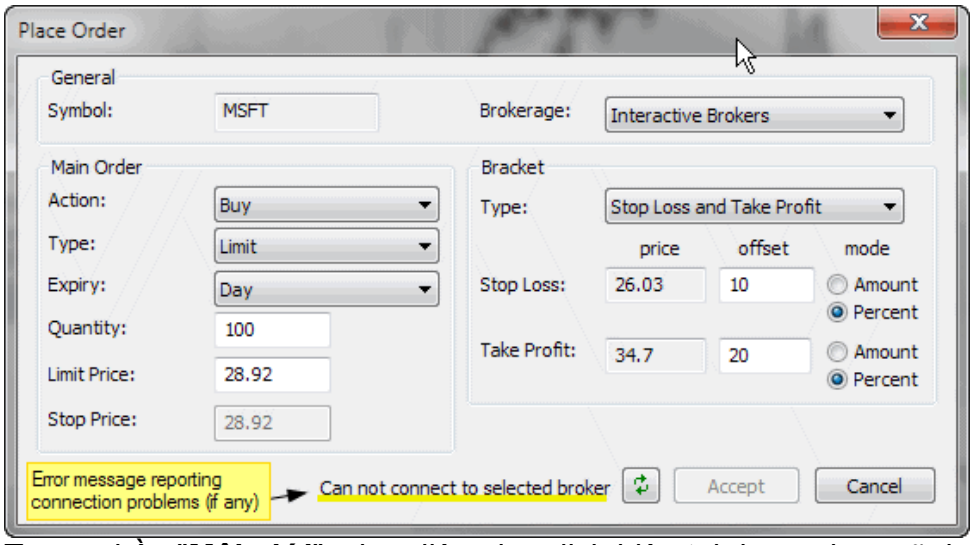
Bản đồ này cung cấp thông tin nhanh về rủi ro và khả năng sinh lợi. Lợi nhuận là tỷ lệ phần trăm trung bình hàng tuần, trong khi rủi ro là độ lệch chuẩn của tỷ lệ phần trăm lợi nhuận hàng tuần. Trục X thể hiện rủi ro, còn trục Y thể hiện lợi nhuận. Vì vậy, ở phần trên của bản đồ, chúng ta có các ký hiệu với lợi nhuận tốt nhất, trong khi rủi ro tăng từ bên trái sang bên phải của bản đồ.

Ký hiệu đã chọn được đánh dấu bằng màu khác, và bạn có thể phóng to phần của bản đồ bằng cách nhấn nút chuột trái và kéo để tạo hình chữ nhật phóng to.

Hộp thoại Đặt Lệnh

LƯU Ý: Chức năng này yêu cầu thêm giao diện giao dịch tự động có thể được tải xuống riêng.

Để đặt lệnh từ biểu đồ, trước tiên hãy chọn menu **Chèn -> Đặt Lệnh Mua hoặc Chèn -> Đặt Lệnh Bán**, hoặc nhấn vào các nút tương ứng trên thanh công cụ Đặt Lệnh. Sau đó, AmiBroker sẽ cho phép bạn vẽ một đường ngang bằng con trỏ chuột trên biểu đồ. Chỉ cần nhấp chuột trái lên biểu đồ và giữ xuống—một đường ngang sẽ hiện ra đánh dấu mức giá. Khi bạn di chuyển đường đến mức chính xác, hãy thả chuột trái để đặt lệnh (hộp thoại sau đó sẽ xuất hiện), hoặc nhấn phím ESC để hủy toàn bộ thao tác.



Trong phần "**Môi giới**", giao diện giao dịch hiện tại được chọn sẽ được hiển thị. Sau khi cài đặt giao diện giao dịch tự động của Interactive Brokers (từ http://www.amibroker.com/at/), văn bản "Interactive Brokers" sẽ xuất hiện. Nếu không có giao diện giao dịch nào được cài đặt, hộp kết hợp sẽ để trống. Nếu bạn đã cài đặt các giao diện giao dịch khác, chúng sẽ xuất hiện trong danh sách.

Trong trường "**Hành động**", bạn có thể chọn **Mua hoặc Bán** - lưu ý rằng tùy chọn đã chọn trước đó từ menu hoặc thanh công cụ sẽ được chọn mặc định.

Trong trường **"Loại**", bạn có thể chọn loại lệnh (Thị trường, Giới hạn, Dừng, Giới hạn Dừng, v.v.), theo mặc định, lệnh "Giới hạn" sẽ được chọn.

Trong trường "**Thời hạn**", bạn có thể chọn thời gian mà lệnh đã cho sẽ có hiệu lực. Hiện tại, có sẵn các tùy chọn Ngày và Hết Hạn.

Trong trường "**Số lượng**", bạn có thể nhập số lượng cổ phiếu/hợp đồng để mua/bán.

Trong trường "**Giá Giới hạn**", bạn có thể nhập giá giới hạn cho lệnh - AmiBroker sẽ tự động điền giá trị đã chọn trên biểu đồ.

Trong trường "**Giá Dừng"**, bạn có thể nhập giá dừng cho các lệnh Dừng và Giới hạn Dừng.

Trong nhóm "**Bảng giá"**, bạn có thể chọn các lệnh tự động bổ sung. Lệnh bảng giá là các lệnh dừng lỗ và/hoặc chốt lời "con" liên kết với lệnh "Cha" chính và hoạt động như nhóm OCA (một lệnh hủy lệnh khác) (do đó, khi lệnh chốt lời được kích hoạt, lệnh dừng lỗ tương ứng sẽ bị hủy). Giá bảng giá được tính toán tự động từ giá Giới hạn. Khoảng cách giữa giá giới hạn và lệnh dừng lỗ/chốt lời các mức được xác định bởi các trường "offset" phù hợp. Khoảng cách có thể được thể hiện bằng lượng (đô la) hoặc phần trăm của giá giới hạn.

Tất cả các giá đều được làm tròn tùy thuộc vào thiết lập TickSize hiện tại của biểu tượng (xem cửa sổ Thông tin). Nếu TickSize không được định rõ (tức là bằng không), thì AmiBroker giả sử là 0.01 (một cent).

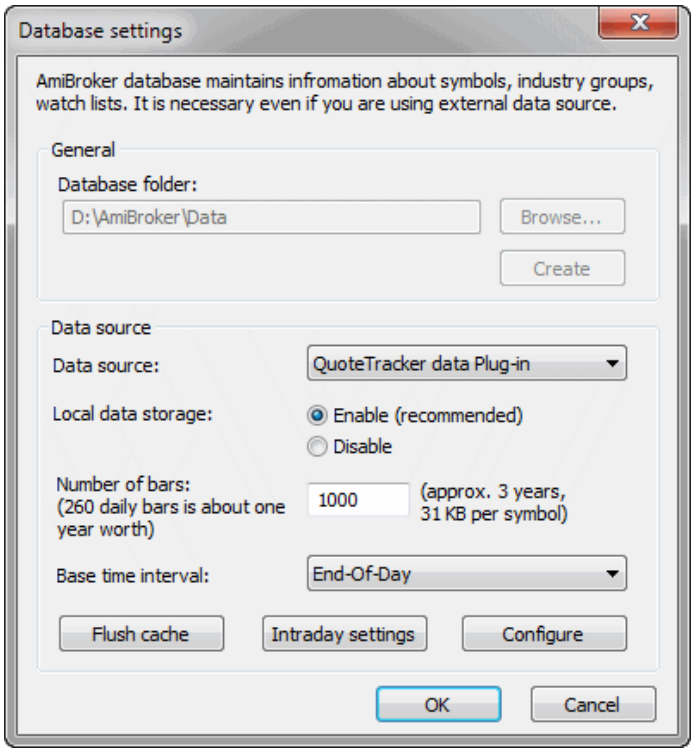
Trường Trạng thái (được tô sáng màu vàng) - hiển thị trạng thái kết nối giữa AmiBroker và giao diện giao dịch. Mọi lỗi kết nối sẽ được hiển thị ở đây và trong trường hợp lỗi, AmiBroker sẽ vô hiệu hóa nút "Accept" và sẽ cố gắng kết nối lại sau mỗi 5 giây. Bạn cũng có thể tự chủ động cố gắng kết nối lại bằng cách nhấn nút có hai mũi tên màu xanh.

Khi trường trạng thái hiển thị "Connected" thì nút Accept được kích hoạt và bạn có thể nhấn nó để đặt lệnh. Lưu ý rằng hiện nay giao diện đặt lệnh với cờ Transmit được đặt thành FALSE. Điều này có nghĩa là các lệnh KHÔNG thực sự được truyền đến sàn mà chờ truyền thủ công trong TWS. Đây là biện pháp an toàn.

Khi hộp thoại được đóng bằng cách nhấn Accept, đường kẻ ngang hiển thị giá giới hạn đã nhập sẽ vẫn còn trên biểu đồ. Theo mặc định, bạn không thể di chuyển nó, nhưng bạn có thể xóa nó bằng cách chọn nó và nhấn "DEL" (Delete) key.

Cài đặt Cơ sở Dữ liệu

Cửa sổ này cho phép bạn xác định các cài đặt theo cơ sở dữ liệu. Bạn có thể truy cập thông qua menu Tệp -> Cài đặt Cơ sở Dữ liệu.



QUAN TRỌNG: Các cài đặt theo cơ sở dữ liệu trong cửa sổ này sẽ có ưu tiên hơn so với các giá trị mặc định có thể định nghĩa trong cửa sổ Tùy chọn. Xem phần giải thích trong Hướng dẫn: Hiểu các khái niệm cơ sở dữ liệu.

Cửa sổ cài đặt cơ sở dữ liệu được chia thành hai phần: Tổng quát và Nguồn dữ liệu.

Phần cài đặt tổng quát **chỉ được kích hoạt trong quá trình tạo cơ sở dữ liệu (Tệp -> Cơ sở dữ liệu Mới).** Khi cơ sở dữ liệu đã được tạo, các điều khiển này sẽ trở nên bị vô hiệu hóa.

- **Duyệt... - cho phép bạn duyệt để chọn thư mục nơi cơ sở dữ liệu mới sẽ được tạo.**
- **Tạo - nhấp vào nút này để tạo cơ sở dữ liệu trong thư mục được chỉ định trong trường chỉnh sửa Thư mục Cơ sở dữ liệu.**

Để biết thêm chi tiết về việc tạo cơ sở dữ liệu mới và làm việc với nguồn dữ liệu cụ thể, vui lòng tham khảo phần Hướng dẫn.

Phần nguồn dữ liệu sẽ được kích hoạt khi cơ sở dữ liệu đã được tạo, và nó có thể được sử dụng để thay đổi các cài đặt cho các cơ sở dữ liệu đã tồn tại (thông qua menu **Tệp -> Cài đặt Cơ sở Dữ liệu).** Các điều khiển sau đây có sẵn:

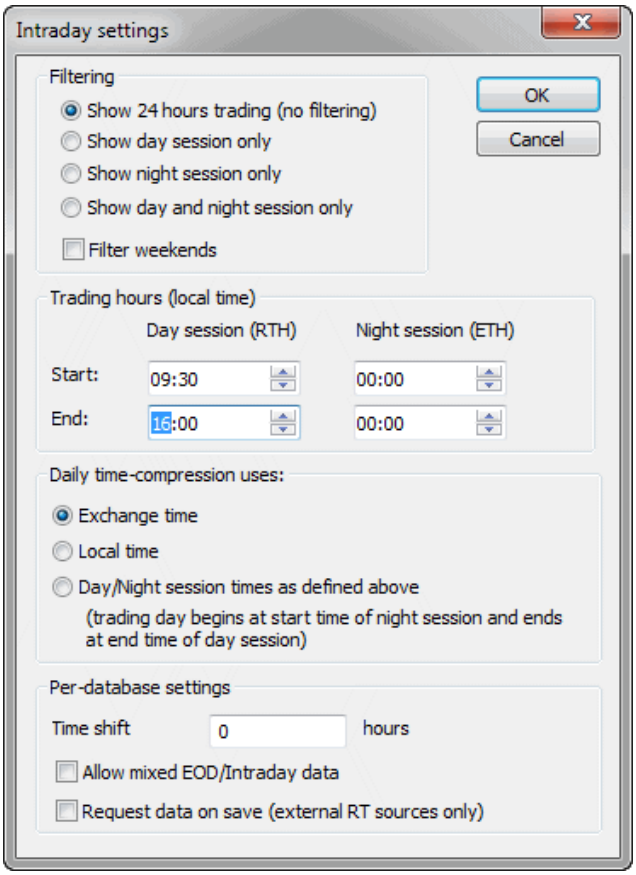
- **Nguồn dữ liệu**: xác định nguồn dữ liệu, có thể là:
    - (cục bộ)- có nghĩa là không có nguồn bên ngoài nào được sử dụng và dữ liệu được duy trì bởi AmiBroker. Cơ sở dữ liệu như vậy có thể được cập nhật bằng cách sử dụng AmiQuote (Công cụ -> Tự động cập nhật báo giá) hoặc sử dụng nhập ASCII - Trình hướng dẫn Nhập khẩu, trình nhập Metastock, hoặc kịch bản.
    - Nguồn dữ liệu bên ngoài (một trong: eSignal, myTrack, QuoteTracker, Quotes Plus, TC2000/TCNet, FastTrack, Metastock) - có nghĩa là dữ liệu được truy xuất trực tiếp từ cơ sở dữ liệu/bên ngoài. Cơ sở dữ liệu như vậy được cập nhật tự động thông qua plugin và không yêu cầu người dùng hành động trong AmiBroker. Ví dụ, nếu bạn sử dụng TC2000 làm nguồn dữ liệu, tất cả dữ liệu có trong hệ thống TC2000 tự động có sẵn trong AmiBroker. Để biết thêm chi tiết, vui lòng đọc Hướng dẫn: Hiểu về khái niệm cơ sở dữ liệu.
- **Lưu trữ dữ liệu cục bộ**: quyết định xem dữ liệu từ nguồn dữ liệu bên ngoài có nên được lưu trữ/cached trong các tệp riêng của AmiBroker hay không. Nếu "Enabled" thì dữ liệu bên ngoài được cached trong các tệp cục bộ. Nếu "Disabled" thì các tệp cục bộ không lưu trữ dữ liệu bên ngoài. Việc chuyển sang "Enabled" là bắt buộc đối với hầu hết các nguồn dữ liệu thời gian thực như eSignal, myTrack, QuoteTracker. Cài đặt này không có tác động nếu nguồn dữ liệu được đặt thành (cục bộ).
- **Số lượng thanh cần tải** - xác định bao nhiêu thanh cần được tải từ nguồn dữ liệu bên ngoài và giữ lại trong AmiBroker. Ví dụ: EOD 10 năm: 2600, 60 ngày intraday 1 phút: 30000 (xấp xỉ). Cài đặt này không có tác động nếu nguồn dữ liệu được đặt thành (cục bộ).
- **Khoảng thời gian cơ bản** - định rõ khoảng thời gian 'cơ bản' nào được sử dụng trong cơ sở dữ liệu này. Đối với nguồn dữ liệu thời gian thực, điều này nên được thiết lập một lần khi tạo cơ sở dữ liệu. Điều này là do các nguồn thời gian thực cần thu thập RT ticks và đóng gói chúng (nén thời gian) thành các thanh khoảng thời gian. Cài đặt này xác định 'grain' tối thiểu. Đối với các nguồn EOD, nó là (Cuối ngày (hàng ngày). Đối với các nguồn thời gian thực, điều này nên được đặt thành 1 phút hoặc cao hơn. Đối với một số nguồn thời gian thực (như eSignal), điều này cũng có thể được đặt thành tick, 5 giây hoặc 15 giây.

  Lưu ý là bạn không thể sử dụng biểu đồ intraday và/hoặc phân tích cho tới khi khoảng thời gian cơ bản được đặt thành gì đó dưới khoảng thời gian cuối ngày (có thể là 1 phút chẳng hạn). Để biết thêm chi tiết, vui lòng đọc Hướng dẫn: Hướng dẫn vẽ biểu đồ cơ bản.

- Flush cache - cho phép đẩy mạnh việc xóa cache và lấy dữ liệu mới từ plugin

- Cấu hình - cho phép hiển thị hộp thoại cấu hình cụ thể của nguồn dữ liệu xem Hướng dẫn để biết chi tiết về cách cấu hình các nguồn dữ liệu khác nhau.
- Cấu hình Intraday - cho phép xác định các cài đặt theo cơ sở dữ liệu cho các cơ sở dữ liệu intraday (xem bên dưới).

Cửa sổ Cài đặt Intraday



- **Lọc** - công cụ này giúp kiểm soát việc hiển thị dữ liệu trong ngày. AmiBroker thu thập tất cả dữ liệu nhưng chỉ hiển thị những dữ liệu nằm trong khoảng thời gian giao dịch được chọn. Lưu ý rằng chức năng này ảnh hưởng đến tất cả các biểu đồ và cửa sổ ngoại trừ Quote Editor, nơi luôn hiển thị tất cả dữ liệu có sẵn.

Hiển thị 24 giờ giao dịch (không lọc) - tất cả dữ liệu đều được hiển thị (không lọc)

Chỉ hiển thị phiên giao dịch ngày - chỉ hiển thị dữ liệu trong khoảng thời gian bắt đầu và kết thúc phiên giao dịch ngày (RTH)

Chỉ hiển thị phiên giao dịch đêm - chỉ hiển thị dữ liệu trong khoảng thời gian bắt đầu và kết thúc phiên giao dịch đêm (ETH)

Chỉ hiển thị phiên ngày và đêm - chỉ hiển thị dữ liệu trong khoảng thời gian bắt đầu/kết thúc phiên giao dịch ngày hoặc đêm

- **Lọc cuối tuần** - khi được chọn, AmiBroker thu thập nhưng không hiển thị dữ liệu từ cuối tuần. Khi không được chọn, dữ liệu đó sẽ được thu thập và hiển thị.
- **Thời gian bắt đầu/ kết thúc giao dịch** - định nghĩa thời gian bắt đầu và kết thúc giao dịch cho các phiên ngày (RTH) và đêm (ETH) riêng biệt. Lưu ý rằng thời gian này phải được chỉ định theo múi giờ của bạn.

Sử dụng nén thời gian hàng ngày - quyết định cách AmiBroker thực hiện nén thời gian từ intraday sang hàng ngày

Thời gian Sở giao dịch - dữ liệu hàng ngày được xây dựng từ các thanh intraday bắt đầu từ 00:00 và kết thúc tại 23:59 trong múi giờ của SỞ GIAO DỊCH (hoặc nguồn dữ liệu)

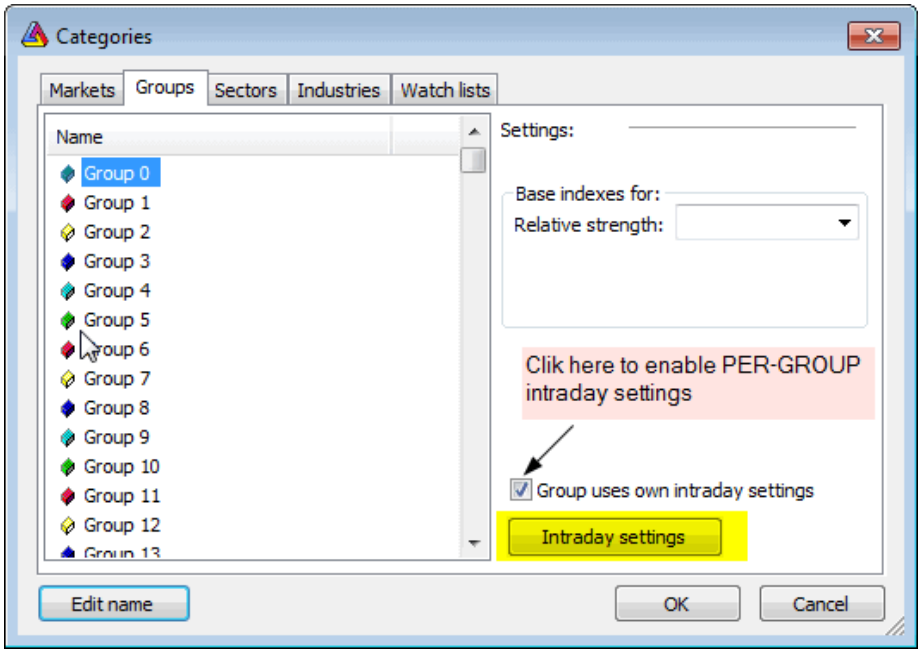
Thời gian địa phương - dữ liệu hàng ngày được xây dựng từ các thanh intraday bắt đầu từ 00:00 và kết thúc lúc 23:59 trong múi giờ ĐỊA PHƯƠNG (máy tính)

Phiên giao dịch ngày/đêm với thời gian đã định nghĩa ở trên - dữ liệu hàng ngày được xây dựng từ các thanh intraday bắt đầu tại thời gian bắt đầu phiên giao dịch đêm (ngày trước) và kết thúc tại thời gian kết thúc phiên giao dịch ngày

- **Thời gian dịch chuyển** - là sự khác biệt về thời gian (tính bằng giờ) giữa múi giờ địa phương của bạn và múi giờ sở giao dịch
- **Cho phép kết hợp dữ liệu EOD -** Nó cho phép làm việc với cơ sở dữ liệu có sự kết hợp giữa dữ liệu intraday và dữ liệu EOD trong một tệp dữ liệu. Nếu tính năng này được bật, trong chế độ intraday, các thanh EOD sẽ bị loại bỏ ngay lập tức, và trong chế độ hàng ngày, các thanh EOD sẽ được hiển thị thay cho các thanh intraday nén thời gian, hoặc nếu không có thanh EOD cho ngày tương ứng, các thanh intraday sẽ được nén như thường lệ.

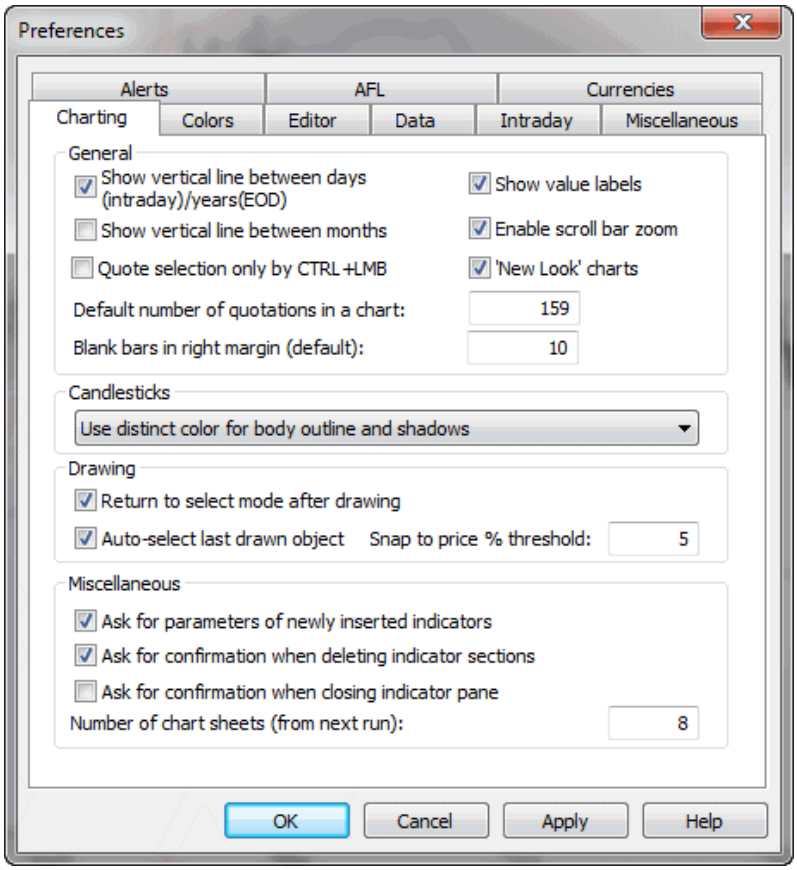
  Chế độ này hoạt động cùng với các phiên bản mới của các plugin cho phép dữ liệu hỗn hợp. Tính đến tháng 6 năm 2008, chế độ hỗn hợp hiện được hỗ trợ bởi plugin IQFeed và plugin eSignal (1.7.0 trở lên). Chế độ hỗn hợp cho phép kết hợp dữ liệu intraday với lịch sử hàng ngày rất dài trong một cơ sở dữ liệu.

Lưu ý rằng **Cài đặt Intraday** có sẵn từ hộp thoại **Cài đặt Cơ sở Dữ liệu** là **CÀI ĐẶT THEO CƠ SỞ DỮ LIỆU.** Tuy nhiên, cũng có tùy chọn để định nghĩa cài đặt intraday THEO NHÓM. Để sử dụng cài đặt intraday **THEO NHÓM**, bạn cần mở cửa sổ Ký hiệu -> Danh mục, chuyển sang tab Nhóm và đánh dấu vào ô "Nhóm sử dụng cài đặt intraday riêng" như hình dưới đây.



Sau đó, bạn có thể nhấp vào nút Cài đặt Intraday để hiển thị cài đặt theo nhóm. Lưu ý rằng mỗi nhóm trong danh sách danh mục có thể có các cài đặt riêng biệt, vì vậy bạn có thể dễ dàng thiết lập các nhóm chứa các công cụ giao dịch trong các giờ khác nhau. Bạn có thể di chuyển các ký hiệu giữa các nhóm bằng cách sử dụng hộp thoại Ký hiệu -> Tổ chức phân công.

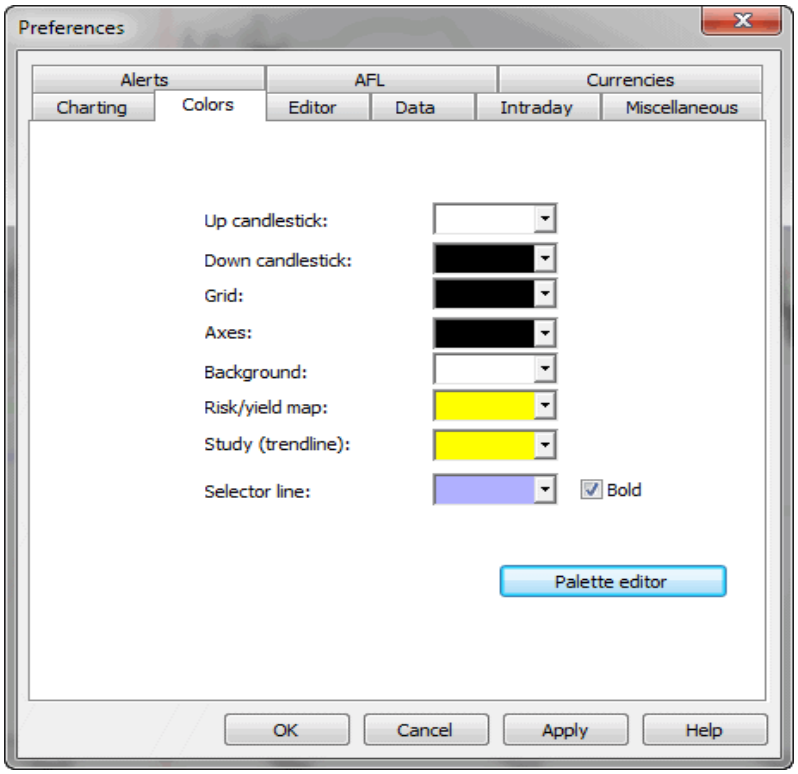
Cửa sổ Tùy chọn



Tab Biểu Đồ- cho phép bạn chỉnh sửa các tùy chọn biểu đồ:

- **Số lượng báo giá mặc định trong biểu đồ** - thiết lập số lượng thanh được hiển thị ban đầu trong biểu đồ (nói cách khác, định nghĩa khoảng thu phóng "bình thường").
- **Thanh trống ở lề bên phải** - xác định số lượng thanh trống được thêm vào lề bên phải (sau báo giá cuối cùng có sẵn). Lề trống này cho phép bạn dự đoán các nghiên cứu (ví dụ như đường xu hướng) vào tương lai.
- **Chọn báo giá chỉ bằng CTRL+LMB** - quyết định cách dòng lựa chọn dọc được gọi. Khi ô này không được đánh dấu, nhấp đơn vào biểu đồ sẽ hiển thị dòng lựa chọn; khi ô này được đánh dấu, bạn phải giữ phím CTRL trong khi nhấp để hiển thị dòng lựa chọn.
- **Hiển thị đường dọc giữa các ngày (intraday)/năm (EOD)**- quyết định xem đường dọc chấm có được hiển thị trên biểu đồ để đánh dấu ranh giới ngày (trong chế độ intraday) hoặc năm (trong chế độ EOD) hay không.
- **Hiển thị nhãn giá trị**- quyết định xem có hiển thị nhãn giá trị cho các chỉ báo/đường biểu đồ giá hay không. Xem hướng dẫn biểu đồ cơ bản để hiểu rõ hơn về nhãn giá trị.

- **Nến** - cài đặt này cung cấp điều khiển chi tiết về hình thức của các nến. Màu sắc khác biệt có thể được sử dụng để vẽ phần của nến hoặc toàn bộ nến có thể được vẽ cùng màu với phần bên trong của nó.
- **Vẽ**
    - **Quay lại chế độ chọn sau khi vẽ** - khi được đánh dấu, công cụ hiện tại sẽ bị vô hiệu hóa sau khi vẽ và chế độ chọn sẽ được kích hoạt. Khi không được đánh dấu, công cụ vẽ hiện tại vẫn hoạt động sau khi vẽ (cho phép bạn vẽ một nghiên cứu sau một nghiên cứu khác). Lưu ý rằng hiệu ứng tương tự cũng có thể đạt được ngay cả khi ô này được đánh dấu - chỉ cần giữ phím SHIFT trong khi vẽ và công cụ sẽ vẫn hoạt động.
    - **Tự động chọn đối tượng vừa vẽ** - tính năng hữu ích này tự động chọn đối tượng vừa vẽ gần đây. Điều này cho phép bạn nhấn ALT+ENTER để hiển thị hộp thuộc tính ngay lập tức mà không cần phải nhấp vào nghiên cứu, và cho phép bạn sao chép nghiên cứu thông qua CTRL+C cũng mà không cần nhấp thêm.
    - **Gắn vào ngưỡng % giá** - xác định khoảng cách mà "nam châm" giá hoạt động, nó sẽ gắn vào giá khi chuột gần hơn mức % ngưỡng so với giá cao/thấp/đóng.
- Khác
    - **Hỏi về tham số của các chỉ báo mới chèn** - khi được đánh dấu, AmiBroker sẽ tự động hiển thị cửa sổ Tham số mỗi khi bạn chèn một chỉ báo mới hoặc chồng một chỉ báo lên chỉ báo khác.
    - **Hỏi xác nhận khi xóa phần chỉ báo**- khi được đánh dấu, AmiBroker sẽ hỏi bạn xác nhận trước khi xóa bất kỳ phần chỉ báo nào đã chồng lên (áp dụng cho các chỉ báo được tạo thông qua kéo và thả). Lưu ý rằng việc xóa phần chỉ báo sẽ sửa đổi công thức cơ sở. Thêm thông tin về điều này có trong Hướng dẫn: Kéo và thả.
    - **Số lượng tối đa các bảng biểu đồ** - xác định số lượng bảng biểu đồ (tab) có sẵn. Thông tin thêm về bảng biểu đồ có trong phần Hướng dẫn tại đây. Lưu ý rằng cài đặt này sẽ có hiệu lực sau khi khởi động lại.



Tab Màu - cho phép bạn định nghĩa màu sắc cho các phần tử biểu đồ cụ thể.

Các điều khiển cung cấp tùy chọn chọn màu do người dùng xác định cho biểu đồ, lưới và nền.

Trình chỉnh sửa bảng màu - cho phép bạn sửa đổi các màu tùy chỉnh mà có thể được tham chiếu sau này thông qua các hằng số colorCustom0 đến colorCustom15.

Tab Biên Tập - điều khiển giao diện và tính năng của trình biên tập AFL.

- **Cài đặt phông chữ** - cho phép bạn định nghĩa kiểu phông chữ và kích thước cho trình biên tập AFL.
- **Sử dụng tô màu cú pháp**- khi được đánh dấu, trình biên tập sẽ tự động tô màu mã của bạn (các màu/kiểu khác nhau cho hàm, hằng số, số, v.v.).
- **Tự động thay đổi chữ hoa** - khi được đánh dấu, tên hàm và biến từ khóa sẽ tự động được viết hoa. Ví dụ, nếu bạn gõ `bARSSince`, trình biên tập sẽ thay đổi thành `BarsSince`.
- **Tự động hoàn thành** - khi được đánh dấu, bạn sẽ có thể sử dụng tính năng tự động hoàn thành (CTRL+SPACE sẽ tự động hoàn thành từ).
- **Thông tin tham số** - khi được đánh dấu, trình biên tập sẽ hiển thị tooltip thông tin tham số khi bạn gõ tên hàm và dấu ngoặc mở.

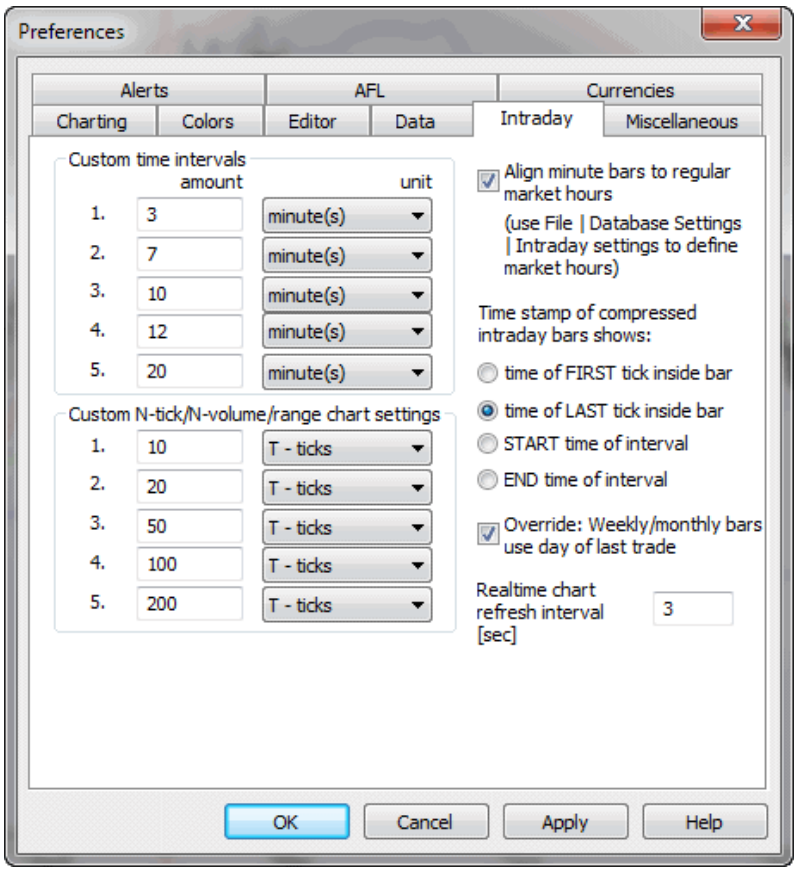
- **Đánh dấu dòng lỗi** - khi được đánh dấu, trình biên tập công thức sẽ đánh dấu dòng mã có chứa lỗi bằng nền màu vàng (chỉ trên Windows 2000 và XP).
- **Sao chép dưới dạng HTML** - khi được đánh dấu, trình biên tập AFL trong lệnh Chỉnh sửa -> Sao chép / Cắt sẽ đưa không chỉ văn bản thuần túy và định dạng RTF vào clipboard, mà còn cả định dạng HTML và DwHTML (HTML của Dreamweaver), cho phép dán mã có tô màu cú pháp vào Macromedia Dreamweaver và các ứng dụng nhận biết HTML khác. Lưu ý: hiếm khi (trên rất ít máy), việc bật tính năng này có thể gây ra vấn đề khi dán vào Outlook.
- Di chuyển các tệp đã chỉnh sửa từ Kéo và Thả đến Thư mục Tùy chỉnh - khi được đánh dấu, trình biên tập sẽ tự động di chuyển các công thức đã chỉnh sửa thủ công được tạo ra bằng cơ chế kéo và thả từ thư mục con ẩn 'Drag-drop' sang thư mục con 'Custom'.
- **Màu sắc và kiểu dáng** - cho phép bạn định nghĩa các màu sắc và kiểu dáng sẽ được sử dụng để đánh dấu các phần tử ngôn ngữ nhất định khi tính năng tô màu cú pháp được bật.

Tab Dữ Liệu - cho phép bạn định nghĩa các giá trị mặc định, toàn cục cho tất cả các cơ sở dữ liệu.

QUAN TRỌNG: một số cài đặt này có thể **bị ghi đè bởi các cài đặt THEO CƠ SỞ DỮ LIỆU** trong **File->Database cửa sổ Cài đặt**. Xem giải thích trong Hướng dẫn: Hiểu về khái niệm cơ sở dữ liệu.

- **Nguồn dữ liệu:** xác định nguồn dữ liệu mặc định (dành cho các cơ sở dữ liệu không chỉ định nguồn khác trong File->Database Settings)
- **Lưu trữ dữ liệu cục bộ**: cài đặt mặc định cho các cơ sở dữ liệu ngoại vi (cài đặt này bị ghi đè bởi File->Database Settings). Nếu "Đã kích hoạt" thì dữ liệu ngoại vi được lưu trữ tạm thời trong các tệp cục bộ. Nếu "Đã tắt" thì các tệp cục bộ không lưu trữ dữ liệu ngoại vi.
- **Bộ nhớ đệm trong bộ nhớ** (tối đa. biểu tượng) - xác định bao nhiêu dữ liệu biểu tượng nên được giữ trong RAM (để truy cập rất nhanh) - cài đặt này hoạt động cùng với cài đặt tiếp theo
- **Bộ nhớ đệm trong bộ nhớ** (tối đa. MegaBytes) - xác định bao nhiêu MB của RAM nên được sử dụng để tạo bộ nhớ đệm dữ liệu tạm thời (để truy cập rất nhanh)
- **Số lượng thanh để tải** - cài đặt mặc định cho cơ sở dữ liệu ngoại vi (cài đặt này bị ghi đè bởi File->Database Settings). Xác định bao nhiêu thanh nên được tải từ nguồn dữ liệu ngoại vi và giữ trong AmiBroker. Ví dụ: EOD 10 năm: 2600, intraday 1 phút 60 ngày: 30000 (khoảng)
- **Giới hạn số lượng báo giá được lưu** - nếu tùy chọn này ĐƯỢC BẬT, AmiBroker sẽ lưu cơ sở dữ liệu với số lượng báo giá giới hạn. Điều này ngăn cơ sở dữ liệu phát triển quá nhiều
- **Số lượng tối đa của báo giá được lưu -** đây là giới hạn. Ưu tiên 300 hoặc cao hơn cho cơ sở dữ liệu EOD, 3000 hoặc cao hơn cho intraday
- **Đường dẫn cơ sở dữ liệu mặc định -** điều này xác định đường dẫn đến cơ sở dữ liệu được tải khi khởi động. Nếu cơ sở dữ liệu như vậy không tồn tại, nó sẽ được tạo lại khi khởi động.

- **Số lượng thanh để tải -** cài đặt mặc định cho cơ sở dữ liệu ngoại vi (cài đặt này bị ghi đè bởi File->Database Settings). Xác định bao nhiêu thanh nên được tải từ nguồn dữ liệu ngoại vi và giữ trong AmiBroker. Ví dụ: EOD 10 năm: 2600, intraday 1 phút 60 ngày: 30000 (khoảng)
- **Giới hạn số lượng báo giá được lưu -** nếu tùy chọn này được bật, AmiBroker sẽ lưu cơ sở dữ liệu với số lượng báo giá hạn chế. Điều này giúp ngăn chặn cơ sở dữ liệu phát triển quá mức.
- **Số lượng tối đa báo giá được lưu** - đây chính là giới hạn. Khuyến nghị 300 hoặc cao hơn cho các cơ sở dữ liệu EOD, 3000 hoặc cao hơn cho các cơ sở dữ liệu intraday.
- **Đường dẫn cơ sở dữ liệu mặc định** - xác định đường dẫn đến cơ sở dữ liệu được tải khi khởi động. Nếu cơ sở dữ liệu như vậy không tồn tại, nó sẽ được tái tạo lại vào thời điểm khởi động.



Tab Intraday -cung cấp các cài đặt cho việc biểu đồ intraday.

- **Khoảng thời gian tùy chỉnh** - cho phép xác định khoảng thời gian N-phút hoặc N-giờ của riêng bạn (có sẵn sau từ menu View->Intraday)
- **Cài đặt biểu đồ N-tick tùy chỉn**h - cho phép xác định biểu đồ N-tick của riêng bạn (có sẵn sau từ menu View->Intraday)
- **Căn chỉnh thanh phút tùy chỉnh vào giờ giao dịch thông thường** - khi được chọn, AmiBroker sẽ cắt thanh phút tùy chỉnh trước thị trường để thanh

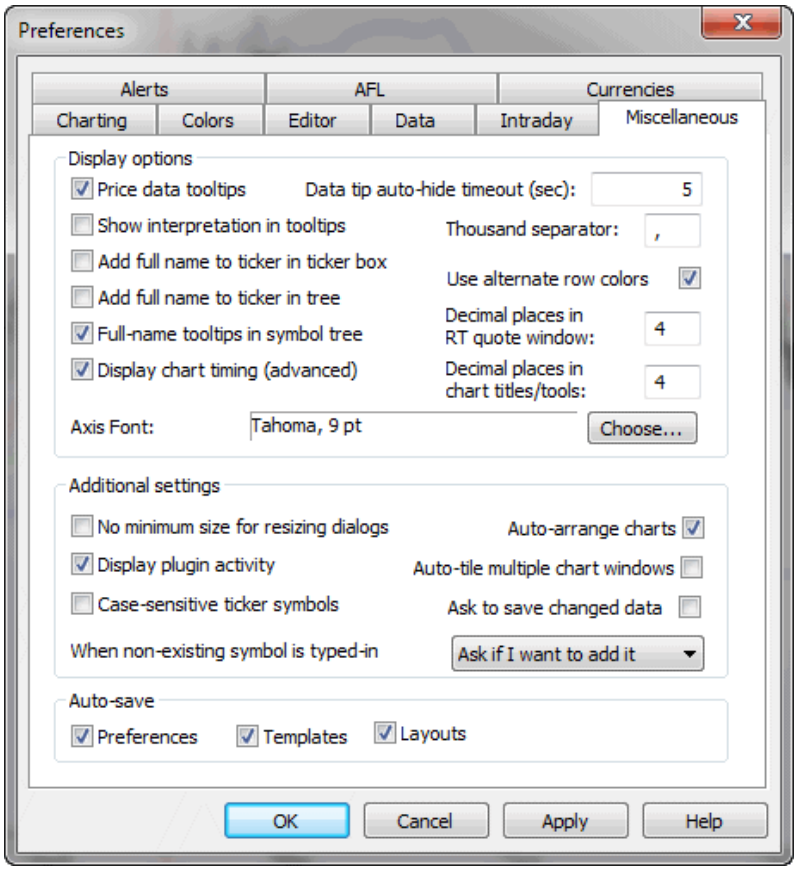
mới bắt đầu chính xác khi giờ giao dịch bắt đầu. Giờ giao dịch có thể được thiết lập cho mỗi cơ sở dữ liệu trong File->Database Settings->Intraday settings. Giả sử rằng chúng ta có thanh 45 phút. Nếu không có cài đặt này, chúng ta sẽ nhận được thanh bắt đầu lúc 9:00, 9:45, 10:30, 11:15, v.v. Khi được bật và giao dịch bắt đầu lúc 9:30, chúng ta có đảm bảo rằng thanh sẽ được căn chỉnh thành 9:30: 8:45, 9:30, 10:15, 11:00

- **Thanh nén thời gian hiển thị:**
  - **thời gian của bước nhảy ĐẦU TIÊN trong thanh** - khi được chọn, thanh nhận dấu thời gian của giao dịch đầu tiên trong dải thời gian đã cho (bar)
  - **thời gian của bước nhảy CUỐI TRONG thanh** - khi được chọn, thanh nhận dấu thời gian của giao dịch cuối cùng trong dải thời gian đã cho (bar)
  - **Thời gian BẮT ĐẦU của khoảng** - khi được chọn, thanh được gắn dấu thời gian với thời gian bắt đầu của dải thời gian (bar). Giả sử rằng thanh 30 phút bao gồm 9:00:00..9:29:59. Khi được chọn, AmiBroker sẽ hiển thị thời gian của thanh này là 9:00
  - **Thời gian KẾT THÚC của khoảng** - khi được chọn, thanh được gắn dấu thời gian với thời gian bắt đầu của dải thời gian (bar). Giả sử rằng thanh 30 phút bao gồm 9:00:00..9:29:59. Khi được chọn, AmiBroker sẽ hiển thị thời gian của thanh này là 9:29:59
- **Khoảng thời gian làm mới biểu đồ thời gian thực** - xác định khoảng thời gian giữa các lần làm mới biểu đồ tự động trong chế độ thời gian thực. Theo mặc định, biểu đồ được làm mới mỗi 3 giây nhưng trong thị trường rất biến động bạn có thể muốn đặt nó thành 1, vì vậy biểu đồ được làm mới mỗi giây trong thời gian thực

Mới trong 4.90: Để bật chức năng 'tự động cập nhật mỗi giây' cho biểu đồ trong Phiên bản Chuyên nghiệp, hãy đi tới Tools->Preferences, tab Intraday và nhập SỐ KHÔNG (0) vào trường "Intraday chart refresh interval". (lưu ý Phiên bản Tiêu chuẩn sẽ không cho phép bạn làm điều này).

Khi bạn nhập số không, AmiBroker sẽ làm mới tất cả các biểu đồ với mỗi giao dịch mới được gửi đi với điều kiện là các công thức bạn sử dụng được thực thi đủ nhanh. Nếu không, nó sẽ tự động điều chỉnh tốc độ làm mới để duy trì tốc độ làm mới tối đa có thể mà không tiêu thụ hơn 50% của CPU (trung bình). Vì vậy, ví dụ, nếu biểu đồ của bạn mất 0,2 giây để thực thi thì AmiBroker sẽ làm mới chúng trung bình 2,5 lần/giây.

Lưu ý: biểu đồ Performance mặc định của Windows hiển thị tổng việc tiêu thụ CPU cho tất cả các quá trình, để xem TẢI CPU THEO QUY TRÌNH sử dụng phần mềm miễn phí SysInternals tại http://www.sysinternals.com/Utilities/ProcessExplorer.html



- **Dữ liệu giá**

  nếu kiểm tra tooltips nhỏ sẽ xuất hiện khi bạn di chuột lên biểu đồ hiển thị ngày của thanh đã chọn, giá/ giá trị chỉ số

- **Hiển thị giải thích trong tooltip**

  nếu kiểm tra dữ liệu tooltips cũng sẽ bao gồm cả văn bản giải thích mà thông thường sẽ hiển thị trong cửa sổ giải thích.

- **Thời gian tự động ẩn tooltip dữ liệu**

  xác định bao nhiêu giây dữ liệu tooltip nên hiển thị trên màn hình nếu bạn không di chuyển chuột.

- T**hêm tên đầy đủ vào dấu ticker trong hộp ticker**

  khi được chọn hộp ticker sẽ hiển thị không chỉ biểu tượng mà còn cả tên đầy đủ của vấn đề được chọn

- **Thêm tên đầy đủ vào ticker trong cây**

khi được chọn, cây không gian làm việc sẽ hiển thị không chỉ biểu tượng mà còn cả tên đầy đủ của vấn đề.

- **Chú giải tên đầy đủ trong cây biểu tượng:**

  khi được chọn, tên đầy đủ của biểu tượng sẽ được hiển thị trong chú giải xuất hiện khi bạn di chuyển chuột qua biểu tượng trong cây biểu tượng.

- **Thời gian tự động ẩn Data-tip:**

  Xác định thời gian (tính bằng giây) Tooltip dữ liệu (hiển thị giá trị của các chỉ số) sẽ được hiển thị khi con trỏ chuột không di chuyển.

- **Dấu phân cách hàng nghìn**:

  Xác định dấu phân cách hàng nghìn cho số được hiển thị trên biểu đồ và tất cả các chế độ xem danh sách.

- **Vị trí thập phân trong cửa sổ báo giá thời gian thực:**

  Xác định số vị trí thập phân cần hiển thị trong cửa sổ báo giá thời gian thực.

- **Phông chữ trục**:

  Xác định kiểu và kích thước phông chữ để sử dụng cho trục biểu đồ và công cụ văn bản.

- **Không kích thước tối thiểu khi thay đổi kích thước hộp thoại:**

  Khi được chọn, cho phép điều chỉnh kích thước các hộp thoại dưới kích thước tối thiểu (điều này khiến một số điều khiển trở nên không thể nhìn thấy).

- **Hiển thị hoạt động plugin:**

  Khi được lựa chọn, AmiBroker hiển thị thông tin về việc truy cập plugin dữ liệu trên thanh trạng thái.

- **Kí hiệu ticker phân biệt chữ hoa chữ thường**:

  Khi được chọn, các ký hiệu ticker phân biệt chữ hoa chữ thường. Nói cách khác, INTC and Intc và iNTc được coi là KHÁC NHAU. Điều này có thể yêu cầu với một số biểu tượng của Canada chẳng hạn. Hãy cẩn trọng khi sử dụng. Nếu sàn giao dịch của bạn không sử dụng ký hiệu ticker phân biệt hoa thường, hãy chắc chắn rằng nó KHÔNG ĐƯỢC CHỌN.

- **Tự động sắp xếp biểu đồ:**

Nếu tùy chọn này được bật, các cửa sổ biểu đồ sẽ được cân đối và sắp xếp để phù hợp với màn hình sau mỗi lần mở / đóng cửa sổ biểu đồ.

- **Tự động chia cửa sổ biểu đồ nhiều hơn**:

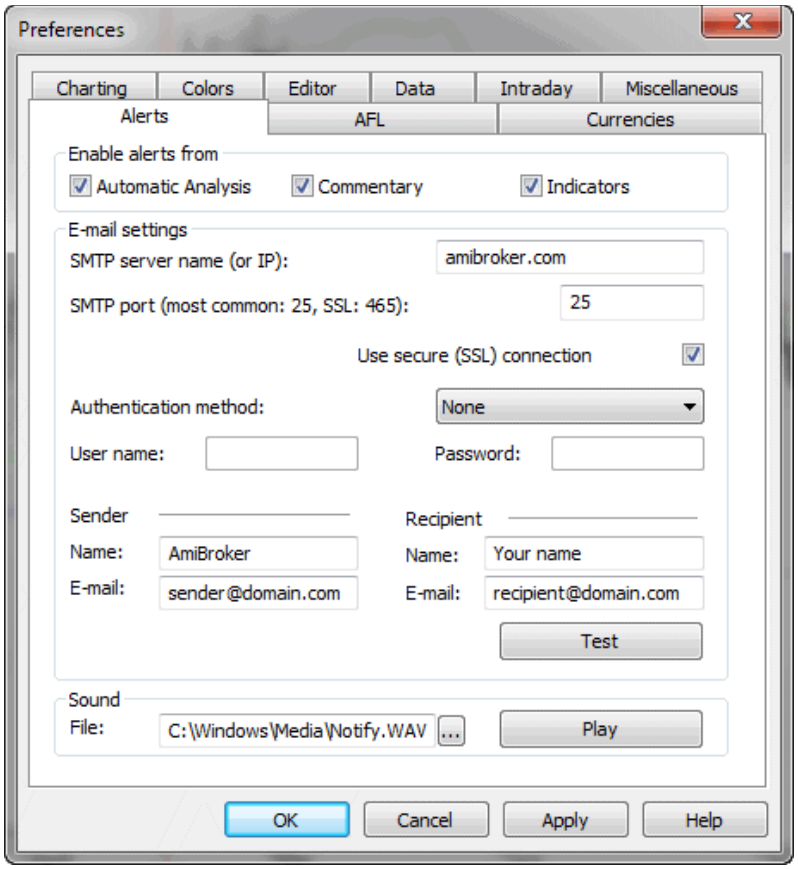
  Khi được chọn, nhiều cửa sổ biểu đồ sẽ luôn được chia dọc mỗi khi kích thước của cửa sổ ứng dụng chính thay đổi.

- **Hỏi để lưu dữ liệu thay đổi:**

  Khi được chọn, AmiBroker sẽ hỏi nếu bạn muốn lưu dữ liệu đã sửa đổi khi thoát

- **Lưu khi thoát: Tùy chọn, Mẫu, Bố cục**

  Điều này điều khiển các cài đặt nào sẽ được lưu tự động khi thoát.



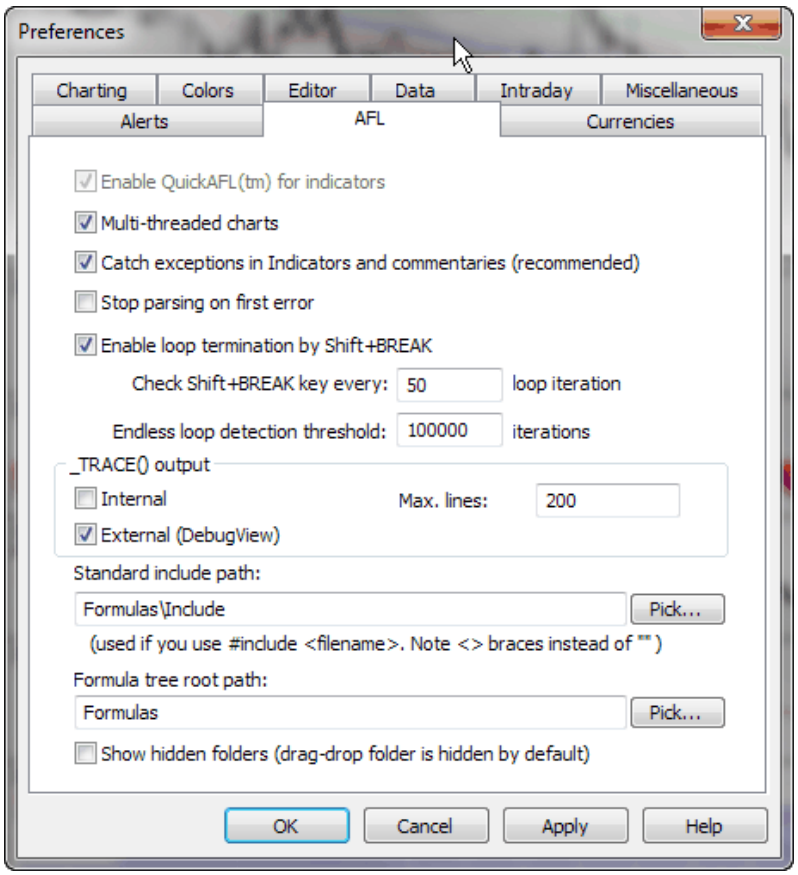
Tab Cảnh báo - Tab này cho phép định nghĩa cài đặt tài khoản email, kiểm tra âm thanh đầu ra và xác định các phần của AmiBroker có thể tạo ra cảnh báo thông qua chức năng AlertIF.

Trang cài đặt email hiện cho phép chọn giữa các phương thức xác thực phổ biến nhất như: AUTH LOGIN (phổ biến nhất), POP3-trước-SMTP (phổ biến), CRAM-

MD5, LOGIN PLAIN. Phiên bản 5.30 cũng cho phép sử dụng SSL (kết nối an toàn) mà GMail sử dụng, ví dụ. Để biết thêm thông tin về cách thiết lập với GMail, xem Hướng dẫn: Cảnh báo dựa trên công thức.

Việc bật cảnh báo từ các hộp kiểm cho phép bạn chọn lọc bật/tắt các cảnh báo được tạo ra bởi Phân tích tự động, Bình luận/Diễn giải và các chỉ báo tùy chỉnh.

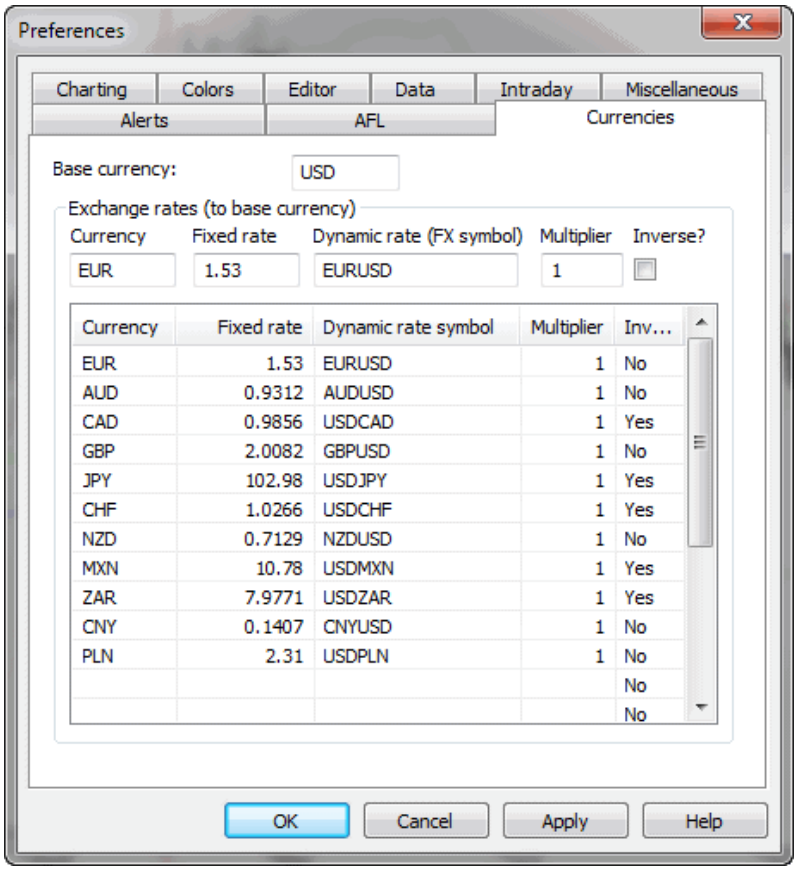
Tab Bàn phím - Tab bàn phím đã được chuyển đến **Tools -> Customize.**



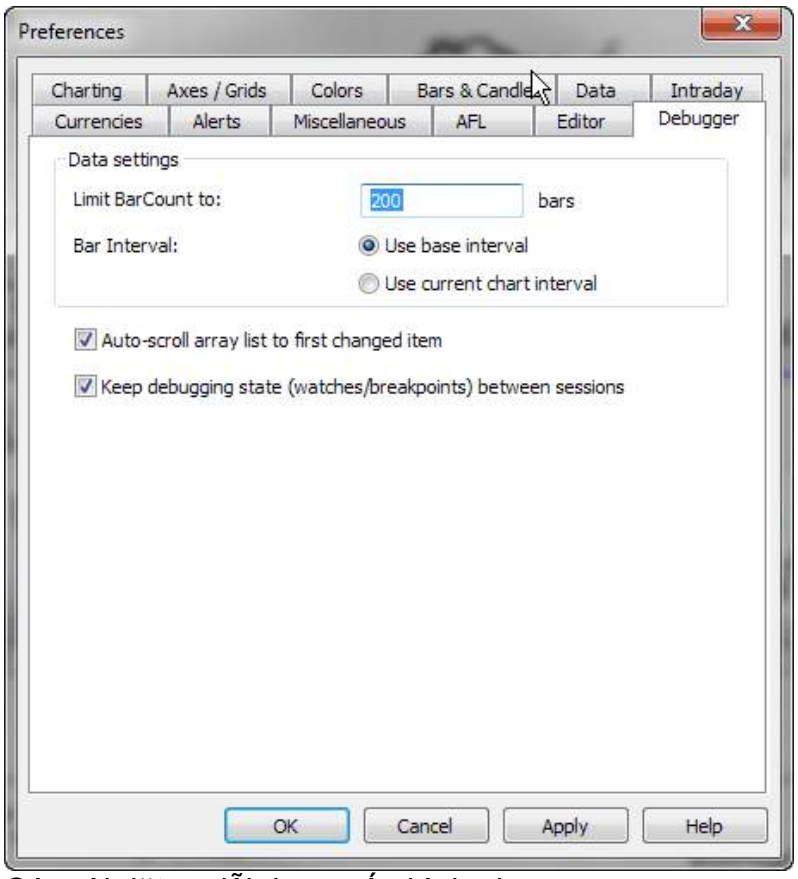
Tab AFL

- **Biểu đồ đa luồng** - Cho phép thực thi đa luồng của AFL trong biểu đồ/chỉ báo. Đa luồng giúp tối đa hóa tốc độ và sử dụng của các máy tính hiện đại với nhiều lõi/multi-CPU. Ví dụ, trên CPU Intel i7 8 lõi, biểu đồ của bạn sẽ chạy nhanh gấp 8 lần so với phiên bản 5.30. Trong phiên bản 5.40, động cơ AFL đã được viết lại hoàn toàn từ đầu để cho phép nhiều phiên bản của động cơ chạy đồng thời. Điều này không chỉ cho phép đa luồng mà còn tăng cường khả năng phản hồi của toàn bộ ứng dụng, vì ngay cả công thức người dùng viết không tốt được sử dụng trong biểu đồ cũng không thể khóa hoặc làm chậm phần còn lại của chương trình. Đa luồng được bật theo mặc định. Nó có thể được tắt bằng cách bỏ chọn hộp này, nhưng không được khuyến khích. Đa luồng nên được bật nếu bạn muốn AmiBroker hoạt động với tốc độ tối đa.

- **Bắt ngoại lệ hệ thống trong các chỉ báo và bình luận** - Khi được chọn, tất cả các ngoại lệ (lỗi thời gian chạy) sẽ được mã vẽ chỉ báo bắt, vì vậy không có cửa sổ Khôi phục Lỗi nào xuất hiện. Thay vào đó, thông tin ngoại lệ sẽ được hiển thị bên trong khung biểu đồ. Được khuyến nghị bật tính năng này, đặc biệt khi bạn sử dụng dữ liệu thời gian thực.
- **Dừng phân tích khi gặp lỗi đầu tiên** - Khi được chọn, bộ phân tích sẽ dừng phân tích mã khi gặp lỗi đầu tiên, vì vậy chỉ một lỗi (lỗi đầu tiên) sẽ được hiển thị trong danh sách lỗi của trình chỉnh sửa công thức. Nếu không được chọn, bộ phân tích sẽ liệt kê tất cả các lỗi tìm thấy. Nên tắt tính năng này.
- **Cho phép dừng vòng lặp bằng Shift-BREAK** - Khi được chọn, AmiBroker sẽ cho phép dừng bất kỳ vòng for(), while() và do-while() nào bằng cách nhấn và giữ phím SHIFT và phím BREAK (PAUSE) trên bàn phím của bạn.
- **Kiểm tra phím Shift+BREAK mỗi** - Định nghĩa tần suất kiểm tra trạng thái bàn phím khi vòng lặp đang được thực thi. Lưu ý rằng việc chỉ định các giá trị nhỏ sẽ làm chậm việc thực thi vòng lặp.
- **Ngưỡng phát hiện vòng lặp vô tận** - Định nghĩa số lượng vòng lặp sau đó AmiBroker sẽ kết thúc vòng lặp với thông báo lỗi "Có thể phát hiện vòng lặp vô tận". Điều này hữu ích trong những tình huống khi mã có vòng lặp vô tận (do sai sót của tác giả công thức) vì nó sẽ không cho phép AmiBroker gặp lỗi do vòng lặp vô hạn
- **Đường dẫn include chuẩn** - đường dẫn mặc định để sử dụng khi lệnh #include sử dụng < > braces thay vì - Đường dẫn root của Formula tree - đường dẫn root của file/danh mục Formula được hiển thị trong tab Charts của cửa sổ Workspace
- **Hiển thị thư mục ẩn** - xác định xem cây công thức có nên hiển thị các thư mục con với thuộc tính "ẩn" không (thư mục kéo-thả được tạo thành "ẩn" bởi chương trình thiết lập)


Preferences
Charting
Colors
Editor
Data
Intraday
Miscellaneous
Alerts
AFL
Currencies
Base currency:
USD
Exchange rates (to base currency)
Currency
Fixed rate
Dynamic rate (FX symbol)
Multiplier
Inverse?
EUR
1.53
EURUSD
1
Currency
Fixed rate
Dynamic rate symbol
Multiplier
Inv...
EUR 1.53 EURUSD 1 No
AUD 0.9312 AUDUSD 1 No
CAD 0.9856 USDCAD 1 Yes
GBP 2.0082 GBPUSD 1 No
JPY 102.98 USDJPY 1 Yes
CHF 1.0266 USDCHF 1 Yes
NZD 0.7129 NZDUSD 1 No
MXN 10.78 USDMXN 1 Yes
ZAR 7.9771 USDZAR 1 Yes
CNY 0.1407 CNYUSD 1 No
PLN 2.31 USDPLN 1 No
No
No
OK
Cancel
Apply
Help

Tab Gỡ lỗi



Các cài đặt gỡ lỗi được cấu hình như sau:

- **Hạn chế BarCount đến** - xác định số lượng tối đa của các thanh trong mảng (BarCount) được sử dụng trong quá trình gỡ lỗi (mặc định là 200 thanh)
- **Khoảng thời gian Bar** - quyết định nếu trình gỡ lỗi sử dụng khoảng thời gian cơ sở hoặc khoảng thời gian biểu đồ hiện tại. Mặc định là "Sử dụng khoảng thời gian cơ sở" vì có thể không có biểu đồ nào mở cả!
- **Tự động cuộn đến mục thay đổi đầu tiên** - khi chức năng này ON, danh sách "Mảng" trong cửa sổ Watch sẽ tự động cuộn đến mục mảng đầu tiên đã thay đổi (vì vậy bạn không cần xác định các phần tử đã thay đổi theo cách thủ công)
- **Duy trì trạng thái gỡ lỗi** - khi chức năng này ON, trình chỉnh sửa AFL lưu trạng thái gỡ lỗi (điểm dừng và giám sát) và dấu trang trong tệp .dbg cùng với công thức khi đóng trình chỉnh sửa và khôi phục trạng thái khi mở lại công thức. Lưu ý rằng tệp .dbg sẽ tự động được AmiBroker xóa khi không có điểm dừng, không có dấu trang và không có giám sát.

Tab Tiền tệ

Trang này cho phép xác định tiền tệ cơ sở và tỷ giá hối đoái (cố định hoặc động) cho các loại tiền tệ khác nhau. Điều này cho phép có kết quả backtest chính xác khi kiểm tra các chứng khoán niêm yết bằng tiền tệ khác so với tiền tệ cơ sở của danh mục

đầu tư của bạn. Để biết thêm chi tiết, vui lòng xem Hướng dẫn: Pyramiding và hỗ trợ nhiều tiền tệ trong bộ kiểm tra phản hồi.

AB biết như thế nào tôi muốn báo giá cố định hay động?

Có những yêu cầu sau để sử dụng điều chỉnh tiền tệ:

a) Biểu tượng->Thông tin, trường "Tiền tệ" hiển thị tiền tệ khác so với tiền tệ BASE

b) Đơn vị tiền tệ phù hợp (được định nghĩa bằng Ký hiệu) có mục tương ứng trên trang Preferences (Tùy chọn) -> Currencies (Tiền tệ)

c) tỷ giá động "FX SYMBOL" được định nghĩa trong phần tùy chọn HIỆN DIỆN trong cơ sở dữ liệu của bạn và CÓ BÁO GIÁ cho mỗi ngày trong phạm vi phân tích.

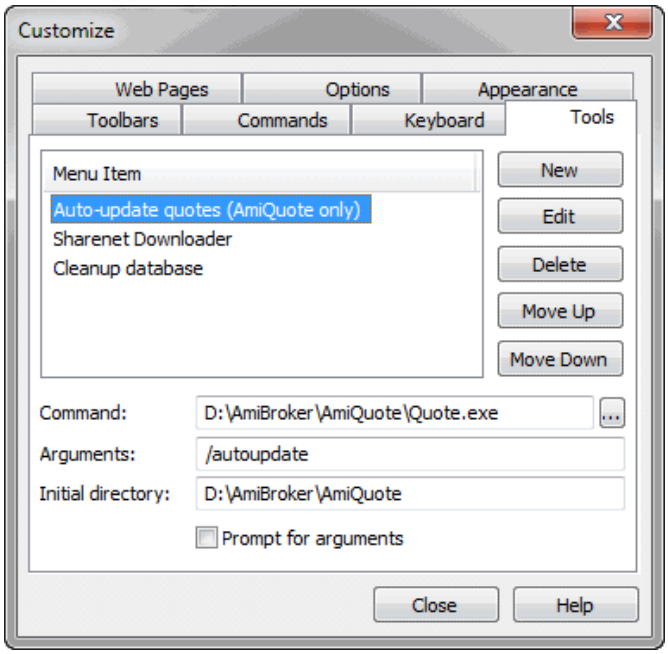
Hộp kiểm "INVERSE" trong phần tùy chọn có tác dụng gì?

Hãy lấy ví dụ về EURUSD.

Khi "USD" là đơn vị tiền tệ CƠ SỞ của bạn thì tỷ giá hối đoái EUR sẽ là "straight" EURUSD fx (tức là 1.3). Nhưng khi "EUR" là đơn vị tiền tệ CƠ SỞ của bạn thì tỷ giá hối đoái USD sẽ là NĂM NGƯỢC của EURUSD (tức là 1 / 1.3). Trường hợp ngược lại cũng đúng với những tỷ giá FX như USDJPY (đã là "inverse").

Tùy chỉnh Cửa sổ Công cụ

Hộp thoại này cho phép bạn tùy chỉnh Giao diện Người dùng. Nó có thể được gọi từ menu Tools->Customize. Trong tab "Tools", bạn xác định các mục menu công cụ tùy chỉnh:



Bạn có thể khởi chạy các tệp tin thực thi (.exe), tệp script (.js, .vbs), trang web (.html) và bất kỳ loại tệp đã đăng ký nào khác từ menu công cụ. Để thêm một công cụ mới, bạn nên mở hộp thoại này và nhấp vào nút "New". Sau đó nhập tên công cụ, lệnh (thủ công hoặc sử dụng hộp thoại tệp tin) các tham số tùy chọn và thư mục ban đầu. Nếu bạn đánh dấu vào ô "Prompt for arguments", AmiBroker sẽ hỏi về các đối số của chương trình mỗi lần

Phiên bản 5.60 mang đến lệnh mới **#import** cho phép import các tệp tin ASCII từ đĩa cục bộ hoặc ngay cả từ nguồn (web) từ xa.

Trong **Tools->Customize**, trang "Tools", bạn hiện có thể xác định công cụ tùy chỉnh sử dụng lệnh #import mới Lệnh: #import

Tham số: URL để tải dữ liệu từ

Initial dir: đường dẫn đến tệp định nghĩa format.

Chức năng này được sử dụng bởi công cụ "Cập nhật danh sách và danh mục ký hiệu Mỹ".

Các tab khác cung cấp các tính năng tùy chỉnh giao diện người dùng được mô tả trong phần hướng dẫn Tùy chỉnh UI.

Tab Bàn phím

Tab Bàn phím cho phép bạn xác định các phím tắt bàn phím của riêng mình.

Để Gán Một Phím Tắt

Trên menu Tools, nhấp vào Customize, sau đó nhấp vào tab Keyboard.

Trong danh sách Categories, chọn menu chứa lệnh mà bạn muốn gán phím tắt.

Trong danh sách Commands, chọn lệnh mà bạn muốn gán phím tắt.

Đặt con trỏ vào hộp Press New Shortcut Key, nhấn phím tắt hoặc tổ hợp phím bạn muốn, sau đó nhấp vào Assign.

Nếu bạn nhấn một phím hoặc tổ hợp phím không hợp lệ, sẽ không có phím nào được hiển thị. Bạn không thể gán các tổ hợp phím với ESC, F1, hoặc các tổ hợp như CTRL+ALT+DEL đang được hệ điều hành sử dụng.

Nếu bạn nhấn một phím hoặc tổ hợp phím hiện tại đã được gán cho một lệnh khác và nhấn "Assign" thì thông báo lỗi sẽ xuất hiện cho bạn lựa chọn giữa việc hủy hay gán lại phím tắt cho lệnh mới.

Để Xóa Một Phím Tắt

Trên menu Tools, nhấp vào Customize, sau đó nhấp vào tab Keyboard.

Trên danh sách Categories và Commands, chọn vị trí cho phím tắt bạn muốn xóa.

Trong danh sách Current Keys, chọn phím tắt bạn muốn xóa và nhấp vào Remove.

Để Đặt Lại Tất Cả Các Phím Tắt Về Giá Trị Mặc Định

Trên menu Tools, nhấp vào Customize, sau đó nhấp vào tab Keyboard.

Nhấp vào Reset All.

Cửa sổ Cây Biểu tượng

Trong cửa sổ này, chúng ta có danh sách các biểu tượng và danh mục có sẵn. Việc chọn một trong các biểu tượng sẽ làm mới tất cả các biểu đồ đã mở và cập nhật thông tin trong các cửa sổ thông tin. Sự lựa chọn này là toàn cầu cho chương trình, nghĩa là tất cả các chức năng biểu tượng sẽ tham chiếu đến biểu tượng được chọn trong cửa sổ này.



Cửa sổ biểu tượng được chia thành ba phần:

a) Hộp tìm kiếm

b) Cây danh mục

c) Danh sách biểu tượng

Hộp tìm kiếm: Cho phép bạn thực hiện tìm kiếm toàn văn (bao gồm cả ký tự đại diện) trong biểu tượng và tên đầy đủ trong danh mục đã chọn. Ví dụ, nếu bạn chọn ngành "Công nghệ" và gõ A* (chữ 'A' và ký tự đại diện *), danh sách biểu tượng sẽ hiển thị tất cả các biểu tượng thuộc ngành Công nghệ có biểu tượng hoặc tên đầy đủ bắt đầu bằng chữ 'A'. Một ví dụ khác là gõ *-A0-FX - điều này sẽ trả về tất cả các biểu tượng forex trong cơ sở dữ liệu eSignal (những biểu tượng kết thúc bằng chuỗi -A0-FX).

Cây danh mục: Hiển thị các loại danh mục khác nhau.

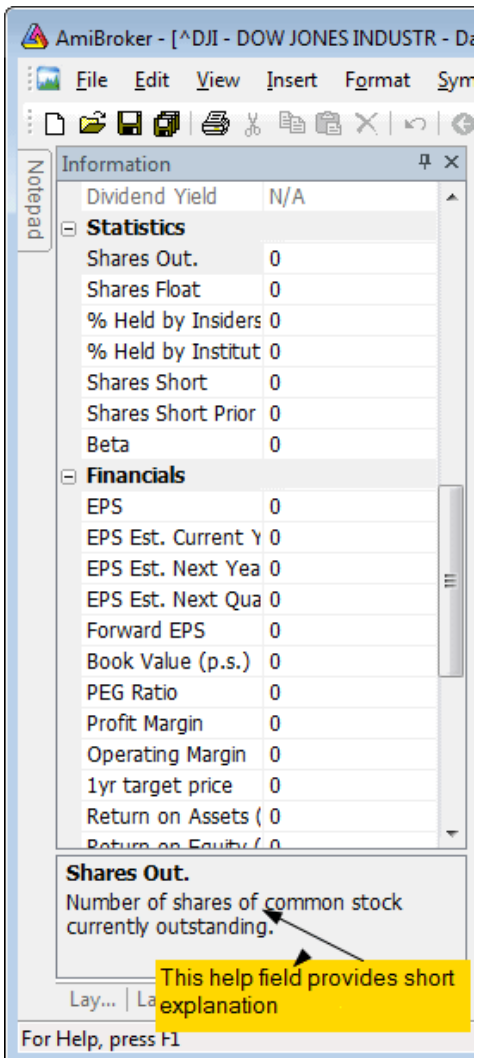
Danh sách biểu tượng: Phần dưới cùng hiển thị danh sách các biểu tượng thuộc danh mục đã chọn. Danh sách biểu tượng có thể được sắp xếp theo biểu tượng hoặc theo tên đầy đủ. Để sắp xếp, chỉ cần nhấp vào tiêu đề của danh sách. Một khi bạn chọn thứ tự sắp xếp mong muốn, nó sẽ được giữ cho tất cả các lựa chọn danh mục và tìm kiếm tiếp theo. Bạn cũng có thể thay đổi thứ tự của các cột, ví dụ, cột tên đầy đủ có thể xuất hiện đầu tiên. Để sắp xếp lại cột, nhấp vào tiêu đề cột, giữ nút chuột và kéo cột đến vị trí mong muốn, sau đó thả nút chuột.

Một biểu tượng có thể thuộc NHIỀU danh mục cùng một lúc. Ví dụ, AAPL (Apple Inc.) sẽ thuộc về:

- Nhóm danh mục Cổ phiếu
- Danh mục thị trường Nasdaq
- Danh mục ngành Thông tin
- Danh mục ngành sản xuất máy tính

Và có thể cũng thuộc về nhiều danh sách theo dõi và danh sách yêu thích cùng một lúc. Do đó, một biểu tượng sẽ xuất hiện trong nhiều nhánh của cây biểu tượng trong không gian làm việc. Nếu bạn xóa BIỂU TƯỢNG, nó sẽ biến mất khỏi TẤT CẢ các danh mục vì bạn đã xóa chính biểu tượng, không phải sự phân loại của nó.

Cửa sổ Thông tin



Cửa sổ này cho phép bạn hiển thị và chỉnh sửa các tùy chọn của biểu tượng.

- **Biểu tượng**
  Tên ngắn được sử dụng trong cửa sổ Chọn và với các chức năng nhập báo giá. Nếu bạn sử dụng chúng, hãy kiểm tra xem mã được cung cấp trong trường này có giống với mã sử dụng trong nguồn dữ liệu báo giá của bạn hay không.

- **Biệt danh**
  Tên ký hiệu thay thế. Điều này sẽ hữu ích nếu bạn lấy báo giá thời gian thực và thông tin lịch sử từ hai nguồn dữ liệu khác nhau, sử dụng các tên ký hiệu khác nhau.

- **Tên đầy đủ**
  Phiên bản chính thức của tên công ty.

- **Mã**
  Số mã biểu tượng.

- **Web ID**

ID web của biểu tượng - có thể được sử dụng khi bạn định nghĩa chế độ xem Hồ sơ.

- **Địa chỉ**
  Địa chỉ công ty.

- **Số lượng phát hành**
  Tổng số cổ phiếu.

- **Nominal value**
- **Book value**
- **Tiền tệ**

- **Thị trường:**

Chỉ rõ thị trường mà biểu tượng thuộc về.

- **Ngành:**

Chỉ rõ ngành mà biểu tượng thuộc về.

- **Nhóm:**

Chỉ rõ nhóm mà biểu tượng thuộc về.

- **Kích thước lô tròn**

Các công cụ khác nhau được giao dịch với nhiều "đơn vị giao dịch" hoặc "khối". Ví dụ, bạn có thể mua số lượng phần tây của quỹ tương hỗ, nhưng không thể mua số lượng phần tây của cổ phiếu. Đôi khi bạn phải mua theo lô 10 hoặc 100. AmiBroker giờ đây cho phép bạn chỉ định kích thước khối trên mức toàn cầu và theo biểu tượng.

Bạn có thể xác định kích thước lô tròn cho từng biểu tượng trong trang Thông tin của Biểu tượng. Giá trị bằng không có nghĩa là biểu tượng này không có kích thước lô tròn đặc biệt và sẽ sử dụng "Kích thước lô tròn mặc định" (cài đặt toàn cục) từ trang cài đặt Phân tích Tự động. Nếu kích thước mặc định cũng được đặt là không, điều này có nghĩa là các số lượng cổ phiếu/hợp đồng phân số được phép.

- **Cấu hình Tick size**

Cài đặt này kiểm soát sự di chuyển giá tối thiểu của biểu tượng đã cho. Bạn có thể xác định kích thước điểm ở mức toàn cầu và theo từng biểu tượng. Giống như kích thước lô tròn, bạn có thể xác định kích thước điểm cho từng biểu tượng trong trang Thông tin của Biểu tượng. Giá trị bằng không yêu cầu AmiBroker sử dụng "kích thước điểm mặc định" được định nghĩa trong trang Cài đặt của cửa sổ Phân tích Tự động. Nếu kích thước điểm mặc định cũng được đặt là không, điều này có nghĩa là không có sự di chuyển giá tối thiểu.

Lưu ý rằng cài đặt kích thước điểm chỉ ảnh hưởng đến các giao dịch thoát bởi các điểm dừng tích hợp và/hoặc ApplyStop(). Bộ kiểm tra ngược giả định rằng dữ liệu giá tuân theo yêu cầu về kích thước điểm và không thay đổi các mảng giá do người dùng cung cấp.

Do đó, việc chỉ định kích thước điểm chỉ có ý nghĩa nếu bạn đang sử dụng các điểm dừng tích hợp để các điểm thoát được tạo ra ở các mức giá "cho phép" thay vì các mức tính toán.Ví dụ, tại Nhật Bản, bạn không thể có các phần thập phân của yen, vì vậy bạn nên định nghĩa kích thước điểm toàn cầu là 1, để các điểm dừng tích hợp thoát giao dịch ở các mức số nguyên.

- **Tiền ký quỹ** - Được giải thích trong Hệ thống Kiểm tra Ngược cho hợp đồng tương lai.
- **Giá trị điểm** - Được giải thích trong Hệ thống Kiểm tra Ngược cho hợp đồng tương lai.

- **Báo giá liên tục**

  Bật tính năng giao dịch liên tục cho biểu tượng này (cho phép biểu đồ nến và điều khiển nhập thủ công mở/cao/thấp/thể tích), nếu không, biểu tượng sẽ được giao dịch với giá cố định.

- **Chỉ số**

Chỉ rõ liệu biểu tượng có thuộc danh mục Chỉ số hay không.

- **Yêu thích**

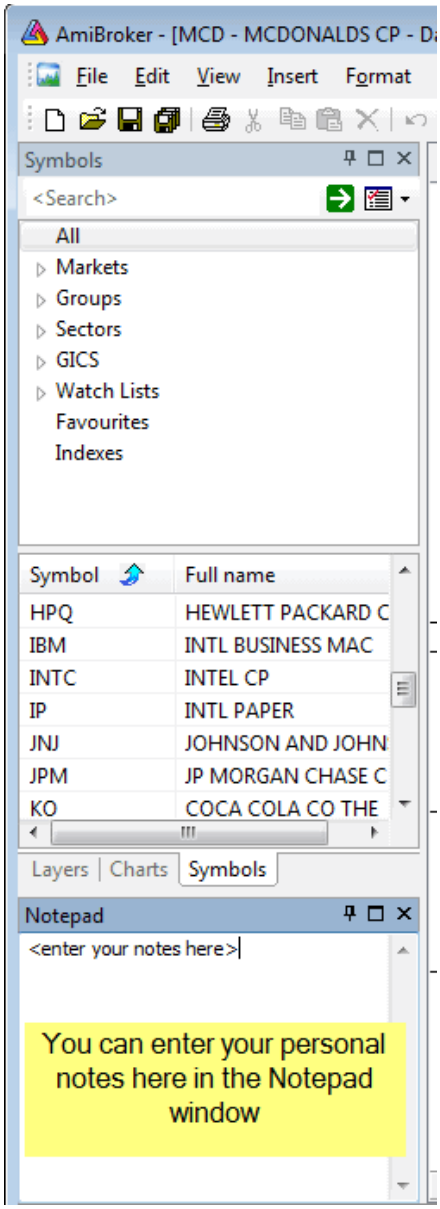
  Chỉ rõ liệu biểu tượng có thuộc danh mục Yêu thích hay không.

- **Chỉ sử dụng cơ sở dữ liệu cục bộ cho biểu tượng này**

  Chỉ ra rằng biểu tượng không được cập nhật qua plugin trong cơ sở dữ liệu thời gian thực. Trường này được chọn theo mặc định nếu biểu tượng được thêm vào cơ sở dữ liệu thời gian thực do kết quả của việc nhập từ tệp ASCII (cũng như tải xuống AmiQuote). Cài đặt này cho phép bạn giữ các biểu tượng bổ sung trong cơ sở dữ liệu và ngăn chặn plugin ghi đè lên dữ liệu đã nhập.

Để hiểu rõ hơn về các trường dữ liệu cơ bản, vui lòng đọc phần "Hướng dẫn: Sử dụng dữ liệu cơ bản" trong hướng dẫn này.

Cửa sổ Notepad



Cửa sổ Notepad (bạn có thể hiển thị/ẩn bằng cách sử dụng menu **Window->Notepad**) cho phép lưu trữ các ghi chú không cố định về bảo mật cụ thể. Chỉ cần gõ bất kỳ văn bản nào và nó sẽ tự động được lưu / đọc lại khi bạn duyệt qua các ký hiệu. Ghi chú là toàn cầu và được lưu trong thư mục con "Notes" dưới dạng các tệp văn bản thông thường.

Ghi chú cũng có thể được đọc và ghi bằng cách sử dụng các hàm NoteGet và NoteSet trong ngôn ngữ AFL.

NoteGet("Symbol");

- lấy ghi chú liên kết với "symbol". Nếu ký hiệu là "" (chuỗi trống) thì ký hiệu hiện tại được sử dụng.

NoteSet("Symbol", "Text...");

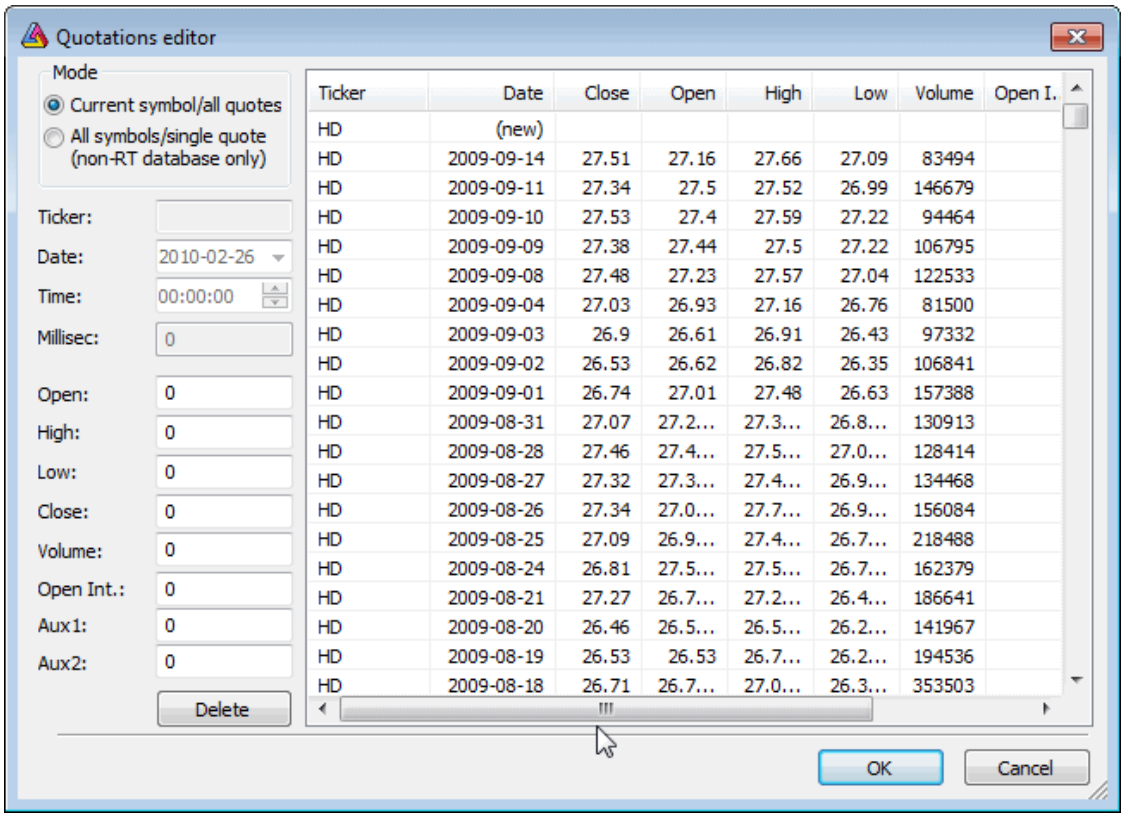
- đặt văn bản của ghi chú liên kết với "symbol".

Nếu ký hiệu là "" (chuỗi trống) thì ký hiệu hiện tại được sử dụng. Nếu bạn ghi đè ghi chú từ mức AFL mà đồng thời được mở trong trình soạn thảo Notepad, trình soạn thảo sẽ hỏi bạn (khi bạn chuyển tiêu điểm vào nó) nếu nó nên tải lại văn bản mới hay cho phép bạn lưu văn bản mà bạn đã nhập thủ công.

Ví dụ:

NoteSet("AMD", "Ngày 15 tháng 6 năm 2004: AMD sẽ giao vi xử lý đa nhân đầu tiên vào năm sau");

Cửa sổ Chỉnh sửa Báo giá



Ticker	Date	Close	Open	High	Low	Volume	Open I..
HD	(new)						
HD	2009-09-14	27.51	27.16	27.66	27.09	83494	
HD	2009-09-11	27.34	27.5	27.52	26.99	146679	
HD	2009-09-10	27.53	27.4	27.59	27.22	94464	
HD	2009-09-09	27.38	27.44	27.5	27.22	106795	
HD	2009-09-08	27.48	27.23	27.57	27.04	122533	
HD	2009-09-04	27.03	26.93	27.16	26.76	81500	
HD	2009-09-03	26.9	26.61	26.91	26.43	97332	
HD	2009-09-02	26.53	26.62	26.82	26.35	106841	
HD	2009-09-01	26.74	27.01	27.48	26.63	157388	
HD	2009-08-31	27.07	27.2...	27.3...	26.8...	130913	
HD	2009-08-28	27.46	27.4...	27.5...	27.0...	128414	
HD	2009-08-27	27.32	27.3...	27.4...	26.9...	134468	
HD	2009-08-26	27.34	27.0...	27.7...	26.9...	156084	
HD	2009-08-25	27.09	26.9...	27.4...	26.7...	218488	
HD	2009-08-24	26.81	27.5...	27.5...	26.7...	162379	
HD	2009-08-21	27.27	26.7...	27.2...	26.4...	186641	
HD	2009-08-20	26.46	26.5...	26.5...	26.2...	141967	
HD	2009-08-19	26.53	26.53	26.7...	26.2...	194536	
HD	2009-08-18	26.71	26.7...	27.0...	26.3...	353503	

Cửa sổ Chỉnh sửa Báo giá cho phép bạn chỉnh sửa, xóa và thêm các báo giá.

Để thêm báo giá mới

- Chọn mục (new).

- Nhập ngày/giờ.

- Nhập dữ liệu giá.

- Nhấp vào mục khác trong danh sách thay vì (new).

Để chỉnh sửa báo giá hiện có

- Chọn báo giá từ danh sách.

- Chỉnh sửa dữ liệu giá.

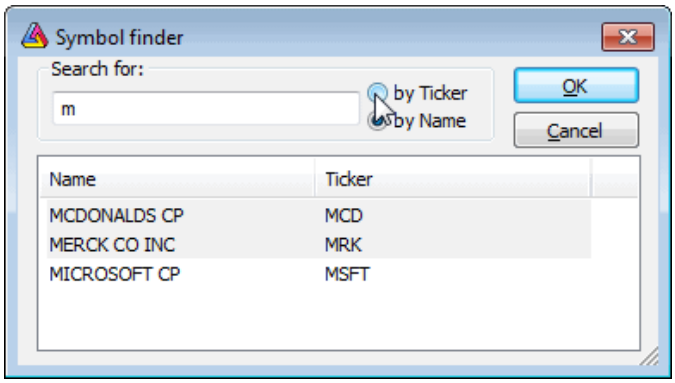
- Nhấp vào mục khác trong danh sách thay vì mục hiện tại.

Để xóa báo giá hiện có

- Đánh dấu một hoặc nhiều báo giá (có thể chọn nhiều bằng cách giữ phím SHIFT hoặc CTRL).

- Nhấp vào nút "Xóa".

Cửa sổ Tìm kiếm Biểu tượng (F3)



Cửa sổ Tìm kiếm Cổ phiếu cho phép bạn nhanh chóng tìm kiếm biểu tượng trong cơ sở dữ liệu bằng cách gõ các chữ cái đầu tiên của tên đầy đủ hoặc mã ký hiệu. Tính năng này rất hữu ích khi bạn không biết mã ký hiệu của biểu tượng. Cửa sổ Tìm kiếm biểu tượng có thể truy cập qua menu Edit->Find symbol, Symbol->Find hoặc bằng cách nhấn phím F3.

Để tìm một ký hiệu, chỉ cần nhập một hoặc nhiều chữ cái vào hộp **Tìm kiếm.** Chọn theo **Tên** nếu bạn muốn thực hiện tìm kiếm tên đầy đủ hoặc chọn theo **Mã chứng khoán** nếu bạn muốn tra cứu mã chứng khoán. Khi bạn nhập các chữ cái vào hộp chỉnh sửa, các ký hiệu thích hợp sẽ xuất hiện trong danh sách. Bạn có thể nhấp vào mục để chọn một hoặc chỉ cần nhấn phím ENTER để chọn mục đầu tiên. Lưu ý rằng tìm kiếm bắt đầu khi hộp chỉnh sửa chứa ít nhất 1 ký tự nếu hộp này trống, sẽ không có ký hiệu nào được hiển thị trong danh sách.

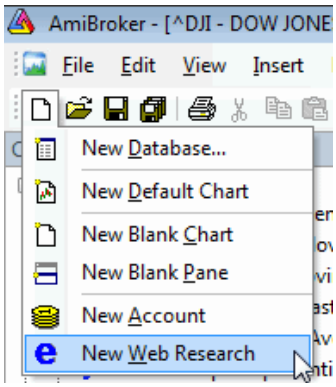
Sử dụng Cửa sổ Nghiên cứu Web

Cửa sổ Nghiên cứu Web cho phép bạn xem tin tức trực tuyến, nghiên cứu, hồ sơ, thống kê và mọi loại thông tin liên quan đến biểu tượng hiện tại được chọn có sẵn trên Internet. Việc sử dụng Cửa sổ Nghiên cứu Web thay vì trình duyệt web thông thường có ưu điểm về tốc độ, vì bạn không cần phải gõ các địa chỉ phức tạp mỗi khi muốn tìm thông tin.

Cửa sổ Nghiên cứu Web được giới thiệu trong phiên bản 4.90, thay thế và nâng cao cửa sổ Hồ sơ trước đây. Giờ đây, nó cho phép số lượng trang nghiên cứu (hồ sơ) do người dùng định nghĩa không giới hạn, duyệt đến bất kỳ trang web nào (chỉ cần gõ URL), duyệt theo tab, mở nhiều trang cùng một lúc và tự động đồng bộ hóa chọn lọc.

Nghiên cứu Web sử dụng engine Internet Explorer, vì vậy bạn có thể yên tâm rằng các trang sẽ được hiển thị với chất lượng tương tự như khi sử dụng trình duyệt độc lập.

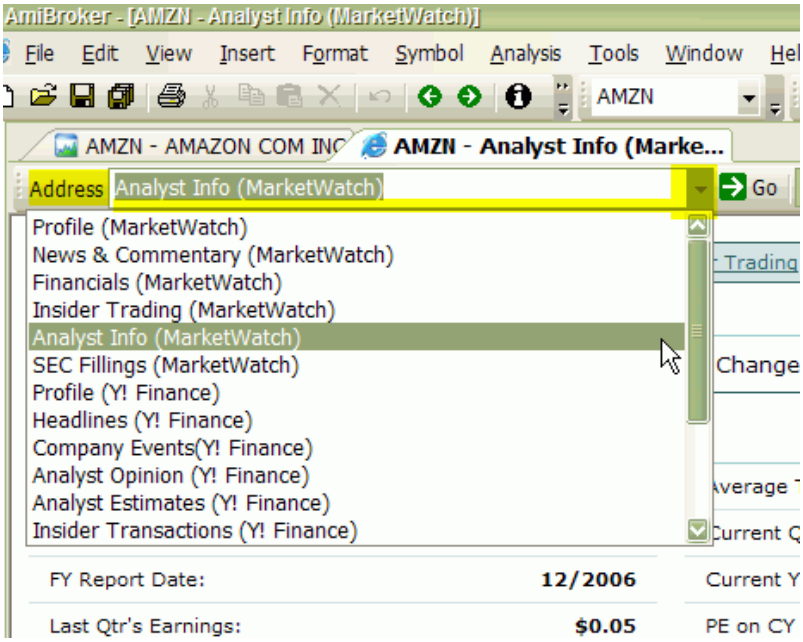
Mở Cửa sổ Nghiên cứu Web Mới



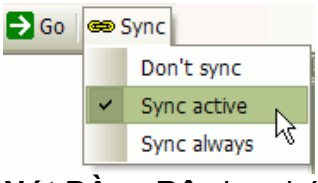
Sử dụng menu File->New->Web Research để tạo cửa sổ nghiên cứu web mới.

Chọn Trang Nghiên cứu Web Đã Định Nghĩa

Để hiển thị bất kỳ trang nghiên cứu web đã định nghĩa nào, chỉ cần nhấp vào mũi tên xuống trong hộp kết hợp Địa chỉ và chọn một mục từ danh sách. Khi bạn làm như vậy, trang web liên quan đến biểu tượng hiện tại sẽ được hiển thị tự động.



Bây giờ bạn có thể chỉ định xem trang hiển thị có nên tự động thay đổi hay không nếu bạn chọn ký hiệu khác.

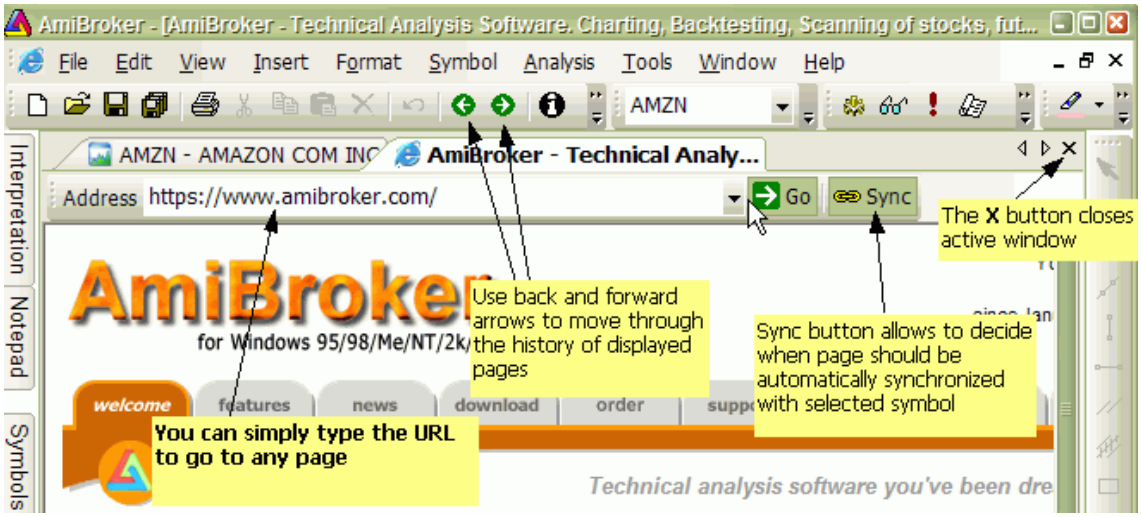


Nút Đồng Bộ cho phép bạn quyết định khi nào trang nên tự động đồng bộ hóa với biểu tượng hiện tại được chọn:

- **Không đồng bộ (Don't sync)** - Trang sẽ không được đồng bộ hóa với biểu tượng hiện tại.
- **Đồng bộ khi hoạt động (Sync active)** - Trang sẽ chỉ được đồng bộ hóa khi nó đang hoạt động hoặc trở thành hoạt động (bằng cách người dùng nhấp vào tab tương ứng). Đây là cài đặt được khuyến nghị cho các hồ sơ nghiên cứu web, vì nó giúp tiết kiệm băng thông và tài nguyên (các trang không hoạt động sẽ không được đồng bộ hóa và không tiêu tốn băng thông).
- **Đồng bộ mọi lúc (Sync always)** - Trang sẽ luôn được đồng bộ hóa với biểu tượng hiện tại, không quan trọng là nó có đang hoạt động hay không.

Điều Hướng

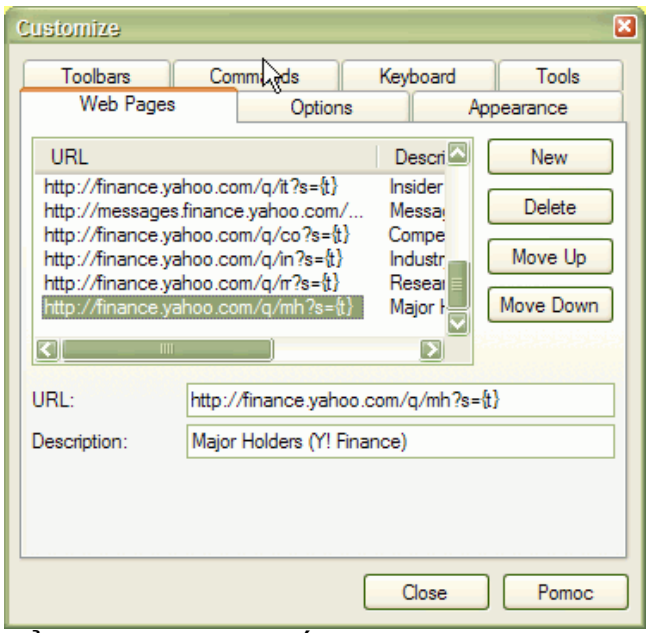
Cửa sổ Nghiên cứu Web hoạt động tương tự như trình duyệt độc lập. Để hiển thị bất kỳ trang web nào, chỉ cần nhập địa chỉ URL vào trường "Địa chỉ" và nhấn phím ENTER (RETURN). Để điều hướng quay lại và tiến tới trong lịch sử, hãy sử dụng các nút <- và ->.



Để đóng trang hiện đang hiển thị, bạn chỉ cần sử dụng nút đóng cửa sổ (X) như đã hiển thị trong hình ảnh ở trên.

Định nghĩa Những Nơi Tìm Kiếm Web Của Riêng Bạn

Ngoài các trang tìm kiếm trên web đã được định nghĩa trước, bạn có thể xác định bất kì số nơi nào của riêng bạn. Để làm như vậy, hãy sử dụng menu **Tools->Customize**, tab **Web Pages.**



Để thêm nơi mới, nhấn nút New, sau đó gõ mẫu URL vào trường URL và mô tả trang web vào trường

The URL template là địa chỉ web trong đó có các phần phụ thuộc vào ký hiệu được chọn. Mẫu URL được AmiBroker phân tích để tạo URL thực tế đến trang web. Ví dụ, để xem trang hồ sơ của Yahoo, bạn có thể sử dụng mẫu URL sau:

http://biz.yahoo.com/p/{t0}/{t}.html.

Các ký hiệu được đặt trong dấu ngoặc **{}** định nghĩa các trường sẽ được đánh giá tại thời điểm thực thi. Ký hiệu **{t0}** sẽ được đánh giá thành ký tự đầu tiên của tên ticker và **{t}** sẽ là tên đầy đủ của ticker. Ví dụ, nếu AAPL được chọn, AmiBroker sẽ tạo ra URL sau từ mẫu trên: http://biz.yahoo.com/p/a/aapl.html.

Sau đó, AmiBroker sử dụng trình duyệt web tích hợp (cửa sổ Web Research) để hiển thị nội dung của trang.

Lược đồ mã hóa các trường đặc biệt

Như được minh hoạ trong ví dụ trên, URL mẫu có thể chứa các trường đặc biệt, chúng sẽ được thay thế vào thời gian chạy bằng các giá trị tương ứng với biểu tượng hiện được chọn. Định dạng của trường đặc biệt là {x} nơi x mô tả loại trường. Hiện tại có ba loại trường được chấp nhận: tên ticker với các ký tự không thay đổi **{t}**, tên ticker viết thường **{s}**, tên ticker viết HOA **{S}**, bí danh **{a}**, id web **{i}**. Bạn có thể chỉ định những trường này ở bất kỳ đâu trong URL và AmiBroker sẽ thay thế chúng bằng các giá trị thích hợp được nhập vào trong cửa sổ Thông tin. Bạn cũng có thể tham chiếu đến các ký tự đơn lẻ của tên ticker, bí danh hoặc id web. Điều này hữu ích khi trang web sử dụng các ký tự đầu tiên của, ví dụ, tên ticker để nhóm các tệp html (trang web Yahoo Finance làm điều này), vì vậy bạn có các tệp cho tên ticker bắt đầu bằng 'a' được lưu trữ trong thư mục con 'a'. Để tham chiếu đến một ký tự đơn lẻ của trường, hãy sử dụng phong cách định dạng thứ hai {xn} nơi x là loại trường được mô tả ở trên và n là chỉ số dựa trên số 0 của ký tự. Vì vậy, **{a0}** sẽ được đánh giá thành ký tự đầu tiên của chuỗi bí danh. Để lấy hai ký tự đầu tiên của một tên ticker, hãy viết đơn giản **{t0}{t1}**. Chú ý về trường id web: trường mới này trong cửa sổ Thông tin đã được thêm vào để xử lý các tình huống khi các trang web không sử dụng tên ticker để lưu trữ các tệp hồ sơ. Tôi đã tìm thấy một số trang web sử dụng hệ thống đánh số riêng của họ nên họ gán một số duy nhất cho mỗi biểu tượng. AmiBroker cho phép bạn sử dụng mã này không tiêu chuẩn để xem hồ sơ. Tất cả những gì bạn phải làm là nhập id web chính xác vào trường ID web và sử dụng URL mẫu phù hợp với từ khóa **{i}**.

Các Trang Lưu Trữ Cục Bộ

Bạn có thể muốn lưu trữ tất cả các trang trên ổ cứng cục bộ của mình. Ưu điểm của cách này là các cấu hình có thể truy cập ngay lập tức nhưng chúng có thể chiếm một lượng lớn dung lượng lưu trữ và bạn sẽ cần cập nhật chúng theo thời gian. Để truy cập các tệp được lưu trữ cục bộ, hãy sử dụng URL mẫu sau (ví dụ: C: biểu thị ổ đĩa): file://C:\the_folder_with_profile_files\{t}.html. Bạn không bị giới hạn ở các tệp HTML, bạn có thể sử dụng các tệp TXT đơn giản thay thế. Sau đó, hãy tạo (hoặc tải xuống) các tệp .html (hoặc txt) cho từng ký hiệu trong danh mục đầu tư. Các tệp này

phải tuân theo quy ước đặt tên sau: <ticker>.html. Ví dụ, đối với APPLE (ticker AAPL), cấu hình phải có tên AAPL.html (hoặc AAPL.txt)

Cấu hình dựa trên web

Nếu bạn muốn hiển thị các cấu hình từ các trang web từ xa, bạn sẽ cần tìm hiểu cách truy cập chúng (URL đến trang web) và cách truy cập dữ liệu cho các ký hiệu khác nhau. Tôi sẽ mô tả vấn đề trên ví dụ về trang web Sharenet (www.sharenet.co.za) cung cấp dữ liệu cho các công ty được niêm yết trên Sàn giao dịch chứng khoán Johannesburg. Sharenet cung cấp thông tin công ty có thể truy cập tại địa chỉ sau (URL):

http://www.sharenet.co.za/free/free_company_na.phtml?code=JSECODE&scheme=default

Vấn đề là cơ sở dữ liệu do Sharenet cung cấp sử dụng tên mã chứng khoán dài và JSECODE là mã ký hiệu ngắn. Ví dụ đối với công ty "Accord Technologies", mã chứng khoán trong cơ sở dữ liệu Sharenet là ACCORD nhưng mã là ACR. Để giải quyết vấn đề, chúng ta sẽ cần sử dụng trường Web ID trong cửa sổ Thông tin ký hiệu. Nếu bạn có cơ sở dữ liệu Sharenet, chỉ cần chọn ACCORD từ danh sách mã chứng khoán, mở cửa sổ Symbol->Information và nhập ACR vào hộp chỉnh sửa **Web ID** rồi nhấp vào OK. Sau đó nhập mẫu URL sau vào hộp chỉnh sửa URL:

http://www.sharenet.co.za/free/free_company_na.phtml?code={i}&scheme=default

Để chắc chắn 100%, hãy chọn văn bản trên bằng chuột. Sau đó, sao chép nó vào clipboard (Edit->Copy, CTRL-C). Kế tiếp, chuyển đến AmiBroker và nhấp vào hộp chỉnh sửa URL hồ sơ. Xóa tất cả mọi thứ từ nó và nhấn CTRL-V (điều này sẽ dán văn bản). Nhập "Sharenet" vào trường **Description.**

Xin lưu ý rằng chúng tôi đã sử dụng trường đặc biệt **{i}** trong mẫu, nó sẽ được thay thế bởi AmiBroker với văn bản đã nhập vào trường Web ID của cửa sổ thông tin biểu tượng. Bây giờ, hãy chọn File->New->Web Research và chọn Sharenet từ hộp combo địa chỉ. Bạn sẽ thấy hồ sơ của công ty ACCORD.

Bạn cũng có thể xóa bất kỳ mục nào bằng cách chọn nó từ danh sách và nhấn nút **Delete**. Bạn có thể thay đổi thứ tự mà các trang xuất hiện trong hộp combo địa chỉ Web Research bằng cách sử dụng các nút **Move Up** và **Move Down** (chọn mục trước và sau đó sử dụng các nút).

Dữ liệu cấu hình được lưu trữ trong tệp văn bản webpages.cfg đơn giản mà chứa bất kỳ số lượng mẫu URL dưới dạng:

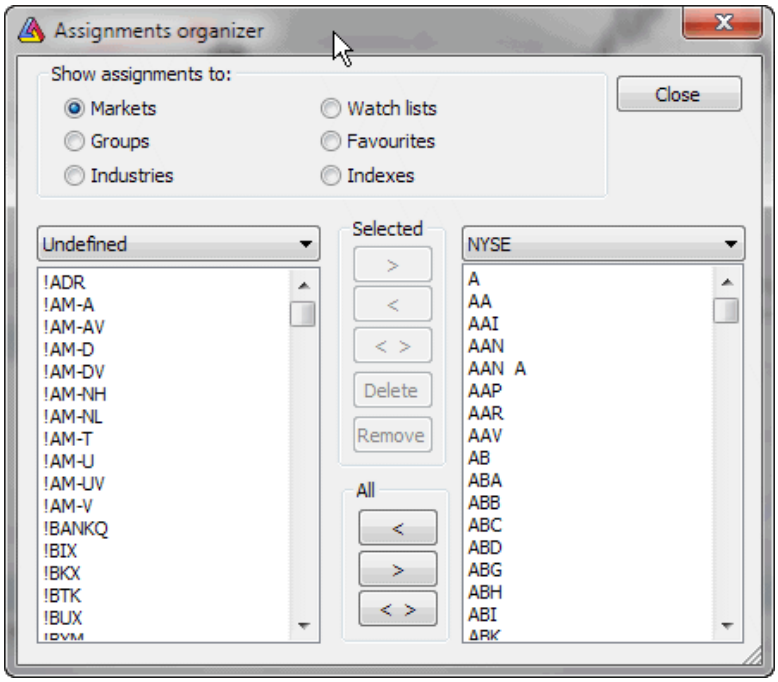
URLTemplate|Description

(mỗi mục tử trong một dòng riêng biệt)

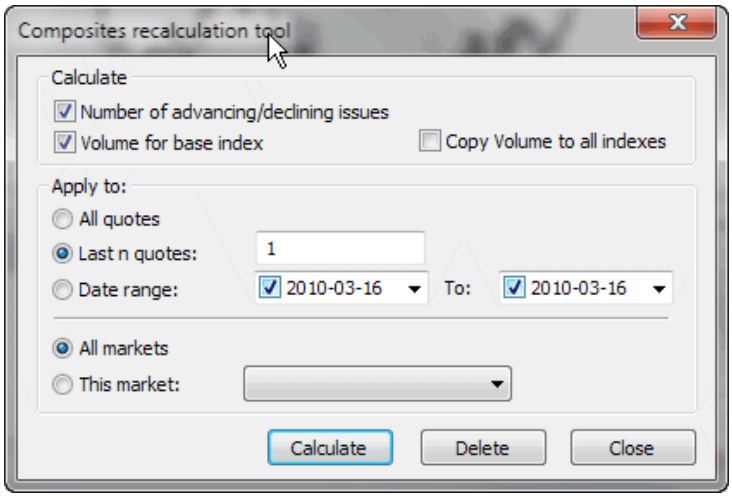
Cửa sổ Quản lý Phân công

Để việc phân công các ký hiệu vào các loại phân loại trở nên đơn giản và nhanh chóng hơn, một trình quản lý phân công mới đã được phát triển. Bây giờ bạn có thể đơn giản đánh dấu một nhóm ký hiệu và nhanh chóng di chuyển chúng từ một loại phân loại này sang loại phân loại khác.

Bạn cũng có thể xóa nhiều ký hiệu sử dụng cửa sổ này. Để làm vậy, chọn một hoặc nhiều ký hiệu từ khung bên trái và nhấn vào nút "Xóa".



Cửa sổ Tính toán lại dữ liệu Composite



Cửa sổ này cho phép tự động tính toán số lượng và khối lượng các vấn đề tiến triển/giảm đi/không thay đổi. Còn có thể trong cửa sổ này là việc tính toán số lượng khối lượng cho các chỉ số nếu được nhập không chính xác. Lưu ý rằng, việc tính toán tự động dữ liệu tổng hợp chỉ có ý nghĩa khi bạn theo dõi toàn bộ sàn giao dịch (tất cả các biểu tượng đều được bao gồm trong cơ sở dữ liệu của bạn), nếu không thì việc tính toán này sẽ cho kết quả sai.

Để tính toán các giá trị tổng hợp trong cơ sở dữ liệu, cần đặt chỉ số cơ sở cho thị trường, vì có thể xảy ra việc không phải tất cả các cổ phiếu được báo giá mỗi ngày làm việc. AmiBroker kiểm tra thời gian giao dịch của 'chỉ số cơ sở' và cố gắng tìm các báo giá tương ứng của tất cả các cổ phiếu thuộc thị trường đó, để tìm ra bao nhiêu vấn đề đã tiến triển, giảm đi và không thay đổi.

Để tính toán giá trị tổng hợp, bạn cần làm следующее:

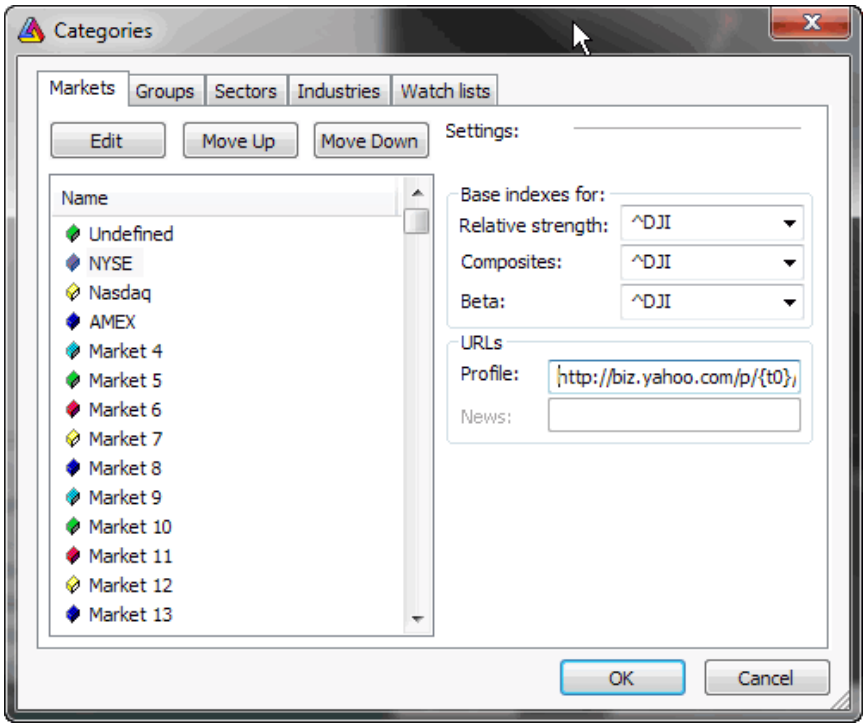
- Mở cửa sổ Danh mục bằng cách sử dụng mục menu Symbol -> Categories.
- Chọn chỉ số cơ sở cho thị trường đã cho trong tab Markets và Base indexes for - Composites combo. Ví dụ, nếu bạn theo dõi NYSE thì có thể là ^DJI (Dow Jones Average) ^DJI phải được đánh dấu là chỉ số trong Symbol -> Information và phải thuộc về cùng một thị trường
- Chọn mục menu "Symbol ->Calculate composites" để mở cửa sổ được hiển thị dưới đây và đánh dấu: Số vấn đề tiến triển/giảm đi và chọn thị trường mà bạn tính toán giá trị tổng hợp và khoảng thời gian.
- Click vào Tính toán.

Cũng có hai trường bổ sung có sẵn:

- Khối lượng cho chỉ mục cơ sở
- Sao chép khối lượng vào tất cả các chỉ mục

Những trường này được cung cấp trong trường hợp bạn KHÔNG có dữ liệu khối lượng thực tế cho các báo giá chỉ số. Trong trường hợp đó, AmiBroker có thể tính toán khối lượng cho chỉ số như là tổng của khối lượng của tất cả các cổ phiếu thuộc thị trường đã cho. Lựa chọn đầu tiên chỉ định khối lượng được tính toán chỉ cho chỉ số cơ sở, lựa chọn thứ hai sao chép số liệu khối lượng vào tất cả các chỉ số thuộc thị trường đã cho.

Cửa sổ Danh mục



Hộp thoại này cho phép bạn định nghĩa tên của các thị trường, nhóm, ngành và công nghiệp. Đối với mỗi thị trường, bạn cũng có thể định nghĩa chỉ mục cơ sở để tính toán sức mạnh tương đối, dữ liệu tổng hợp, beta hoặc URL hồ sơ web. Thông tin chi tiết về các danh mục có thể được tìm thấy trong chương Hiểu về danh mục của hướng dẫn này.

Để Chỉnh sửa tên của một danh mục nhất định, vui lòng chọn nó từ danh sách và nhấn nút 'Chỉnh sửa'.

Chỉ mục cơ sở cho các trường cho phép bạn đặt chỉ mục được sử dụng trong việc tính toán của:

- Chỉ số Sức mạnh tương đối
- Cấu hình tổng hợp thông qua tùy chọn tính toán tổng hợp
- Beta

Trường Hồ sơ cho phép bạn định nghĩa URL-mẫu để xem trực tuyến (hoặc ngoại tuyến) hồ sơ của các công ty. Những URL-mẫu này dựa trên thị trường, tức là bạn có thể có các mẫu khác nhau cho mỗi thị trường. Mẫu sau đó sẽ được phân tích cú pháp để tạo URL thực tế đến trang web, sẽ được hiển thị trong trình duyệt web nhúng. Để tìm hiểu thêm, hãy đọc chương Cách thiết lập chế độ xem hồ sơ.

Sắp xếp lại Danh mục

Phiên bản AmiBroker 6.10 đã thêm khả năng sắp xếp lại thứ tự các thị trường/nhóm/ngành/công nghiệp và danh sách theo dõi bằng cách sử dụng nút "Di chuyển lên" / "Di chuyển xuống". Đây là nhiệm vụ không hề đơn giản vì tất cả các biểu tượng phải được đồng bộ hóa với sự thay đổi, khi thứ tự của các danh mục thay đổi thì tất cả dữ liệu

biểu tượng phải được tái chỉ mục để phản ánh thứ tự mới vì biểu tượng tham chiếu đến vị trí theo thứ tự của danh mục.

Trình Wizard Nhập Dữ Liệu ASCII

Trình Wizard Nhập Dữ Liệu ASCII cung cấp cách dễ dàng để nhập dữ liệu báo giá của bạn cũng như định nghĩa các định dạng nhập riêng cho việc sử dụng trong tương lai. Lưu ý rằng trình wizard chỉ cung cấp một tập hợp các tính năng cơ bản có sẵn trong trình nhập dữ liệu ASCII, do đó nó được thiết kế cho người dùng mới bắt đầu.

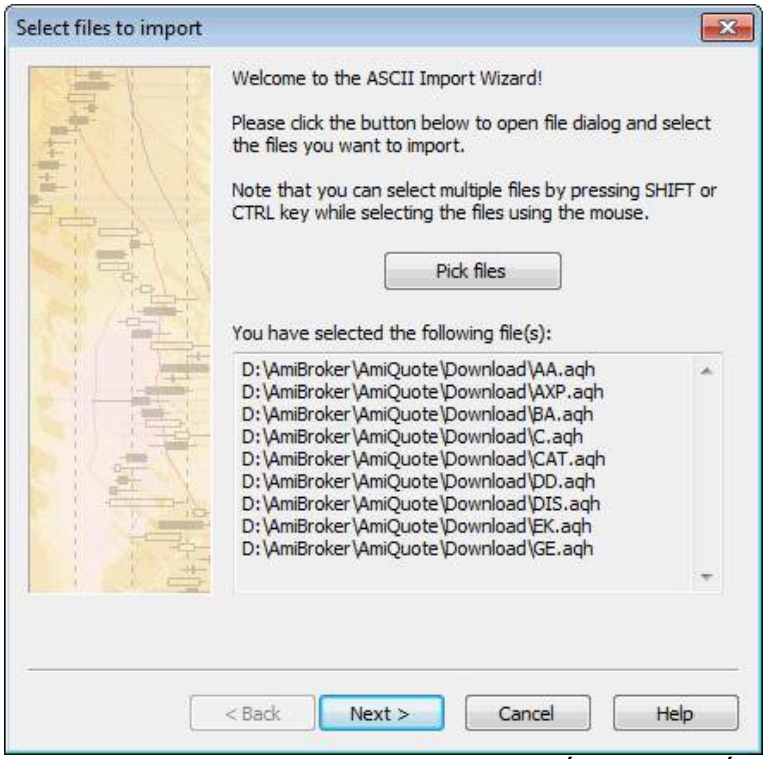
Trình wizard hướng dẫn bạn qua 3 bước đơn giản:

1. Chọn tệp cần nhập

2. Định nghĩa các trường

3. Cài đặt bổ sung

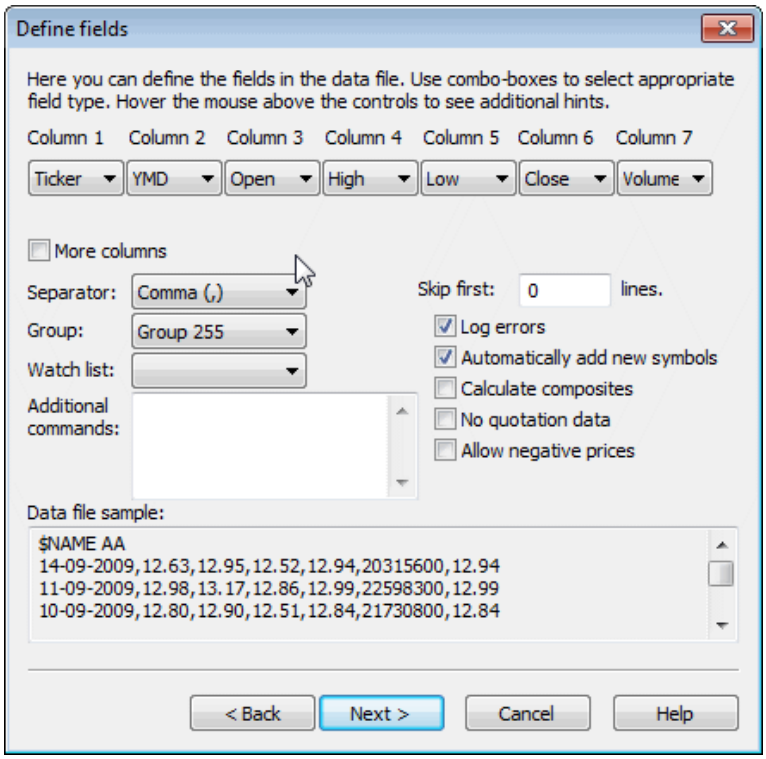
Bước 1: Chọn Tệp



Trong bước này, bạn chọn các tệp bạn muốn nhập. Nhấn vào nút "Pick files" để mở hộp thoại tệp. Duyệt đến thư mục nơi các tệp dữ liệu của bạn được lưu và chọn tệp (hoặc tệp) cần nhập. Bạn có thể chọn nhiều tệp bằng cách giữ phím CTRL hoặc SHIFT trong khi nhấp vào các tệp. Sau khi chọn xong, nhấn "Open".

Danh sách đầy đủ các tệp bạn đã chọn sẽ được hiển thị ở phần dưới cùng của cửa sổ wizard. Vui lòng kiểm tra xem danh sách có chính xác không; nếu không, nhấn "Pick files" để sửa lại lựa chọn của bạn.

Bước 2: Định Nghĩa Các Trường



Trong bước này, bạn định nghĩa các loại trường trong tệp dữ liệu. Để thuận tiện cho bạn, mẫu tệp ngày được hiển thị (một vài dòng đầu tiên của tệp đầu tiên được chọn) ở cuối cửa sổ.

Để định nghĩa các trường, vui lòng chọn các loại trường phù hợp từ các hộp kết hợp **Cột N**. Ví dụ: nếu trường đầu tiên (cột) trong tệp dữ liệu của bạn là mã chứng khoán ký hiệu, vui lòng chọn "Mã chứng khoán" từ hộp kết hợp **Cột 1**. Nếu trường thứ hai trong tệp dữ liệu của bạn là ngày theo định dạng Năm-Tháng-Ngày, vui lòng chọn "YMD" từ hộp kết hợp thứ hai. Bạn cũng có thể chọn DMY cho ngày Ngày-Tháng-Năm, MDY cho ngày Tháng-Ngày-Năm. Các loại trường khác có sẵn từ trình hướng dẫn là: "Mở", "Đóng", "Cao", "Thấp" cho giá và "Khối lượng".

Lưu ý về ngày: AmiBroker nhận dạng cả ngày năm 4 chữ số và 2 chữ số. Đối với tháng, cả số và mã ba chữ cái ("Tháng 1", "Tháng 2", ...) đều được phép. Ngoài ra, ngày, tháng và năm có thể được phân tách bằng bất kỳ ký tự nào sau đây: / (gạch chéo), \ (gạch chéo ngược), - (dấu trừ) hoặc có thể không được phân tách. Tất cả những gì bạn phải làm là chỉ định thứ tự: DMY, MDY, YMD. Ví dụ, ngày YMD hợp lệ là (31 tháng 12 năm 2000):

20001231,

001231,

2000-12-31

2000/12/31

2000-Dec-31

00-12-31

00/12/31

00\12\31

Nếu tệp của bạn có nhiều hơn 7 cột, vui lòng chọn hộp More columns và bạn sẽ thấy các combo-box bổ sung.

Các điều khiển còn lại ở đây là:

Group: tại đây, bạn nên chọn nhóm ký hiệu mới được thêm vào

Chọn Danh Sách Theo Dõi: Tại đây, bạn cần chọn danh sách theo dõi mà các biểu tượng mới sẽ được thêm vào. Nếu không chọn, các biểu tượng sẽ không được thêm vào bất kỳ danh sách nào.

Ký Tự Phân Cách: Chọn ký tự được sử dụng làm ký tự phân cách cho các trường (dấu phẩy là phổ biến nhất).

Bỏ Qua Dòng: Đây là nơi bạn xác định số dòng đầu tiên cần bị bỏ qua (không được xem xét). Ví dụ, một vài dòng đầu tiên có thể chứa nhận xét hoặc thông tin khác cần bị bỏ qua.

Ghi Lại Lỗi: Tùy chọn này cho biết AmiBroker nên ghi lại tất cả lỗi vào tệp (import.log). Trong trường hợp có lỗi, tệp log này sẽ được hiển thị cho người dùng sau khi quá trình nhập hoàn tất.

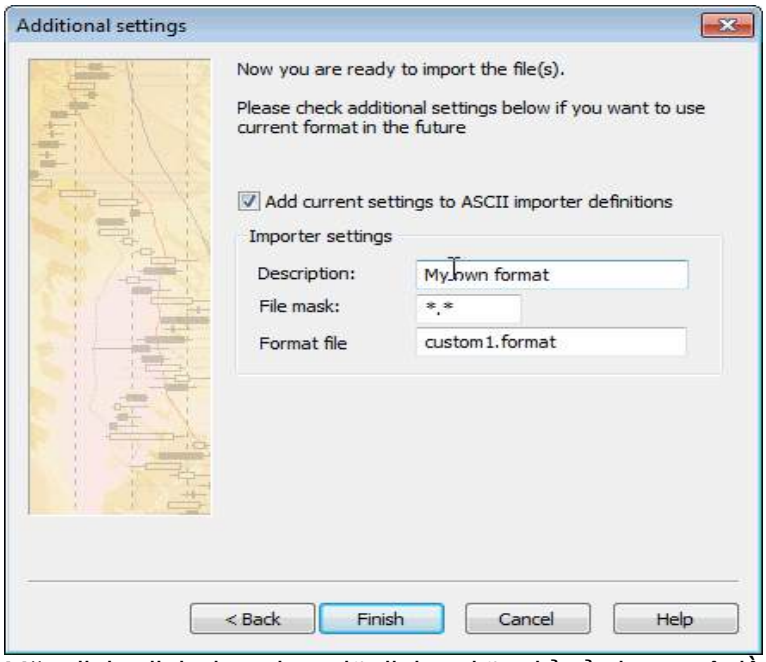
Tự Động Thêm Biểu Tượng Mới: Tùy chọn này yêu cầu AmiBroker thêm các biểu tượng xuất hiện trong tệp dữ liệu nhưng chưa có trong cơ sở dữ liệu của AmiBroker.

Tính Toán Composite: Tùy chọn này yêu cầu AmiBroker tính toán các chỉ số tiến/giảm và khối lượng cho các chỉ số sau khi nhập (điều này yêu cầu các chỉ số phải được thiết lập đúng trước khi nhập).

Cho Phép Giá Âm: Tùy chọn này cho phép các giá trị âm trong các trường đóng, mở, cao, thấp. Mặc định, giá trị bằng 0 và âm sẽ không được phép.

Không Có Dữ Liệu Báo Giá: Cho phép nhập dữ liệu không chứa giá cả, chẳng hạn như danh sách mã chứng khoán và/hoặc danh mục.

Bước 3: Cài Đặt Bổ Sung



Mặc định, định dạng bạn đã định nghĩa chỉ sử dụng một lần. Điều này phù hợp cho người dùng mới bắt đầu và cho việc thử nghiệm với trình wizard.

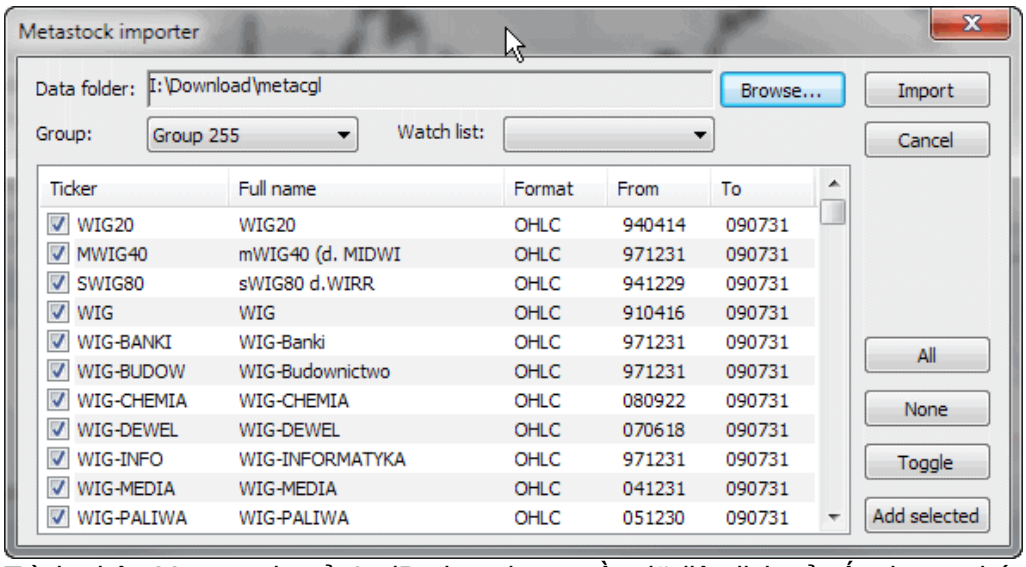
Nếu bạn muốn tạo ra một định nghĩa vĩnh viễn và có thể sử dụng trong tương lai qua ASCII importer, bạn nên đánh dấu vào hộp kiểm Add current settings to ASCII importer definitions. Sau đó, bạn cần nhập vào Format description, File mask và Format file name (hoặc bạn có thể chấp nhận các giá trị mặc định được tạo tự động). Nếu bạn làm như vậy, bạn sẽ có thể sử dụng định dạng đã xác định trong cửa sổ ASCII importer - chỉ bằng cách chọn định dạng của riêng bạn (như đã nhập vào trường Format description) từ combo "Files of type" của hộp thoại file.

Bất kể lựa chọn của bạn, bạn nên nhấn nút "Finish" để bắt đầu nhập dữ liệu của mình.

Cửa sổ Nhập Metastock

LƯU Ý QUAN TRỌNG: Trình nhập Metastock chỉ nên được sử dụng CHỈ khi bạn muốn nhập dữ liệu MS vào cơ sở dữ liệu AmiBroker cục bộ, gốc một lần. Nếu bạn muốn AmiBroker chỉ đọc cơ sở dữ liệu Metastock TRỰC TIẾP mà không cần nhập dữ liệu mới liên tục, vui lòng thiết lập cơ sở dữ liệu CÓ PLUGIN METASTOCK như mô tả trong Hướng dẫn.

LƯU Ý 2: nếu bạn thiết lập cơ sở dữ liệu với Plugin MS, bạn KHÔNG nên sử dụng trình nhập Metastock, vì không có ý nghĩa gì khi dữ liệu của bạn đã được cung cấp bởi plugin.



Trình nhập Metastock mở AmiBroker cho nguồn dữ liệu lịch sử rất phong phú. Trình nhập hỗ trợ cả định dạng Metastock cũ 6.5 và mới 7.x (XMASTER).

Cơ bản, dữ liệu Metastock bao gồm:

- Tệp MASTER/EMASTER chứa thông tin tổng quan về các mã chứng khoán, tên cổ phiếu, v.v.

- Các tệp F1.DAT....Fxx.DAT chứa dữ liệu báo giá thực tế.

Tệp MASTER/EMASTER quan trọng vì nó chứa các tham chiếu đến các tệp Fxx.DAT. Các tệp Fxx.DAT chỉ lưu trữ báo giá trong định dạng 5 trường (ngày/cao/thấp/đóng/khối lượng), 6 hoặc 7 trường (ngày/mở/caothấp/đóng/khối lượng/quyền mua). Như bạn thấy, các tệp MASTER/EMASTER và Fxx.DAT liên quan chặt chẽ và bạn cần tất cả chúng để nhập dữ liệu.

Sử dụng

Để nhập dữ liệu Metastock, bạn nên làm như sau:

Chọn nhập Metastock từ menu Sử dụng yêu cầu thư mục (Browse...) chọn vị trí của dữ liệu theo định dạng Metastock (thư mục chứa tệp MASTER/EMASTER và Fxx.DAT)

Sau khi chọn đúng thư mục, AmiBroker sẽ hiển thị danh sách các biểu tượng và phạm vi ngày có sẵn. Theo mặc định, tất cả các biểu tượng có sẵn sẽ được đánh dấu để nhập khẩu (dấu kiểm ở đầu danh sách). Bây giờ bạn có thể loại bỏ một số biểu tượng khỏi danh sách nhập bằng cách nhấp vào mục phù hợp trong danh sách (dấu kiểm sẽ chuyển đổi khi bạn nhấp).

Bạn có thể tự quyết định biểu tượng mới được thêm vào nhóm và danh sách theo dõi nào bằng cách sử dụng các hộp tổ hợp Group (Nhóm) và Watch List (Danh sách theo dõi).

Sau khi lựa chọn, nhấn nút 'Import' (Nhập) để bắt đầu quá trình nhập.

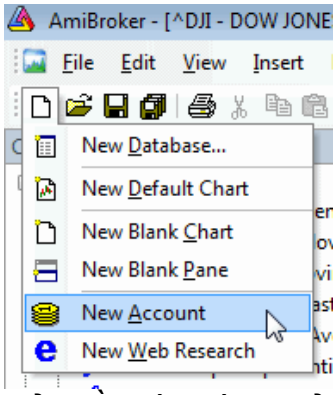
Trong quá trình này, bạn có thể hủy bỏ thao tác bằng cách nhấn nút 'Abort' (Hủy bỏ) trong cửa sổ tiến trình.

Sử dụng Account Manager

Account Manager là công cụ để theo dõi các giao dịch và hiệu suất của bạn. Bạn có thể nhập các giao dịch bạn thực hiện, nạp/rút tiền, kiểm tra số liệu thống kê và hiệu suất lịch sử. Tất cả các giao dịch đều được ghi nhận nên bạn sẽ không bao giờ quên những gì đã xảy ra trong quá khứ. Account Manager cho phép bạn theo dõi số lượng tài khoản không giới hạn. Account Manager mới thay thế và tăng cường chức năng do portfolio manager cung cấp trong các phiên bản trước 4.90.

Tạo Một Tài Khoản Mới

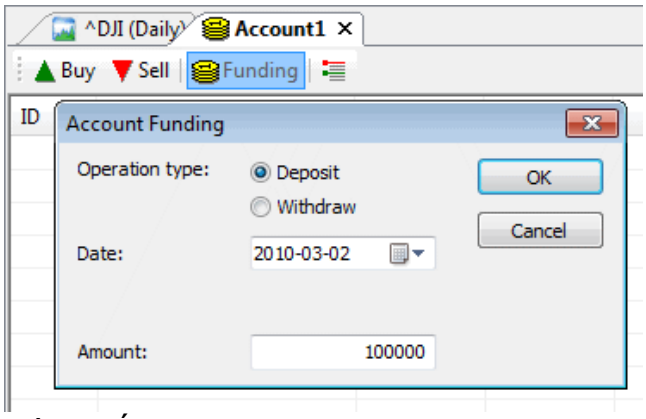
Sử dụng menu File->New->Account để tạo tài khoản mới



Gửi Tiền vào Tài Khoản

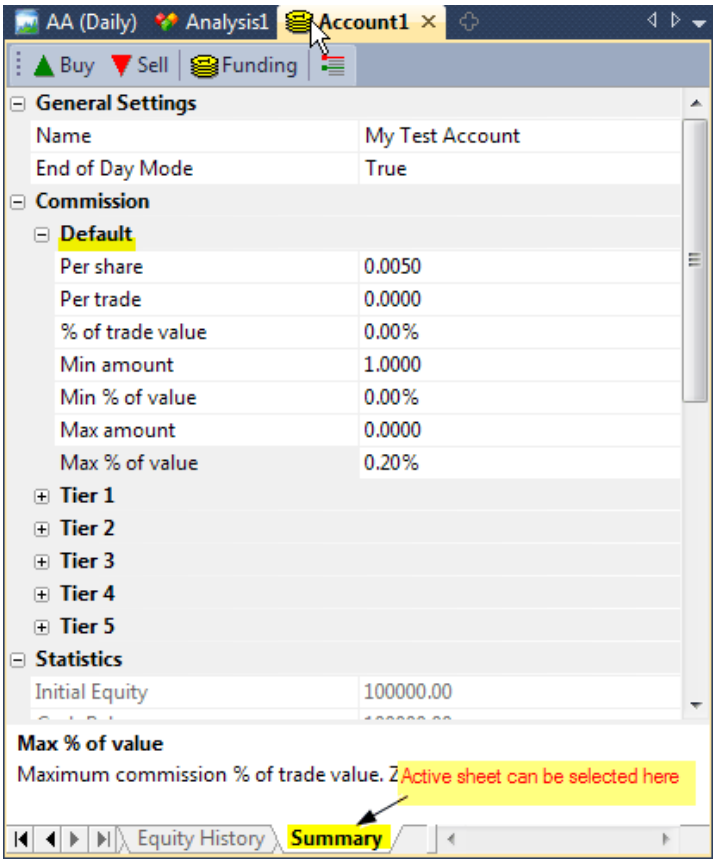
Trước khi bạn thực hiện bất kỳ giao dịch nào, bạn phải gửi tiền vào tài khoản của mình. Để làm điều đó, nhấn nút "FUNDING" trên thanh công cụ của account manager, sau đó chọn "Deposit" như loại hoạt động, nhập NGÀY bạn đã nạp tiền vào tài khoản và nhập số tiền.

Lưu ý rằng ngày gửi tiền phải TRƯỚC mọi giao dịch, vì account manager sẽ không cho phép bạn giao dịch trước ngày nạp tiền. Khoản gửi ban đầu sẽ hiển thị dưới dạng "initial equity" trong tab tóm tắt.



Các Thiết Lập

Ý tưởng tốt là đi đến "Summary tab" và thiết lập hoa hồng và chế độ giao dịch. Nếu tài khoản này được sử dụng cho giao dịch hết ngày, bạn nên đặt "EOD Mode" thành YES, nếu không (nếu bạn giao dịch intraday) bạn nên đặt "EOD Mode" thành "Không". Tùy thuộc vào cài đặt này, hộp thoại Mua/Bán sẽ cho phép bạn nhập ngày và thời gian giao dịch hoặc chỉ ngày.



Bảng phí giao dịch cho phép bạn nhập cả phí chia sẻ (phí hợp đồng) và phí được biểu thị dưới dạng phần trăm giá trị giao dịch. Hoặc là sự kết hợp của cả hai. Bạn cũng có thể đặt giá trị tối thiểu và tối đa được biểu thị bằng số tiền đô la và/hoặc phần trăm giá trị giao dịch. Ví dụ, nếu sàn giao dịch của bạn sử dụng 0.01$ (một xu) cho mỗi cổ phần, bạn sẽ sử dụng PerShare = 0.01 và %OfTradeValue = 0. Nếu sàn giao dịch của bạn sử dụng, hãy nói là 0.2% giá trị giao dịch thì bạn sẽ sử dụng PerShare = 0 và %OfTradeValue = 0.2.

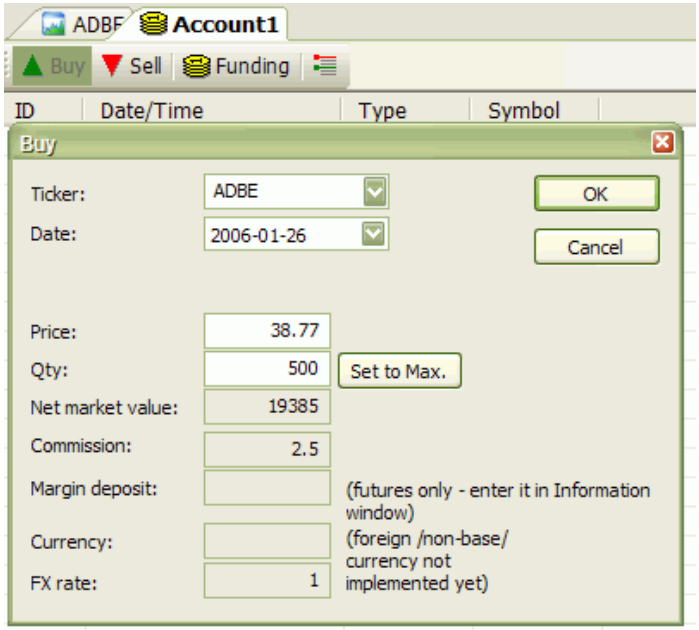
Ví dụ thực tế: Phí mặc định của Interactive Brokers cho các cổ phiếu Mỹ là: 0.005 mỗi cổ phiếu nhưng không dưới 1 đô la và không quá 0.2% giá trị giao dịch. Các cài đặt phù hợp cho lịch trình như vậy được hiển thị trong ảnh chụp màn hình ở trên.

Bảng phí giao dịch hoạt động như sau: trước tiên tổng số phí mỗi cổ phiếu và % giá trị giao dịch được tính toán. Sau đó, kết quả được kiểm tra với giới hạn tối thiểu và tối đa và nếu giá trị được tính toán vượt quá giới hạn thì phí giao dịch sẽ được thiết lập là giá trị của giới hạn đó, ngược lại, giá trị được tính toán sẽ được sử dụng mà không thay đổi.

Trang Tóm tắt chứa một chút thông tin thống kê cơ bản.

Thực hiện giao dịch

Sau khi bạn gửi tiền vào tài khoản, bạn có thể thực hiện giao dịch. Để mua (vào vị trí dài hoặc đắp thêm vị trí bán) nhấp vào nút "MUA".



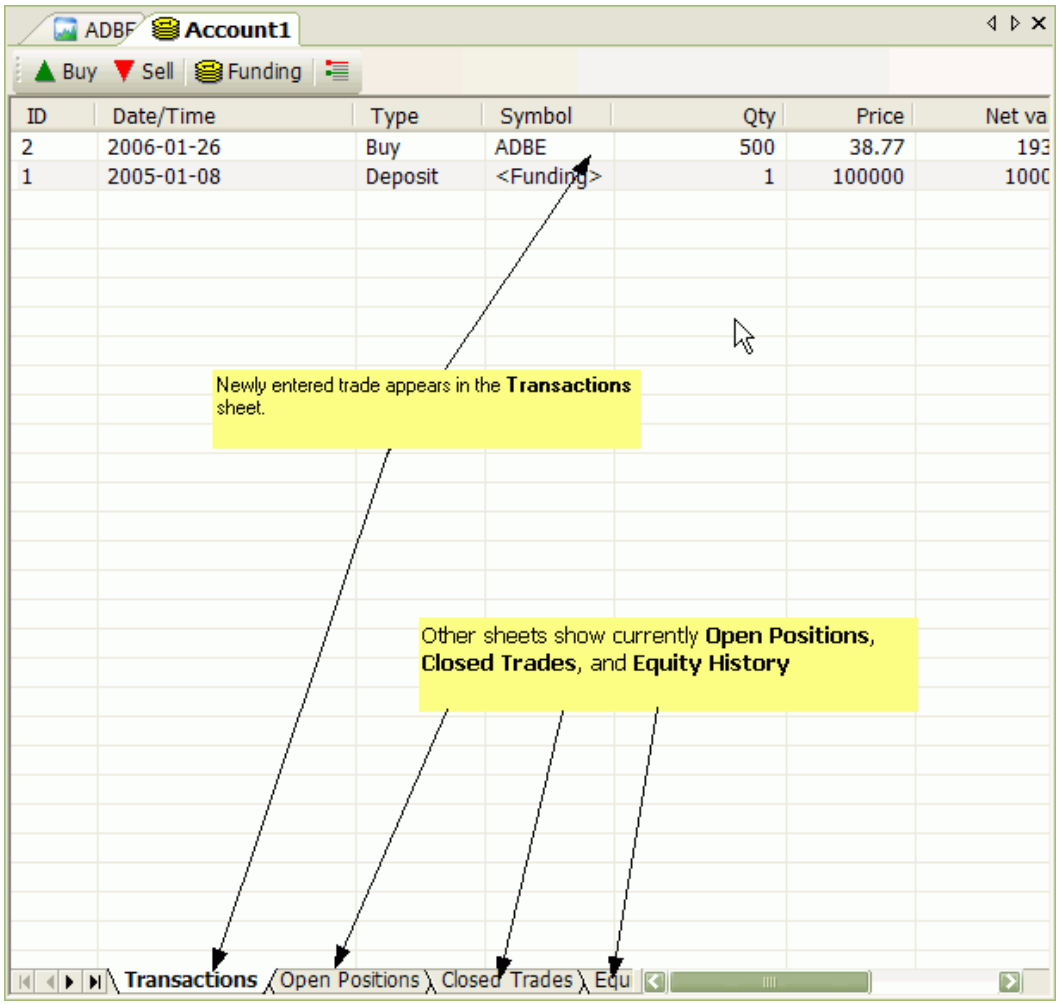
Trong hộp thoại Mua, bạn cần chọn ký hiệu và ngày/giờ giao dịch. Khi được nhập, AmiBroker sẽ hiển thị giá của ký hiệu đã chọn tại ngày/giờ đó (hoặc giá gần nhất trước đó nếu không tìm thấy giá chính xác). Nó cũng sẽ tính toán số lượng tối đa có thể giao dịch, dựa trên giá và số tiền có sẵn.

Bạn có thể thay đổi giá và số lượng một cách thủ công.

Tất cả các giá trị khác (giá trị thị trường ròng, hoa hồng, tiền gửi thị trường, tiền tệ, tỷ giá) sẽ được tính toán hoặc lấy tự động từ trang Thông tin Ký hiệu. Khi các giá trị đã chính xác, hãy nhấn OK để xác nhận giao dịch. Nếu bạn mắc sai sót, bạn có thể nhấn UNDO (Chỉnh sửa -> Hoàn tác) để khôi phục giao dịch cuối cùng.

Quy trình tương tự cũng áp dụng cho việc bán (mở vị thế ngắn hoặc đóng vị thế dài), chỉ khác là bạn nên nhấn nút "BÁN".

Tất cả các giao dịch bạn đã thực hiện sẽ được liệt kê trong bảng "Giao dịch". Tất cả các vị thế mở sẽ được liệt kê trong bảng "Vị thế Mở". Nếu bạn thực hiện giao dịch cho ký hiệu đã có vị thế mở, AmiBroker sẽ điều chỉnh "vị thế mở" tương ứng (thực hiện việc tăng/giảm vị thế). Khi vị thế mở được đóng, nó sẽ bị xóa khỏi danh sách "vị thế mở" và chuyển sang bảng "Giao dịch Đã Đóng".



Sau mỗi giao dịch, bảng "Lịch sử Vốn" sẽ được cập nhật với giá trị vốn hiện tại của tài khoản và bảng "Tóm tắt" cũng sẽ được cập nhật với các số liệu giao dịch cơ bản (sẽ có thêm nhiều số liệu khác trong tương lai).

QUAN TRỌNG

Bạn cần nhớ rằng phải nhập tất cả giao dịch theo thứ tự thời gian (giao dịch cũ trước, giao dịch mới sau), vì quản lý tài khoản sẽ không cho phép bạn thêm giao dịch ngoài thứ tự. Nếu bạn mắc sai sót, có một tùy chọn hoàn tác một cấp mà bạn có thể sử dụng để trở lại trạng thái trước giao dịch cuối cùng. Nếu bạn mắc nhiều sai sót, lựa chọn duy nhất là đóng tài khoản mà không lưu và mở lại tệp gốc.

LƯU DỮ LIỆU TÀI KHOẢN

Để lưu các chỉnh sửa đã thực hiện trên tài khoản, hãy sử dụng Tệp -> Lưu (hoặc Tệp -> Lưu dưới dạng để lưu với tên mới). Lưu ý rằng **các tệp tài khoản hiện không được mã hóa**, và bất kỳ ai có quyền truy cập đều có thể dễ dàng đọc tệp này. Vì vậy, hãy đảm bảo không để tệp của bạn trên máy tính công cộng. Chức năng bảo vệ bằng mật khẩu/mã hóa đang được lên kế hoạch nhưng vẫn chưa được triển khai.

MỞ TÀI KHOẢN ĐÃ TẠO TRƯỚC

Để mở tệp tài khoản, hãy vào Tệp -> Mở, trong hộp thoại Tệp, chọn "Tài khoản (*.acx)" từ danh sách "Loại tệp", và chọn tệp tài khoản bạn muốn tải.

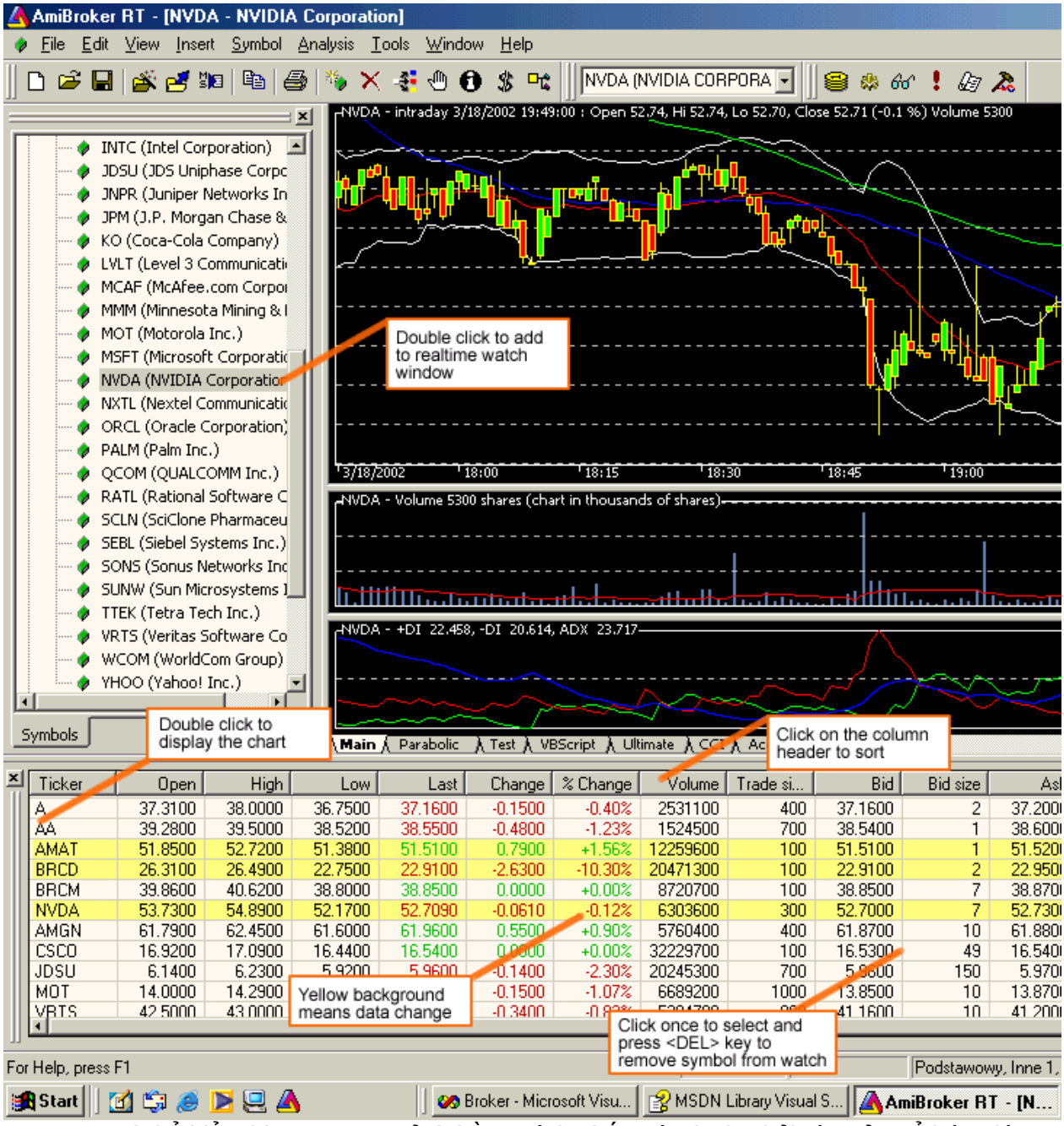
TÀI KHOẢN ĐA DẠNG

Bạn có thể tạo/mở nhiều tài khoản cùng một lúc (chỉ cần sử dụng File->New->Account, File->Open nhiều lần).

Cửa sổ báo giá thời gian thực

Làm việc với cửa sổ báo giá thời gian thực

Cửa sổ báo giá RT cung cấp các báo giá streaming thời gian thực và một số dữ liệu cơ bản. Nó khá dễ sử dụng, như trong hình dưới đây:



Bạn cũng có thể hiển thị menu ngữ cảnh bằng cách nhấn nút chuột phải vào cửa sổ báo giá RT.



Menu ngữ cảnh cho phép bạn truy cập các tùy chọn sau:

Time & Sales

Mở cửa sổ "Time & Sales" cung cấp thông tin về mỗi lượt đấu giá, hỏi và giao dịch từ thị trường.

Easy Alerts

Mở cửa sổ "Easy Alerts" cung cấp cách để định rõ các cảnh báo theo thời gian thực khi giá đấu/bán và các trường khác đạt đến mức do người dùng định trước.

Add Symbol

Thêm biểu tượng hiện tại vào danh sách Báo giá theo thời gian thực.

Thêm danh sách theo dõi

Thêm toàn bộ danh sách theo dõi vào cửa sổ báo giá thời gian thực

Nhập ký hiệu

Cho phép nhập trực tiếp các ký hiệu dưới dạng danh sách phân tách bằng dấu phẩy.

Chèn dòng trống

Thêm một dòng trống (dòng phân tách) - hữu ích để nhóm các ký hiệu lại với nhau.

Xóa Ký hiệu

Xóa dòng (ký hiệu) đang được chọn khỏi danh sách báo giá thời gian thực.

Xóa Tất cả

Xóa tất cả các ký hiệu khỏi danh sách báo giá thời gian thực.

Ẩn

Ẩn danh sách báo giá thời gian thực.

Sắp xếp lại các ký hiệu bằng cách kéo và thả

Cửa sổ báo giá thời gian thực (từ phiên bản 5.10 trở đi) cho phép bạn sắp xếp lại danh sách ký hiệu bằng cơ chế kéo và thả. Chỉ cần nhấn chuột trái vào ký hiệu, giữ nó xuống và kéo đến vị trí mong muốn, sau đó thả nút chuột.

Chỉ báo xu hướng giá mua/bán

Last	Change	% Change	Volume	Trade size	Bid/Ask Trend
3.3940	0.0149	+0.44%	13,465	1	
1.2438	-0.0036	-0.29%	91,126	1	
175.8000	-0.2200	-0.12%	200	200	

Phiên bản 5.90 thêm vào Bid/Ask trend - một chỉ số đồ họa biểu thị hướng của 10 sự thay đổi gần nhất trong giá đấu thầu/ đề nghị thời gian thực. Hộp ở bên phải cùng là gần nhất và khi các báo giá bid/ask mới tới, chúng được dịch sang phía trái. Mã màu là như sau:

- Màu xanh lá cây đậm: bid > bid trước hoặc ask > ask trước

- Màu xanh lá cây sáng: bid > bid trước và ask > ask trước

- Màu đỏ đậm: bid < bid trước hoặc ask < ask trước

- Màu đỏ sáng: bid < bid trước và ask < ask trước

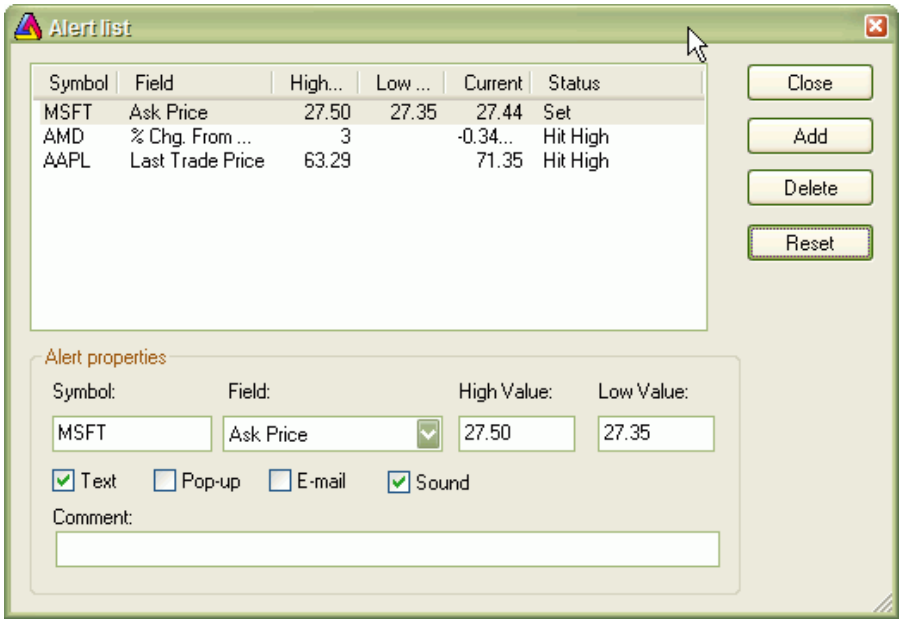
- Hộp màu đỏ / xanh lá cây: ask < ask trước và bid > bid trước

- Hộp màu xanh lá cây / đỏ: ask > ask trước và bid < bid trước

Nếu giá bid/ask không thay đổi thì không có hộp mới. LƯU Ý: Cột này chỉ hoạt động nếu có dòng báo giá thời gian thực (thị trường đang mở)

Cửa sổ Cảnh báo Dễ Dàng

Cửa sổ cảnh báo dễ dàng cho phép bạn định nghĩa các cảnh báo thời gian thực mà không cần lập trình. Xin lưu ý rằng chức năng này chỉ có sẵn nếu bạn đang sử dụng plugin dữ liệu thời gian thực và không khả dụng trong chế độ kết thúc ngày.



Thêm Cảnh Báo Mới

- Nhấn nút **Thêm.**

- Nhập ký hiệu vào trường **Ký hiệu.**

- Chọn trường giá từ hộp chọn **Trường**

- Nhập giá trị kích hoạt cao và/hoặc giá trị kích hoạt thấp.

- Chọn ít nhất một trong các trường **Văn bản/Thông báo/Email/Âm thanh.**

Cảnh báo sẽ được tạo khi trường giá đã chọn (ví dụ: Giá chào bán) trở thành bằng hoặc lớn hơn giá trị cao (nếu đã định nghĩa), hoặc khi trường giá đã chọn trở thành bằng hoặc nhỏ hơn giá trị thấp (nếu đã định nghĩa). Cảnh báo sẽ không kích hoạt lại cho đến khi bạn nhấn **Đặt lại.**

Chỉnh sửa Cảnh Báo

Chọn một trong các cảnh báo đã liệt kê và chỉnh sửa giá trị trong các trường chỉnh sửa bên dưới. Nếu bạn muốn chỉnh sửa một cảnh báo đã được kích hoạt, sau khi thực hiện các chỉnh sửa, hãy nhấn nút **Đặt lại**

Xóa Cảnh Báo

Chọn một hoặc nhiều cảnh báo từ danh sách (có thể chọn nhiều bằng cách giữ phím SHIFT) và sau đó nhấn nút **Xóa.**

Đặt lại Cảnh Báo Đã Kích Hoạt

Cảnh báo đã được kích hoạt sẽ được đánh dấu là "Đã đạt cao" hoặc "Đã đạt thấp" trong trường trạng thái và trở thành không hoạt động (sẽ không kích hoạt nữa). Nếu bạn muốn kích hoạt lại, hãy chọn nó từ danh sách và nhấn nút **Đặt lại.**

Các Loại Đầu Ra Cảnh Báo

Văn bản

khi hộp kiểm này được đánh dấu, khi cảnh báo được kích hoạt, văn bản được định nghĩa trong trường bình luận sẽ được hiển thị trong cửa sổ Alert Output (sử dụng menu Window->Alert output để hiển thị nó)

Pop-up

khi hộp kiểm này được đánh dấu, cảnh báo kích hoạt sẽ hiển thị hộp thoại pop-up

E-mail

khi hộp kiểm này được đánh dấu, cảnh báo kích hoạt sẽ gửi một e-mail đến tài khoản được định nghĩa trong Preferences/Alerts.

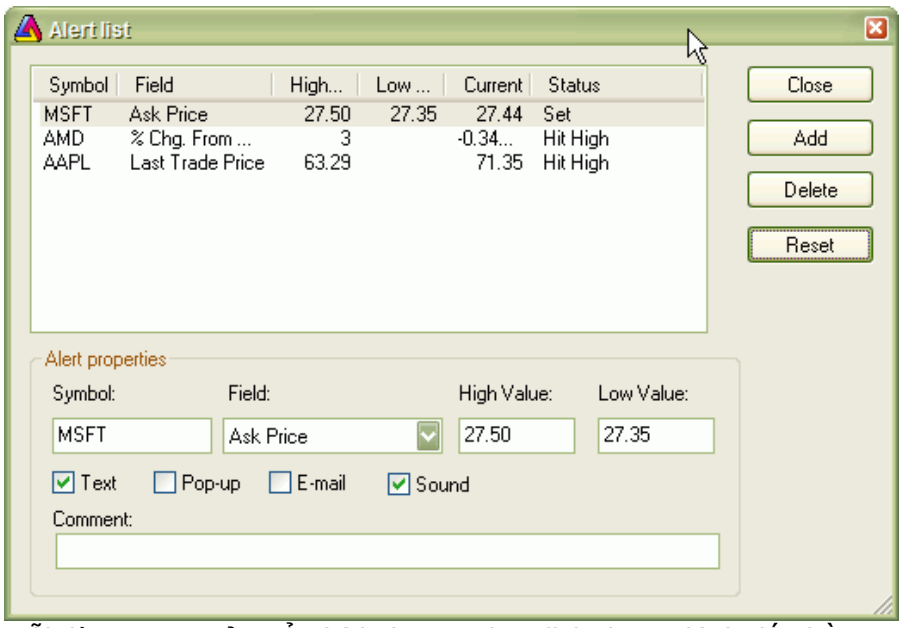
Sound

khi hộp kiểm này được đánh dấu, cảnh báo kích hoạt sẽ phát âm thanh được định nghĩa trong Preferences/Alerts.

Cửa sổ Thời Gian & Giao Dịch

LƯU Ý: Phiên bản Tiêu chuẩn giới hạn chỉ một cửa sổ Thời gian & Giao dịch, trong khi Phiên bản Chuyên nghiệp cho phép mở số lượng không giới hạn cửa sổ Thời gian & Giao dịch đồng thời.

Cửa sổ Thời gian & Giao dịch cung cấp thông tin về mọi giá chào mua, giá chào bán và giao dịch streaming từ thị trường. Mỗi dòng hiển thị đại diện cho một giao dịch mới, một giá chào mua mới hoặc một giá chào bán mới được gửi bởi nguồn dữ liệu streaming.



Mỗi dòng trong cửa sổ Thời gian & Giao dịch được đánh dấu bằng màu sắc để dễ phân biệt giữa các điều kiện khác nhau.

Quy tắc phân màu như sau:

- Nền xanh nhạt nghĩa là GIÁ CHÀO BÁN MỚI

- Nền đỏ nhạt nghĩa là GIÁ CHÀO MUA MỚI.

- Nền trắng nghĩa là GIAO DỊCH MỚI.

- Giá giao dịch cuối đỏ nghĩa là giao dịch xảy ra ở mức giá chào mua hiện tại hoặc thấp hơn.

- Văn bản xanh cho giá/chỉ số chào mua/chào bán nghĩa là giá trị lớn hơn giá trị trước đó.

- Giá giao dịch cuối xanh nghĩa là giao dịch xảy ra ở mức giá chào bán hiện tại hoặc cao hơn.

- Văn bản đen cho giá/chỉ số chào mua/chào bán/khối lượng nghĩa là giá trị không thay đổi.

- Văn bản đỏ cho giá/chỉ số chào mua/chào bán nghĩa là giá trị thấp hơn giá trị trước đó của cùng loại (ví dụ, giá chào mua được viết bằng chữ đỏ có nghĩa là giá chào mua mới thấp hơn giá chào mua trước đó; trường khối lượng xanh có nghĩa là khối lượng giao dịch cuối

cùng hoặc kích thước chào mua/chào bán lớn hơn khối lượng giao dịch trước đó hoặc kích thước chào mua/chào bán).

- Giá giao dịch cuối màu đen nghĩa là giao dịch xảy ra trong khoảng giá chào mua-chào bán hiện tại (lớn hơn giá chào mua và nhỏ hơn giá chào bán).

Cửa sổ Thời gian & Giao dịch trong phiên bản 5.30 hiển thị một số "thống kê gần đây" liên quan đến giao dịch, bao gồm:

- Số lượng giao dịch và số lượng giao dịch trung bình mỗi giây.

- Số lượng giao dịch và cổ phiếu được giao dịch ở mức giá chào bán hoặc cao hơn.

- Số lượng giao dịch và cổ phiếu được giao dịch ở mức giá chào mua hoặc thấp hơn.

- Chênh lệch giữa giá chào bán và giá chào mua được thể hiện bằng số lượng giao dịch và cổ phiếu.

- Chênh lệch giữa giá chào bán và giá chào mua được thể hiện dưới dạng tỷ lệ phần trăm so với tổng số giao dịch/tổng khối lượng giao dịch.

Một chút lý thuyết:

Ask trừ bid: các số dương đại diện cho nhiều giao dịch diễn ra ở phía ASK hơn so với phía BID. Theo lý thuyết, điều này có thể có nghĩa là có nhiều mua hơn là bán, nhưng thực tế điều này phụ thuộc nhiều vào loại chứng khoán đang giao dịch. Đặc biệt, các hồ chứa thanh khoản tối không hiển thị trong sổ lệnh và có thể báo cáo giao dịch cho giấy dán vài giây sau đó do đó làm mất hiệu lực mối quan hệ giữa bid/ask và giá giao dịch thực tế.

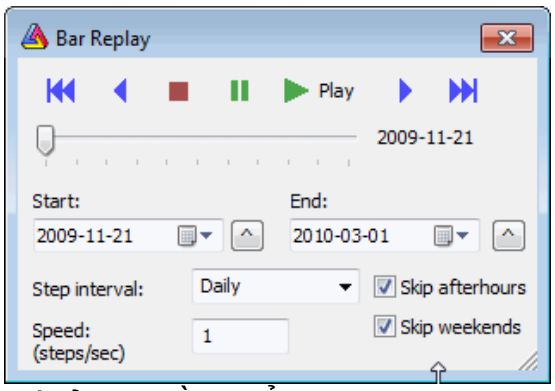
Quan trọng:

Đây là những thông số ngắn hạn, tạm thời - chúng chỉ bao gồm các giao dịch được hiển thị trong cửa sổ T&S kể từ khi mở cửa sổ HOẶC thiết lập lại thống kê.

Bạn có thể thiết lập lại số liệu thống kê bằng cách sử dụng menu chuột phải: "Reset Stats"

Cửa sổ Phát Lại Thanh (Bar Replay)

Công cụ Phát Lại Thanh có sẵn từ menu Công cụ -> Phát Lại Thanh. Tính năng **Phát Lại Thanh cho phép phát lại dữ liệu cho tất cả các ký hiệu cùng một lúc** với tốc độ do người dùng xác định. Điều này có nghĩa là dữ liệu cho tất cả các ký hiệu sẽ kết thúc tại "vị trí phát lại" đã chỉ định. Điều này ảnh hưởng đến tất cả các công thức (bất kể chúng được sử dụng trong biểu đồ (chỉ báo) hay phân tích tự động).



Mô tả các điều khiển

Thanh điều hướng:

- **Rewind** to the beginning: Quay lại đầu.

- One Step Back: **Lùi lại** một bước.

- **Stop**: **Tắt** phát lại thanh (biểu đồ không bị ảnh hưởng bởi phát lại).

- Pause: **Tạm dừng** phát lại hiện tại hoặc vào chế độ tạm dừng cho phép kéo thanh trượt để điều chỉnh hiển thị biểu đồ - trong chế độ PAUSE, dữ liệu được sửa đổi nội bộ nên các báo giá sau vị trí "phát lại" đã chọn sẽ không hiển thị trong bất kỳ phần nào của AmiBroker (biểu đồ/phân tích tự động), ngoại trừ trình chỉnh sửa báo giá.

- Play: Phát lại lịch sử thanh.

- One Step Forward: **Tiến** một bước.

- Forward to end of selected range: **Tiến** đến cuối khoảng đã chọn.

Thanh trượt: Cho phép xem tiến trình phát lại cũng như di chuyển thủ công qua lại bằng cách kéo đầu thanh trượt.

Điều khiển Bắt Đầu/Kết Thúc: Cung cấp các ngày mô phỏng bắt đầu và kết thúc. Phát lại hoạt động sao cho tất cả dữ liệu đến vị trí "Phát lại" hiện tại đều được hiển thị. Dữ liệu sau vị trí này sẽ không hiển thị. "Vị trí phát lại" có thể thay đổi từ ngày "Bắt Đầu" đến ngày "Kết Thúc" do người dùng xác định. Các nút nhỏ ^ ở bên phải các trường ngày Bắt Đầu/Kết Thúc cho phép thiết lập ngày Bắt Đầu/Kết Thúc thành ngày hiện tại đã chọn trên biểu đồ.

Khoảng thời gian Bước: Xác định khoảng thời gian của mỗi bước. Thiết lập khuyến nghị là khoảng thời gian cơ sở của cơ sở dữ liệu của bạn. Nếu bạn có cơ sở dữ liệu 1 phút, khoảng thời gian bước nên là 1 phút. Nếu bạn có cơ sở dữ liệu EOD, khoảng thời gian

bước nên là hàng ngày, tuy nhiên có thể chọn khoảng thời gian bước lớn hơn. Lưu ý rằng khoảng thời gian xem biểu đồ độc lập với điều này. Bạn có thể phát lại cơ sở dữ liệu 1 phút và xem các thanh 15 phút (chúng sẽ giống như thực tế - xây dựng thanh "ma" cuối cùng khi dữ liệu mới đến).

Tham số Tốc độ: Xác định tần suất bước. Điều này có nghĩa là số lượng bước sẽ được phát lại trong một giây. Mặc định là 1. Tối đa là 5 và tối thiểu là 0.1. Nếu bạn chọn 3, ví dụ, AmiBroker sẽ phát một bước mỗi 0.333 giây, tổng cộng là 3 bước mỗi giây

Bỏ qua sau giờ làm việc - khi bật, phần phát lại sẽ bỏ qua những giờ ngoài giờ giao dịch thông thường được xác định trong File->Database Settings->Intraday Settings

Bỏ qua cuối tuần - khi bật, phần phát lại sẽ bỏ qua thứ Bảy và Chủ Nhật

Cách sử dụng

Để vào chế độ phát lại - nhấn nút PLAY hoặc PAUSE - sau đó dữ liệu sẽ bị cắt xén tại vị trí "phát lại" hiện tại.

Để thoát chế độ phát lại - nhấn nút STOP hoặc đóng hộp thoại Bar Replay - toàn bộ tập dữ liệu sẽ được phục hồi.

Lưu ý rằng mô phỏng phát lại được thực hiện nội bộ và cơ sở dữ liệu thực tế được giữ nguyên vẹn (tất cả dữ liệu vẫn còn hiển thị trong Quote Editor), vì vậy không có rủi ro khi sử dụng Bar Reply.

Trình Soạn Thảo Công Thức

AFL Formula Editor ra mắt các tính năng mới:

- Nổi bật cú pháp (cải tiến trong phiên bản 5.80)

- Tự động ghép nối/khớp ngoặc (MỚI trong 5.80)

- Tự động thụt vào (MỚI trong 5.80)

- Có dấu thụt vào (MỚI trong 5.80)

- Tự động hoàn thiện với hai chế độ (tức thì (MỚI trong 5.80) và theo yêu cầu)

- Thông tin gợi ý cho tham số

- Đánh dấu số dòng và biên lựa chọn (MỚI trong 5.80)

- Gấp đoạn mã (MỚI trong 5.80)

- Báo cáo lỗi trực tiếp trong dòng (MỚI trong 5.80)

Giao diện người dùng mới với các tab, có khả năng làm việc trong cả chế độ MDI và chế độ khung nổi độc lập, có thể được di chuyển sau màn hình chính của AmiBroker và đưa trở lại (Window->Toggle Frame) (MỚI trong 5.80) hoặc giữ ở đỉnh (Window->Keep on top)

- Sao chép/xóa khối chữ vuông (Sử dụng chuột và giữ phím Alt trái để đánh dấu khối vuông) (Mới trong 5.80)

- Tự động viết hoa (thay đổi trường hợp)

- Không gian ảo (Mới trong 5.80)

- In nâng cao (với nổi bật cú pháp và tiêu đề/chân trang)

- Đoạn mã code mẫu (Mới trong 5.80)

- Trình Gỡ lỗi Tích hợp (MỚI trong 6.10)

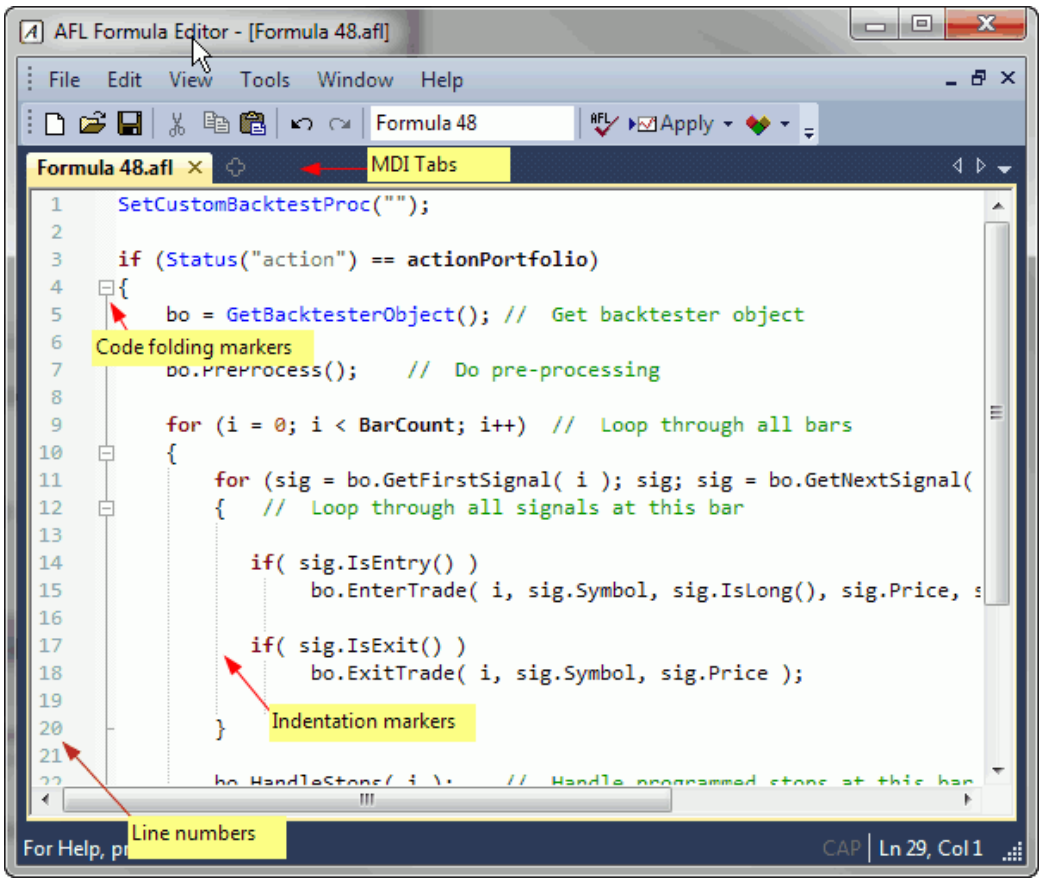
- Đánh dấu trang (MỚI trong 6.10)

- Tìm trong tập tin (MỚI trong 6.10)

- Bình luận/tiêu chuẩn khối (MỚI trong 6.10)

- Liên kết có thể nhấp vào trong bình luận (MỚI trong 6.10)

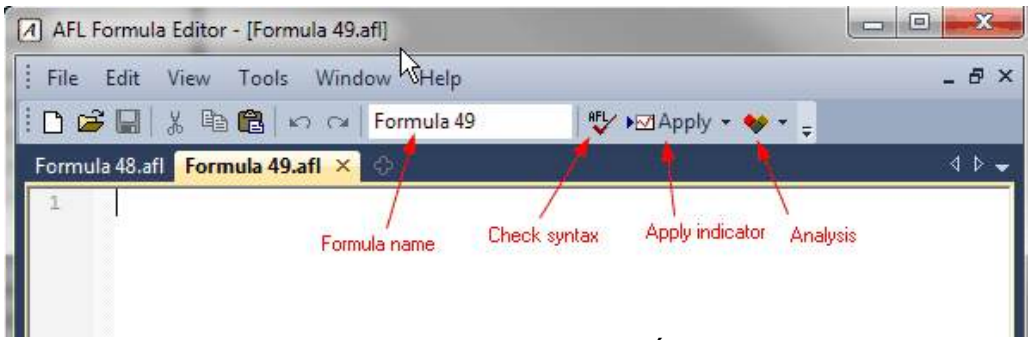
Những tính năng này đã đơn giản hóa việc viết công thức và cung cấp sự trợ giúp tức thì nên thời gian cần thiết để viết công thức giảm đáng kể.



Menu

Các tùy chọn menu của Trình Biên Tập Công Thức được mô tả chi tiết trong chương Menus: Trình Biên Tập Công Thức của hướng dẫn.

Thanh Công Cụ



Thanh công cụ Trình Biên Tập Công Thức cung cấp các nút sau:

Mới: Xóa nội dung trong cửa sổ trình biên tập công thức.

Mở: Mở tệp công thức.

Lưu: Lưu công thức với tên hiện tại.

In: In công thức.

Cắt: Cắt phần đã chọn và sao chép vào clipboard.

Sao chép: Sao chép phần đã chọn vào clipboard.

Dán: Dán nội dung hiện tại trong clipboard tại vị trí con trỏ hiện tại

Undo - Hoàn tác hành động gần đây (đa cấp)

Redo - Thực hiện lại hành động gần đây (đa cấp)

Tên công thức - là một trường EDIT cho phép bạn thay đổi tên của tập tin công thức, khi bạn thay đổi tên ở đây và nhấn nút **Save**, công thức sẽ được lưu lại dưới tên mới và thay đổi này sẽ được phản ánh trong thanh tiêu đề của trình soạn thảo (CAPTION BAR) cũng như trong THANH TRẠNG THÁI (STATUS BAR- Thanh trạng thái hiển thị đường dẫn đầy đủ).

Check syntax - Kiểm tra công thức hiện tại có lỗi hay không

Apply indicator - Lưu và áp dụng công thức hiện tại như một biểu đồ / chỉ báo MỘT LẦN

Phân tích - Lưu công thức và chọn nó làm công thức hiện tại trong cửa sổ Phân tích tự động và lặp lại hành động Phân tích được sử dụng gần đây nhất (ví dụ: quét hoặc khám phá hoặc Backtest hoặc tối ưu hóa)

Cách sử dụng

Cách sử dụng thông thường của Formula Editor như sau:

Mở Formula Editor

Nhập công thức

Nhập tên có ý nghĩa mô tả mục đích của mã của bạn vào trường **Formula Name**

Nhấn nút **Apply indicator** (nếu bạn đã viết mã chỉ báo)

.. hoặc..

nhấn nút **Analysis** để hiện cửa sổ Phân tích Tự động (khi bạn đã viết mã khám phá/quét hoặc hệ thống giao dịch)

Highlight Cú Pháp

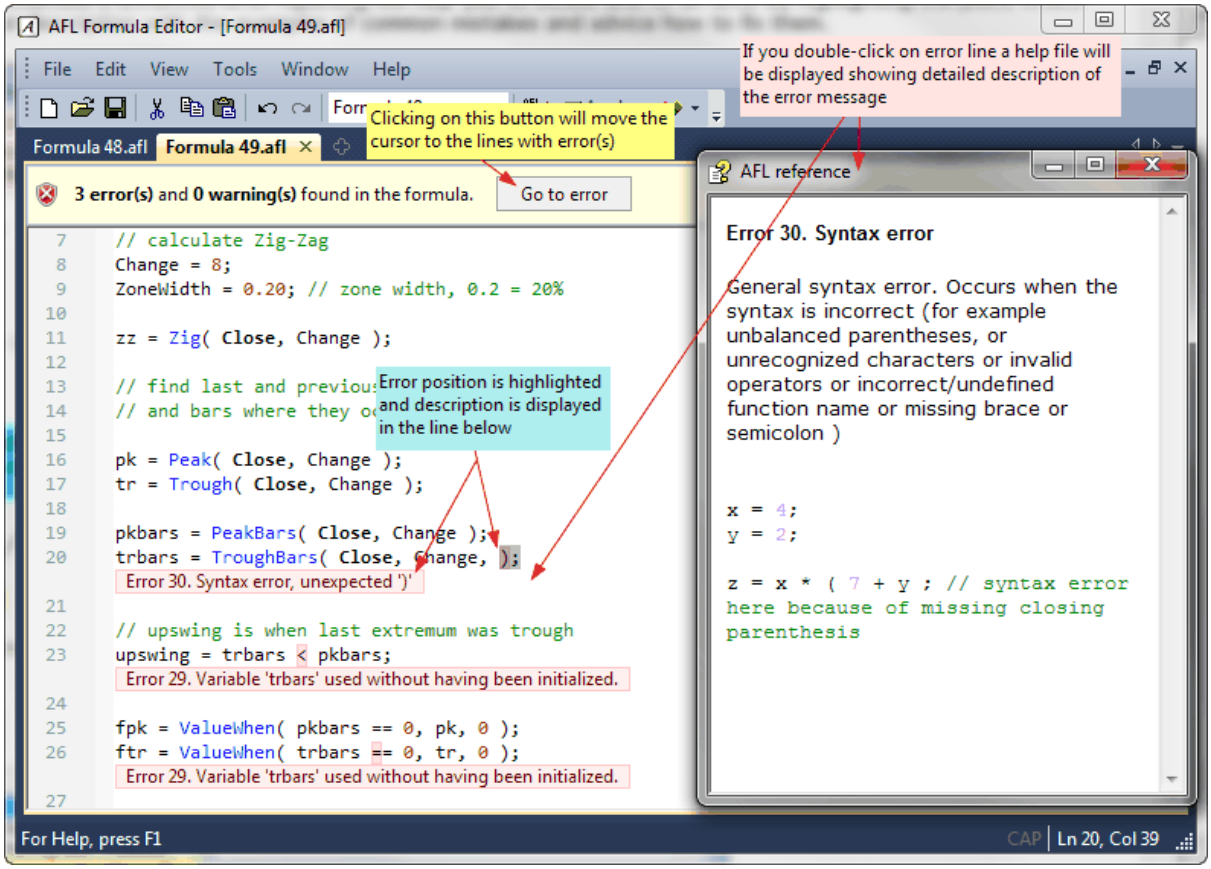
Trình soạn thảo AFL của AmiBroker có tính năng đánh dấu cú pháp có thể tùy chỉnh của người dùng, tự động áp dụng các màu sắc và kiểu do người dùng xác định cho các yếu tố ngôn ngữ khác nhau như hàm và tên biến đặt trước, chuỗi, số, bình luận, v.v. Tính năng này đơn giản hóa việc viết mã. Bạn có thể sửa đổi lược đồ màu sắc trong cửa sổ Preferences.

Báo cáo lỗi nâng cao

Khi bạn mắc một lỗi trong công thức của mình, hệ thống báo cáo lỗi nâng cao của AmiBroker sẽ giúp bạn xác định và sửa lỗi bằng cách đánh dấu vị trí lỗi xảy ra và hiển thị

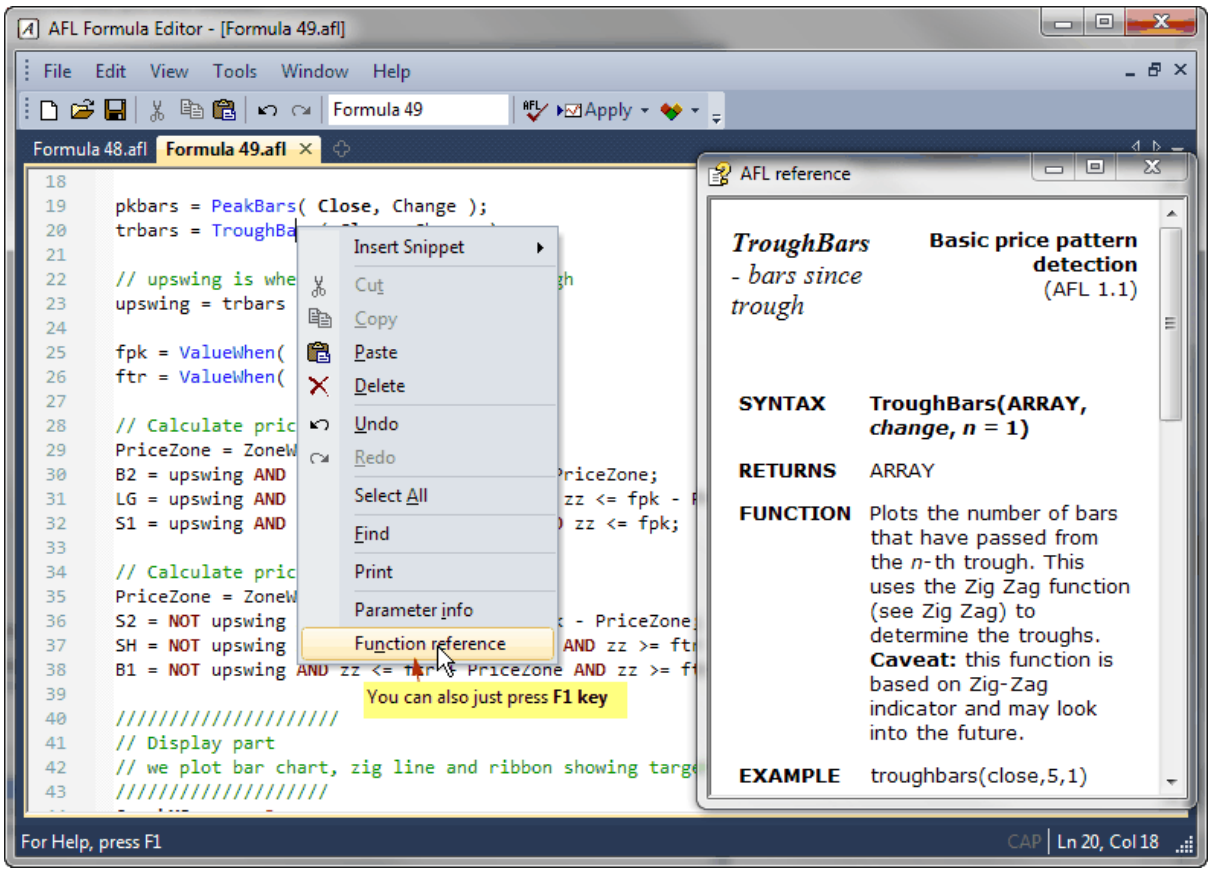
mô tả lỗi mở rộng với các ví dụ về lỗi phổ biến và lời khuyên về cách khắc phục chúng. Trong phiên bản 5.80, mô tả về lỗi được hiển thị trực tiếp với mã.

Một thanh tin nhắn hiển thị tổng số lỗi và / hoặc cảnh báo. Nếu bạn nhấn nút "Go to error", trình chỉnh sửa sẽ di chuyển con trỏ đến dòng có lỗi, nếu bạn nhấn nó lần nữa, nó sẽ di chuyển đến lỗi tiếp theo và cứ tiếp tục như vậy. Nếu bạn đóng thanh thông báo bằng nút "X", tất cả các thông báo lỗi sẽ bị xóa (ẩn) khỏi màn hình. Bạn có thể sử dụng menu Chỉnh sửa -> Xóa Thông Báo Lỗi (Ctrl+E) để xóa thông báo lỗi riêng lẻ (trong dòng hiện tại).



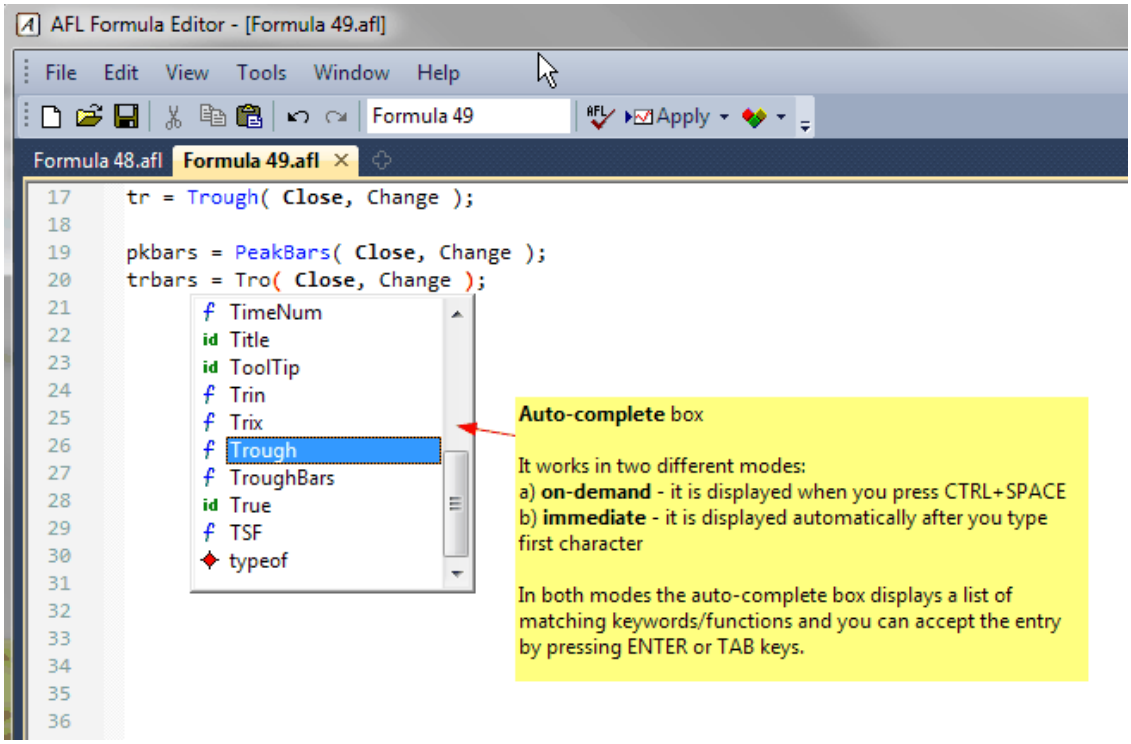
Trợ Giúp Theo Ngữ Cảnh

Bạn có thể nhanh chóng hiển thị trang tham khảo chức năng AFL liên quan nếu bạn nhấn phím F1 hoặc chọn "Tham khảo chức năng" từ menu ngữ cảnh trong khi con trỏ đang ở bên trong hoặc ngay sau tên hàm như hình dưới đây:



Tự động hoàn thành câu lệnh

Tính năng hoàn thành tự động (có sẵn khi bạn nhấn tổ hợp phím CTRL+SPACE) sẽ hoàn tất việc nhập các hàm và biến dự trữ cho bạn, hoặc hiển thị danh sách các ứng viên nếu những gì bạn đã nhập có hơn một kết quả phù hợp. Bạn có thể chọn mục trong danh sách bằng cách sử dụng phím mũi tên lên/xuống hoặc chuột. Để chấp nhận lựa chọn, nhấn phím RETURN (ENTER). Bạn cũng có thể ngay lập tức nhấn phím cách (đối với biến) hoặc dấu ngoặc mở (đối với hàm), và AmiBroker sẽ tự động hoàn thành từ đang chọn và đóng danh sách. Để thoát khỏi danh sách, nhấn phím ESC.



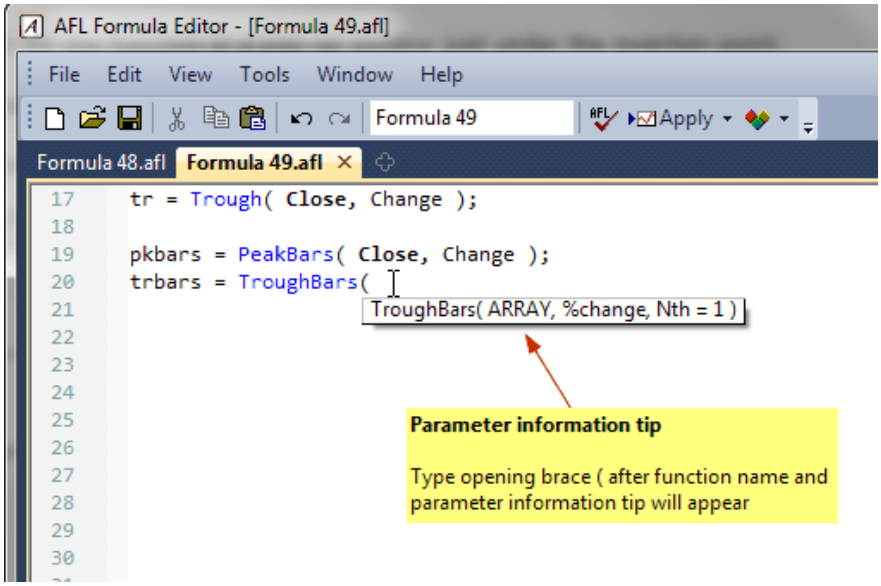
Thông Tin Tham Số

Khi bạn đang nhập một hàm, bạn có thể hiển thị một Tool Tip chứa đầy đủ cấu trúc hàm, bao gồm các tham số. Tool Tip **Thông Tin Tham Số** cũng được hiển thị cho các hàm lồng nhau.Khi con trỏ của bạn ở bên cạnh một hàm, hãy nhập dấu ngoặc đơn mở như thường lệ để bao gồm danh sách tham số.

AmiBroker sẽ hiển thị đầy đủ khai báo cho hàm trong một cửa sổ pop-up ngay dưới vị trí con trỏ.

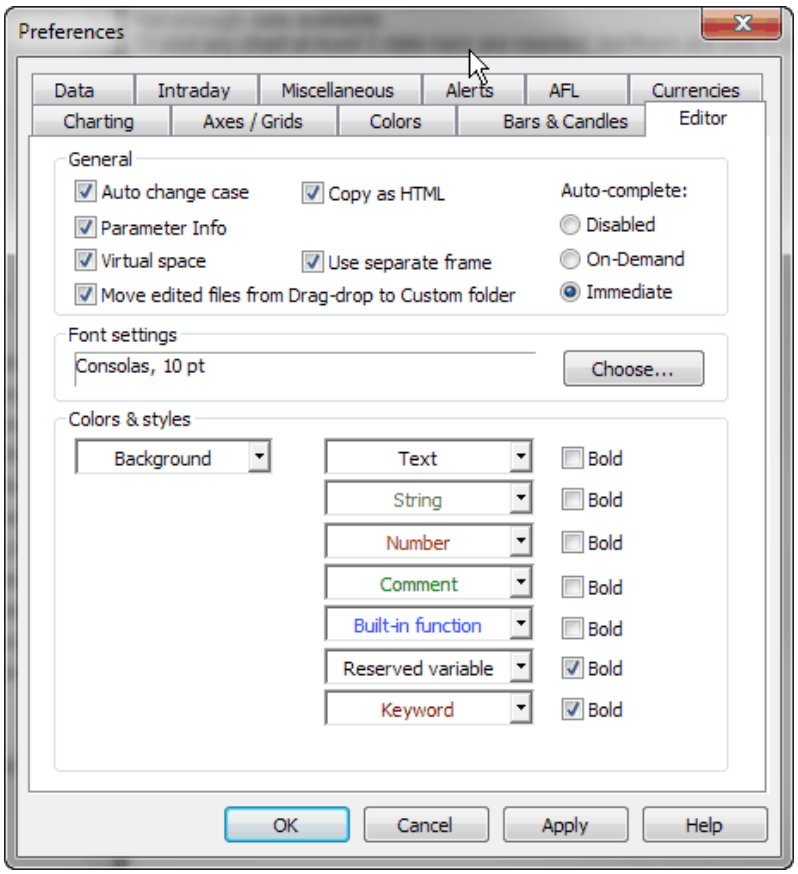
Việc nhập dấu ngoặc đơn đóng sẽ ẩn danh sách tham số.

Bạn cũng có thể ẩn danh sách này bằng cách nhấn phím mũi tên lên/xuống, nhấp chuột hoặc nhấn phím RETURN.



Cấu Hình Trình Biên Tập

Các cài đặt của trình biên tập AFL có thể được thay đổi thông qua menu Công cụ -> Tùy chọn, trên trang Trình Biên Tập:



Tự Động Thay Đổi Chữ Hoa/Chữ Thường - Tùy chọn này kiểm soát việc trình biên tập tự động thay đổi chữ hoa của các từ khóa dự trữ. Ví dụ, nếu người dùng nhập "valuewhen", nó sẽ tự động chuyển thành "ValueWhen".

Thông Tin Tham Số - Tùy chọn này kiểm soát việc hiển thị các Tool Tip thông tin tham số. Khi được bật, thông tin chi tiết về tham số của hàm sẽ xuất hiện khi bạn nhập hàm trong trình biên tập.

Không Gian Ảo - Tùy chọn này kiểm soát việc có thể đặt con trỏ tự do ở bất kỳ vị trí nào sau khi kết thúc một dòng.

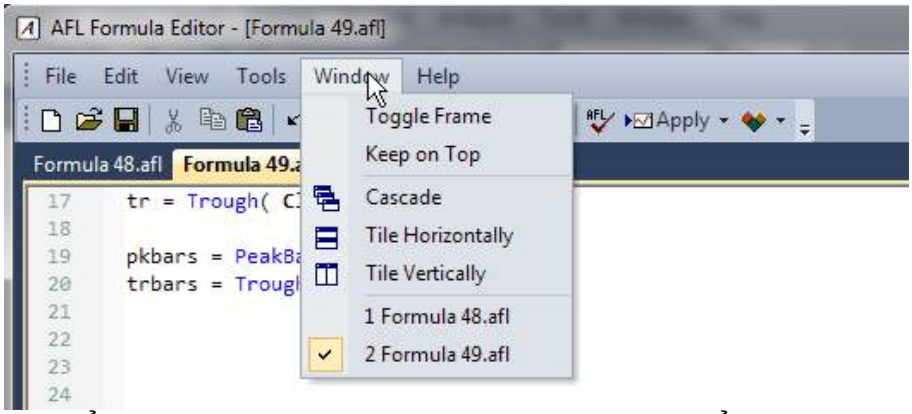
Chuyển Tệp Đã Chỉnh Sửa từ Kéo-Thả đến Thư Mục Tùy Chọn - Thông thường, các công thức được tạo bằng cơ chế kéo-thả sẽ nằm trong thư mục kéo-thả ẩn. Nếu bạn muốn chỉnh sửa chúng, bạn có thể giữ nguyên trong thư mục kéo-thả (ẩn) hoặc chọn tự động chuyển chúng đến thư mục "tùy chọn". Tùy chọn này cho phép di chuyển tự động đến thư mục tùy chọn.

Sao Chép Dưới Dạng HTML - Cho phép sao chép ở định dạng HTML, do đó mã AFL sẽ được sao chép kèm theo màu sắc. Nếu không, nó sẽ được sao chép dưới dạng văn bản thuần túy mà không có định dạng.

Sử Dụng Khung Riêng - Nếu bật, nó sẽ hiển thị Trình Biên Tập AFL trong một khung hoàn toàn riêng biệt, hoạt động như một ứng dụng riêng. Nếu tắt, trình biên tập sẽ được hiển thị dưới dạng tab MDI trong khung chính của AmiBroker (cùng với biểu đồ, cửa sổ phân tích, web, cửa sổ tài khoản, v.v.). Theo mặc định, tùy chọn này được bật.

Tự Động Hoàn Thành - Trong chế độ "Theo yêu cầu", danh sách hoàn thành tự động chỉ xuất hiện khi bạn nhấn Ctrl+SPACE. Trong chế độ "Ngay lập tức", danh sách hoàn thành tự động sẽ xuất hiện ngay khi bạn nhập ký tự đầu tiên (chữ cái) của định danh.

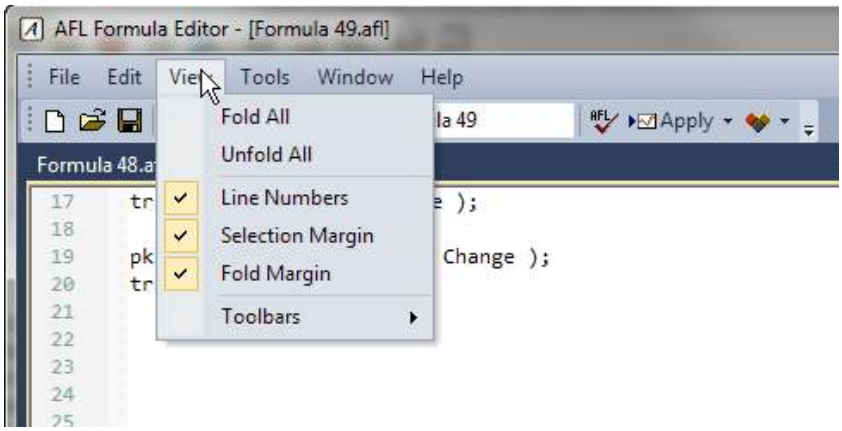
Điều Khiển Cửa Sổ



Cửa sổ Trình Biên Tập AFL như một khung riêng có thể được đưa lên trên hoặc lùi xuống như bất kỳ cửa sổ ứng dụng nào khác sử dụng Thanh tác vụ Windows. Ngoài ra, có menu **Cửa Sổ -> Chuyển Khung** (và phím tắt **Ctrl+**`, ` là phím dấu ngã nằm ngay trên phím TAB trên hầu hết bàn phím) cho phép nhanh chóng chuyển đổi giữa khung chính của AmiBroker và khung trình biên tập AFL.

Người dùng cũng có thể bật tính năng **Cửa Sổ -> Giữ Trên Cùng** để giữ cửa sổ trình biên tập trên cùng của khung chính AmiBroker.

Lề



Lề cho số dòng, lựa chọn lề và lề gấp có thể được bật/tắt bằng cách sử dụng menu **View.** Trong menu này cũng có các tùy chọn để gấp/mở tất cả mã.

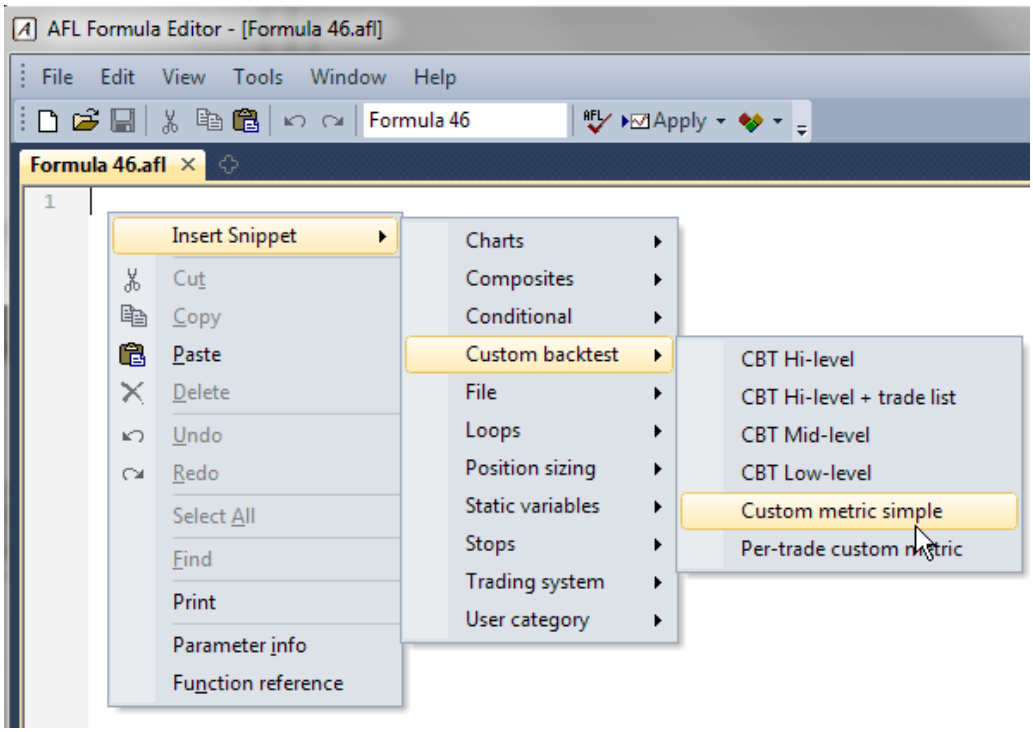
Mảnh Mã

Mảnh mã là một phần nhỏ của mã AFL. Nó có thể được chèn bằng cách:

- nhấp chuột phải vào cửa sổ soạn thảo AFL và chọn menu "Insert Snippet", hoặc

- kéo một mảnh mã từ cửa sổ Code Snippet, hoặc

- gõ kích hoạt bàn phím (như @for) trong trình soạn thảo



Bookmarks

Bookmark là một dấu mà cho phép nhanh chóng nhảy đến dòng được đánh dấu trong công thức.

Để đặt/xoá dấu trang (bookmark), hãy sử dụng menu **Edit->Bookmark->Insert /Remove** hoặc phím **Ctrl+F2.** Khi bạn đặt dấu trang, một dấu đánh dấu màu xanh nhạt sẽ xuất hiện ở lề bên trái. Giờ đây, với một vài dấu trang đã được đặt, bạn có thể di chuyển nhanh (nhảy) giữa chúng bằng phím **F2** (đi đến dấu trang tiếp theo), **Shift+F2 key** (đi đến dấu trang trước). Bạn có thể xoá tất cả các dấu trang bằng cách sử dụng **Edit->Bookmark->Clear All**. Hãy lưu ý rằng dấu trang được lưu lưu cùng với thông tin gỡ lỗi (breakpoints/watches) trong một tệp riêng có phần mở rộng là .dbg.

Tìm Kiếm Trong Các Tệp

Chức năng Tìm kiếm trong các tệp có sẵn từ menu Edit->Find in Files và cho phép bạn tìm kiếm tất cả các tệp trong các thư mục đã chọn cho văn bản được chỉ định. Kết quả tìm kiếm được trình bày trong cửa sổ Output (**Window->Output menu)** như sau:

Filepath\Filename.afl(line_number): Line contents.

Giờ đây, nếu bạn nhấp đúp vào dòng output đó, AmiBroker sẽ tự động mở tệp đã cho trong tab biên tập mới.

Nhận Xét Khối/Bỏ Nhận Xét Khối

Trình soạn thảo AFL giờ đây cung cấp một công cụ để tự động ghi chú ra các khối các dòng đã chọn. Công cụ này thêm một // (gạch chéo kép) vào mỗi dòng trong khu vực đã chọn. Để sử dụng công cụ này, chọn một số dòng bằng chuột hoặc con trỏ + phím SHIFT, sau đó sử

dụng **Edit->Line Comment.** Tất cả các dòng trong phạm vi đã chọn sẽ được ghi chú với // (gạch chéo kép). Để bỏ ghi chú, chọn lại phạm vi và chọn **Edit->Line Comment** một lần nữa.

Liên Kết Có Thể Nhấp Chuột Trong Các Nhận Xét

Trình soạn thảo AFL giờ đây hỗ trợ liên kết có thể nhấp chuột theo kiểu JavaDoc / Doxygen trong các nhận xét tài liệu. Để sử dụng liên kết trong các nhận xét, bạn phải đặt lệnh @link kèm theo đường dẫn tới tệp hoặc URL bên trong nhận xét kiểu JavaDoc/Doxygen:

Nhận xét đa dòng bắt đầu với / giống như:

/

@link readme.html

@link http://www.amibroker.com

*/

(lưu ý dấu sao kép sau dấu gạch chéo đầu tiên)

Nhận xét dòng đơn bắt đầu bằng gạch chéo gấp ba lần

/// @link readme.html

/// @link c:\program files\amibroker\readme.html

/// @link http://www.amibroker.com

Bây giờ, khi bạn di chuột qua lệnh **@link**, bạn sẽ thấy gạch chân cho thấy nó có thể nhấp được. Lệnh này phản ứng với việc nhấp đúp (không phải nhấp đơn). Khi bạn nhấp đúp vào nó, tài liệu liên kết sẽ được mở.

Lệnh **@link** có thể mở các trang web, tệp cục bộ (cả đường dẫn tương đối và tuyệt đối đều được hỗ trợ) với chương trình đã đăng ký trong Windows để mở loại tệp đó. Vì vậy, nếu bạn sử dụng:

/// @link something.doc

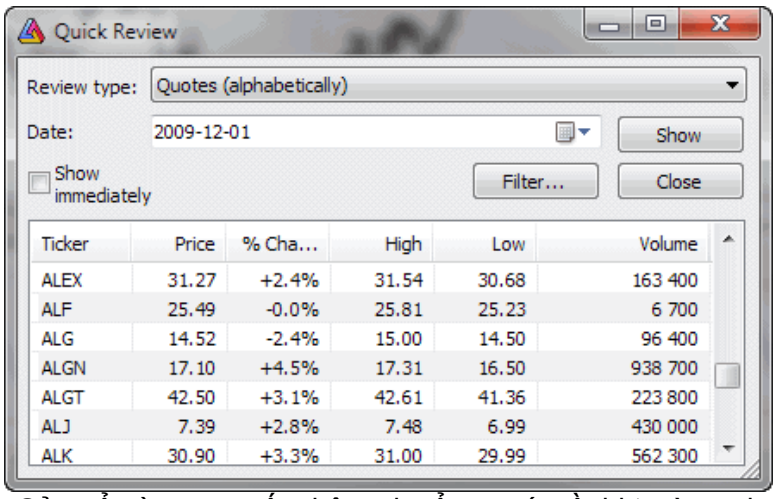
Sau đây là MS Word sẽ được sử dụng.

/// @link test.xls

sẽ mở file test.xls bằng Excel.

Đường dẫn tương đối liên quan đến thư mục hoạt động của AmiBroker. Các file Html sẽ được mở bằng trình duyệt mặc định, các file txt thường được mở bằng Notepad (hoặc bất kì ứng dụng nào bạn sử dụng). Nếu file không tồn tại thì bạn sẽ nhận được một thông báo lỗi.

Cửa Sổ Đánh Giá Nhanh



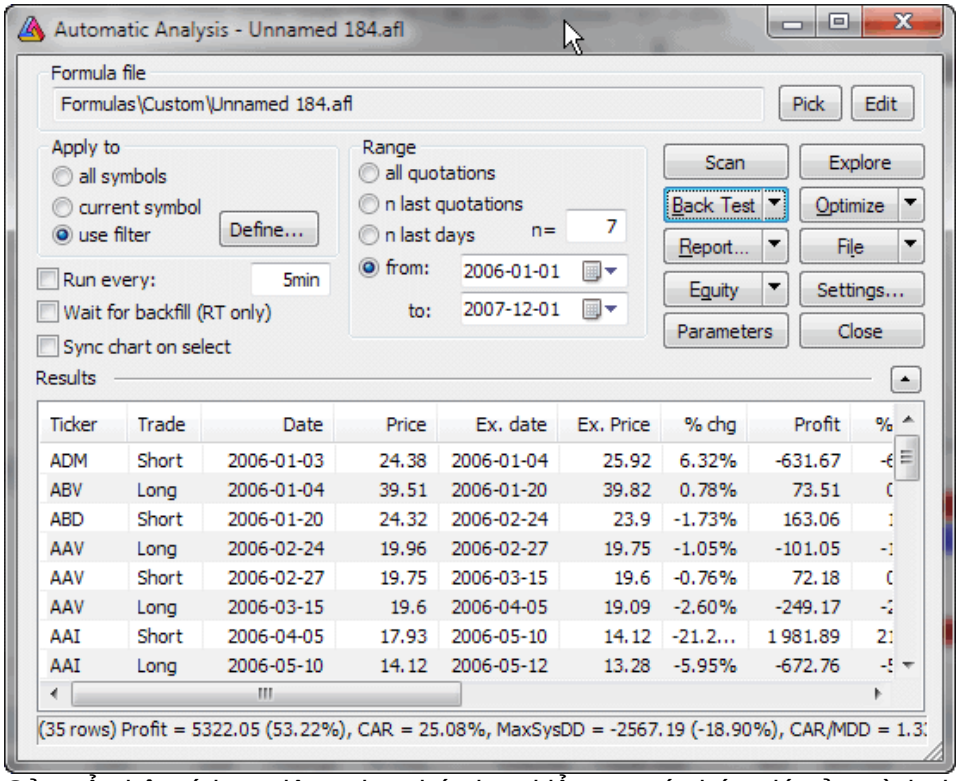
Cửa sổ này cung cấp thông tin tổng quát về thị trường như:

- Báo giá ký hiệu hàng ngày
- Bảng so sánh lợi nhuận hàng tuần
- Bảng so sánh lợi nhuận hàng tháng
- Bảng so sánh lợi nhuận hàng quý
- Bảng so sánh lợi nhuận hàng năm
- So sánh Giá/Lợi nhuận
- So sánh Giá/Giá trị sổ sách

Trong trường **Ngày**, bạn có thể chọn ngày cơ sở để thực hiện các so sánh. Ví dụ, lợi nhuận hàng tuần được tính bằng cách chia giá đóng cửa của ngày cơ sở với giá đóng cửa của một tuần trước đó.

Nút Lọc cho phép bạn thu hẹp tìm kiếm đến các ký hiệu được định nghĩa trong cửa sổ cài đặt Lọc.

Cửa Sổ Phân Tích Tự Động



Cửa sổ phân tích tự động cho phép bạn kiểm tra các báo giá của mình dựa trên các quy tắc mua/bán đã định nghĩa. AmiBroker có thể tạo báo cáo cho bạn biết liệu có tín hiệu mua/bán nào xảy ra trên biểu tượng cụ thể trong khoảng thời gian đã chỉ định hay không. Nó cũng có thể mô phỏng giao dịch, giúp bạn có cái nhìn về hiệu suất của hệ thống của mình.

Ở phần trên cùng của cửa sổ, bạn có thể thấy đường dẫn đến công thức đang được sử dụng cùng với các nút **Chọn** và **Chỉnh sửa.**

Nút **Chọn** mở một hộp thoại cho phép bạn chọn công thức mà bạn muốn sử dụng cho phân tích.

Nút **Chỉnh sửa** mở Trình chỉnh sửa Công thức AFL, cho phép bạn chỉnh sửa công thức hiện đang được chọn

Nếu bạn muốn tạo công thức mới, chỉ cần mở Trình chỉnh sửa Công thức trực tiếp từ menu **Công cụ -> Trình chỉnh sửa Công thức**, nhập công thức và nhấn nút **Phân tích** trên thanh công cụ Trình chỉnh sửa Công thức.

Trong trình chỉnh sửa công thức, bạn cần viết mã để xác định hoặc là quét/khám phá mà bạn muốn thực hiện, hoặc là một hệ thống giao dịch mà bạn muốn kiểm tra lại. Bạn có thể tìm thấy mô tả về ngôn ngữ này trong hướng dẫn tham chiếu AFL.

Để mọi thứ hoạt động, bạn nên viết hai câu lệnh gán (một cho quy tắc mua, một cho quy tắc bán), ví dụ:

```
buy = cross(macd(), 0);
```

```
sell = cross(0, macd());
```

Dưới các trường này có một số điều khiển để thiết lập:

1. Biểu tượng nào sẽ được áp dụng phân tích. Bạn có thể chọn tất cả các biểu tượng, chỉ biểu tượng hiện đang được chọn (được chọn trong Cửa sổ Chọn) hoặc… cài đặt bộ lọc tùy chỉnh

2. Thời gian phân tích

phân tích có thể được áp dụng cho tất cả các báo giá có sẵn hoặc chỉ cho một số lượng báo giá gần đây nhất (hoặc những ngày) hoặc trong một khoảng thời gian (từ/ngày đến)

Ở phần dưới cửa sổ phân tích, bạn có thể thấy 4 nút:

1. Scan

Nút này bắt đầu chế độ quét tín hiệu - AmiBroker sẽ tìm kiếm trong phạm vi các biểu tượng и báo giá đã định nghĩa để tìm tín hiệu mua/bán do quy tắc giao dịch của bạn xác định. Nếu một trong các điều kiện mua/bán được thỏa mãn, AmiBroker sẽ hiển thị một dòng mô tả khi nào và biểu tượng nào đã phát tín hiệu. Tiếp theo, AmiBroker tiếp tục đến cuối phạm vi để có thể tạo nhiều tín hiệu trên một biểu tượng.

2. Khám phá

Chế độ này bắt đầu khi AmiBroker quét qua cơ sở dữ liệu để tìm các biểu tượng phù hợp với bộ lọc mà người dùng đã định nghĩa. Người dùng có thể xác định các cột đầu ra hiển thị bất kỳ loại thông tin nào cần thiết. Để biết thêm thông tin, vui lòng xem "Hướng dẫn: Cách tạo bộ lọc của riêng bạn".

3. Back Test

Chế độ này bắt đầu chế độ kiểm tra lùi - AmiBroker sẽ tìm kiếm trong khoảng phạm vi các biểu ngữ và báo giá để tìm tín hiệu MUA được xác định bởi quy tắc mua của bạn. Nếu quy tắc mua được thỏa mãn, AmiBroker sẽ "mua" cổ phiếu đang phân tích. Kế đó nó sẽ tìm kiếm tín hiệu BÁN. Sau đó, nếu quy tắc bán được thỏa mãn, AmiBroker sẽ "bán" các biểu tượng đã mua trước đó. Đồng thời, nó sẽ hiển thị thông tin về giao dịch này trong danh sách xem. Sau khi thực hiện mô phỏng, sẽ hiển thị bản tóm tắt. Đọc thêm trong "Hướng dẫn: Cách kiểm tra hệ thống giao dịch của bạn". Các tham số kiểm tra lùi có thể được thay đổi bằng cửa sổ Cài đặt.

4.**Cài đặt** - cho phép bạn thay đổi cài đặt kiểm tra lùi.

5. **Tối ưu hóa** - cho phép bạn tối ưu hóa hệ thống giao dịch của mình. Đọc thêm trong "Hướng dẫn: Cách tối ưu hóa hệ thống giao dịch của bạn".

6. **Kiểm tra** - tùy chọn này cho phép bạn kiểm tra xem công thức của bạn có tham chiếu đến các báo giá trong tương lai hay không. AmiBroker phân tích công thức của bạn và phát hiện xem nó có sử dụng các báo giá vượt qua thanh hiện tại hay không. Xin lưu ý rằng các công thức tham chiếu đến tương lai có thể mang lại kết quả kiểm tra lại không thực tế mà

không thể tái hiện trong giao dịch thực tế, vì vậy bạn nên tránh các hệ thống tham chiếu đến tương lai.

7. **Báo cáo** - tùy chọn này hiển thị cửa sổ Báo cáo cho phép bạn xem, in và lưu kết quả thử nghiệm.

8. **Vốn** - chỉ có sẵn sau khi kiểm tra lại - hiển thị đường cong Vốn cho biểu tượng hiện đang được chọn trong một khung biểu đồ mới. Hãy kiểm tra "AFL: Biểu đồ vốn và chức năng".

9. **Xuất** - cho phép bạn xuất kết quả ra tệp CSV (các giá trị phân tách bằng dấu phẩy).

10. **Đóng** - tùy chọn này sẽ đóng cửa sổ phân tích.

Ngoài ra, bạn còn có hai tùy chọn "Tải" và "Lưu" để tải và lưu các quy tắc giao dịch từ/tới các tệp.

Mở rộng chế độ xem kết quả trong cửa sổ phân tích tự động

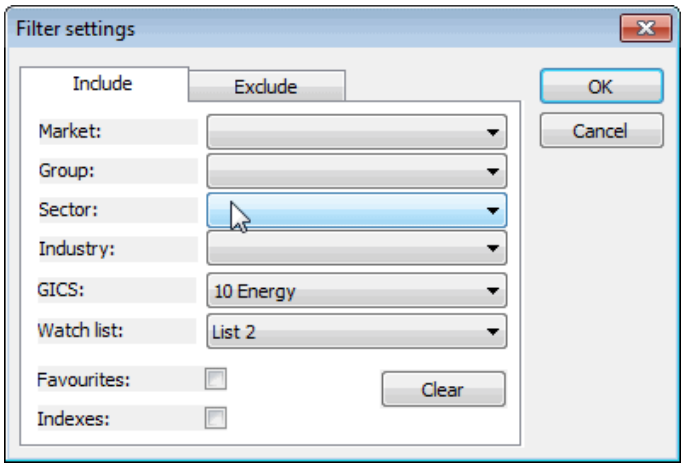
Có một nút mũi tên nhỏ bên cạnh đường phân cách "Danh sách kết quả". Nút này được cung cấp để mở rộng/thu nhỏ danh sách kết quả. Khi bạn đang chỉnh sửa công thức của mình, việc có chế độ xem chỉnh sửa lớn hơn là điều tốt, nhưng để xem các kết quả kiểm tra lại, sẽ thuận tiện hơn nếu mở rộng danh sách kết quả. Trong trường hợp đó, chỉ cần nhấp vào nút đó và danh sách kết quả sẽ được mở rộng (và trường chỉnh sửa sẽ bị thu nhỏ). Để làm ngược lại, chỉ cần nhấp lại vào nút.

Cửa sổ Thiết lập Bộ lọc

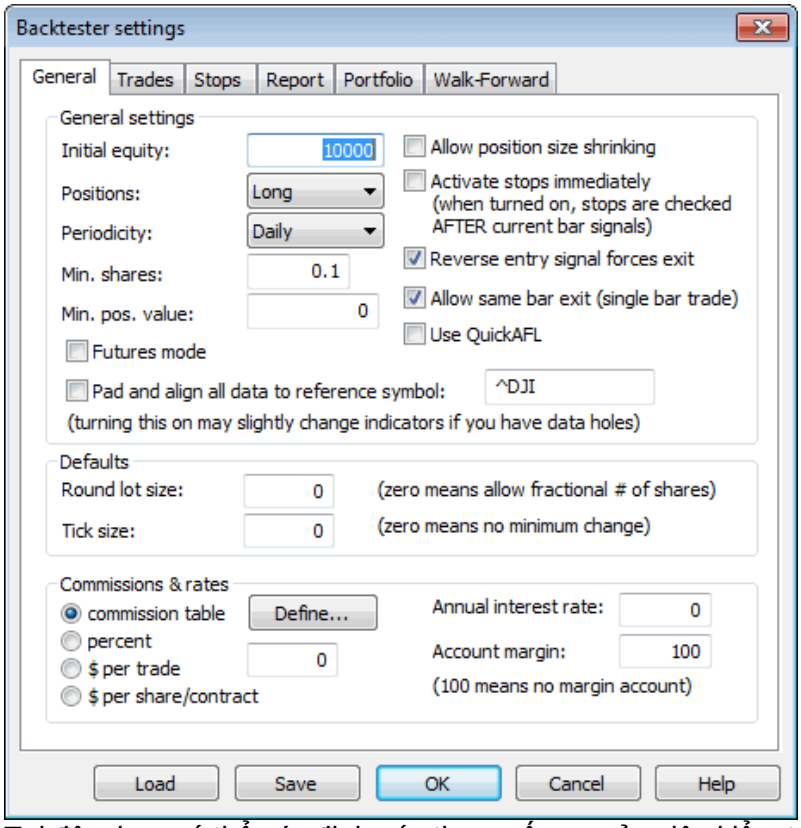
Cửa sổ này có sẵn từ nút "Filter/Define..." trong cửa sổ xem nhanh và phân tích.

Tùy chọn lọc cho phép bạn thu hẹp tìm kiếm đến các biểu tượng thuộc về thị trường, nhóm, ngành và nghành được chỉ định. Bạn cũng có thể đánh dấu để chỉ bao gồm các mục ưa thích hoặc chỉ số. Bạn có thể sử dụng bộ lọc theo kiểu bao gồm và/hoặc loại trừ để bạn cũng có thể loại trừ một cách lựa chọn một số loại biểu tượng.

Nếu bạn sử dụng nhiều hơn một danh mục (ví dụ bạn chọn Thị trường và Ngành) thì bộ lọc chỉ trả kết quả cho những biểu tượng phù hợp với danh mục đầu TIÊN VÀ danh mục thứ hai (điều này là liên kết logic, không phải giả thuyết)



Cửa sổ Cài đặt Kiểm tra Hệ thống



Tại đây, bạn có thể xác định các tham số sau của việc kiểm tra lại:

Tab Chung

Vốn ban đầu - xác định kích cỡ tài khoản của bạn. Trong Portfolio backtest - nó đại diện cho kích cỡ portfolio toàn bộ. Trong backtest "Cá nhân" nó là vốn ban đầu cho mỗi biểu tượng.

Vị trí được coi trọng (dài, ngắn, cả dài và ngắn)

Chế độ Futures

Check box này trong trang cài đặt là chìa khóa để backtesting futures. Nó hướng dẫn backtester sử dụng tiền đặt cọc và giá trị điểm trong các phép tính.

Min. shares

Số lượng cổ phiếu tối thiểu được phép mua/short. Backtester sẽ không thực hiện các giao dịch dưới giới hạn này. Nên là 1 cho cổ phiếu. Các giá trị phân số tốt cho các quỹ tương hỗ.

Min. pos value

Giá trị vị trí tối thiểu (theo tiền tệ gốc) của giao dịch được phép nhập. Backtester sẽ không thực hiện các giao dịch dưới giới hạn này. Zero có nghĩa là không giới hạn.

Đệm và căn chỉnh theo biểu tượng tham chiếu

Khi chức năng này được bật, tất cả các dấu dấu của các biểu tượng sẽ được đệm và căn chỉnh theo biểu tượng tham chiếu. Lưu ý: theo mặc định, cài đặt này đều OFF. Hãy sử dụng một cách có trách nhiệm. Nó có thể làm chậm quá trình backtest/khám phá/quét và đưa ra một số thay đổi nhỏ về giá trị chỉ số khi dữ liệu của bạn có lỗ và lỗ được điền với dữ liệu thanh trước đó. Tính năng này được thiết kế để sử dụng khi hệ thống của bạn sử dụng thời gian đầu tư chung lên thị trường (tạo ra tín hiệu toàn cầu dựa trên dữ liệu và/hoặc chỉ số được tính từ "biểu tượng tham chiếu") hoặc khi bạn tạo tổng hợp từ dữ liệu không căn chỉnh. Lưu ý: nếu biểu tượng tham chiếu không tồn tại, dữ liệu sẽ không được đệm.

Tài khoản bảo lãnh

Cài đặt này xác định yêu cầu bảo lãnh theo phần trăm cho toàn bộ tài khoản. Giá trị mặc định của tài khoản bảo lãnh là 100. Điều này có nghĩa là bạn phải cung cấp 100% số tiền để tham gia vào giao dịch, và đây là cách mà backtester hoạt động trong các phiên bản trước. Nhưng bây giờ bạn có thể mô phỏng tài khoản ký quỹ. Khi bạn mua hàng trên ký quỹ, bạn chỉ đơn giản là mượn tiền từ người môi giới của mình để mua cổ phiếu. Với quy định hiện tại, bạn có thể đưa ra 50% giá trị mua cổ phiếu mà bạn muốn mua và mượn phần còn lại từ người môi giới của bạn. Để mô phỏng điều này hãy nhập 50 vào trường giá trị Ký quỹ Tài khoản (xem hình. 1). Nếu tài sản ban đầu của bạn được đặt là 10000 thì khả năng mua hàng của bạn sẽ là 20000 và bạn sẽ có thể tham gia vào các giao dịch lớn hơn. Xin lưu ý rằng cài đặt này đặt bảo lãnh cho toàn bộ tài khoản và NÓ KHÔNG liên quan đến giao dịch tương lai. Nói cách khác, bạn có thể giao dịch cổ phiếu trên tài khoản ký quỹ.

Lệ phí

Bảng lệ phí - backtester sẽ sử dụng bảng lệ phí như được định nghĩa trong cửa sổ bảng Lịch mềm lệ phí (nhấn vào nút Define... để hiển thị nó).

Phần trăm - lệ phí được hiện thị dưới dạng phần trăm của giá trị trao dịch

Bảng hoa hồng - Hoa hồng là một số tiền cố định bằng đô la (hoặc đơn vị tiền tệ của bạn) cho mỗi giao dịch.

$ mỗi cổ phiếu/hợp đồng - Hoa hồng được biểu diễn bằng đô la (hoặc đơn vị tiền tệ của bạn) cho mỗi cổ phiếu/hợp đồng được mua/bán.

Lãi suất hàng năm

Thiết lập này cho phép bạn xác định lãi suất hàng năm mà bạn kiếm được khi không tham gia thị trường hoặc khi vị trí của bạn thấp hơn vốn khả dụng.

Chu kỳ

Thiết lập này kiểm soát khoảng thời gian thanh mà bạn sử dụng cho việc kiểm tra lại/quét/khám phá/tối ưu hóa. Để kiểm tra lại dữ liệu intraday, bạn nên chuyển sang khoảng thời gian phù hợp và sau đó chạy kiểm tra lại.

Cho phép thu nhỏ kích thước vị trí

Nếu bạn đánh dấu ô này, AmiBroker sẽ thu nhỏ vị trí nếu vốn khả dụng ít hơn kích thước vị trí yêu cầu (qua biến PositionSize). Nếu ô này không được đánh dấu, các vị trí sẽ không được vào trong trường hợp đó.

Kích hoạt dừng ngay lập tức

Khi bạn giao dịch tại mức mở và muốn có các lệnh dừng tích hợp được kích hoạt trên cùng một thanh - chỉ cần đánh dấu ô này.

Nếu bạn giao dịch tại mức đóng và muốn các lệnh dừng tích hợp được kích hoạt từ thanh tiếp theo - hãy bỏ đánh dấu ô này.

Bạn có thể tự hỏi tại sao không đơn giản kiểm tra mảng buyprice hoặc shortprice xem nó có bằng mức mở không. Thật không may, điều này sẽ không hiệu quả. Tại sao? Bởi vì có những ngày doji khi giá mở bằng giá đóng, và khi đó bộ kiểm tra lại sẽ không bao giờ biết liệu giao dịch đã được thực hiện ở mức mở hay đóng.

Kích thước Lô Tròn

Các loại công cụ khác nhau được giao dịch với nhiều "đơn vị giao dịch" hoặc "khối" khác nhau. Ví dụ, bạn có thể mua một số lượng phần tử của quỹ đầu tư chung, nhưng bạn không thể mua một số lượng phần tử của cổ phiếu. Đôi khi, bạn phải mua theo các lô từ 10 đến 100. AmiBroker giờ đây cho phép bạn xác định kích thước của khối trên cấp độ toàn cầu và cấp độ từng ký hiệu.

Bạn có thể xác định kích thước lô tròn theo từng ký hiệu ở trang Thông tin ký hiệu (Symbol->Information page). Giá trị bằng không có nghĩa là ký hiệu không có kích thước lô tròn đặc biệt và sẽ sử dụng "Kích thước lô tròn mặc định" (thiết lập toàn cầu) từ trang Settings của cửa sổ Automatic Analysis. Nếu kích thước mặc định cũng được đặt bằng không, điều này có nghĩa là số lượng phần tử của cổ phiếu/hợp đồng được phép.

Bạn cũng có thể kiểm soát kích thước lô tròn trực tiếp từ công thức AFL của bạn bằng cách sử dụng biến dự phòng RoundLotSize, ví dụ:

```
RoundLotSize = 10;
```

Kích Thước Tick

Cài đặt này kiểm soát mức chuyển động giá tối thiểu của ký hiệu cho. Bạn có thể xác định nó trên cấp độ toàn cầu và từng ký hiệu. Giống như kích thước lô tròn, bạn cũng có thể xác định kích thước tick theo từng ký hiệu ở trang Thông tin ký hiệu. Giá trị bằng không hướng dẫn AmiBroker sử dụng "kích thước tick mặc định" được xác định trong trang Settings của cửa sổ Automatic Analysis. Nếu kích thước tick mặc định cũng được đặt bằng không, điều này có nghĩa là không có mức chuyển động giá tối thiểu nào.

Bạn cũng có thể đặt và lấy kích thước tick từ công thức AFL sử dụng biến dự trữ TickSize, ví dụ:

```
TickSize = 0.01;
```

Xin lưu ý rằng cài đặt kích thước tick chỉ ảnh hưởng đến các giao dịch được thoát bởi các lệnh dừng tích hợp và/hoặc ApplyStop(). Bộ kiểm tra lại giả định rằng dữ liệu giá tuân theo yêu cầu về kích thước tick và không thay đổi các mảng giá mà người dùng cung cấp.

Vì vậy, việc xác định kích thước tick chỉ có ý nghĩa nếu bạn đang sử dụng các lệnh dừng tích hợp, để các điểm thoát được tạo ra ở các mức giá "được phép" thay vì các mức tính toán. Ví dụ, ở Nhật Bản - bạn không thể có phần thập phân của yên, vì vậy bạn nên định

nghĩa kích thước tick toàn cầu là 1, để các lệnh dừng tích hợp thoát giao dịch ở các mức nguyên.

Tín hiệu vào ngược buộc thoát

Khi tùy chọn này BẬT (cài đặt mặc định) - bộ kiểm tra lại hoạt động giống như các phiên bản trước và đóng vị trí đã mở nếu gặp tín hiệu vào mới theo hướng ngược. Nếu công tắc này TẮT - ngay cả khi tín hiệu ngược xảy ra, bộ kiểm tra lại sẽ duy trì giao dịch hiện tại và không đóng vị trí cho đến khi có tín hiệu thoát thông thường (bán hoặc phủ).

Nói cách khác, khi công tắc này TẮT, bộ kiểm tra lại sẽ bỏ qua các tín hiệu ngắn trong suốt các giao dịch dài và bỏ qua các tín hiệu mua trong các giao dịch ngắn.

Cho phép thoát trên cùng một thanh (giao dịch một thanh)

Khi tùy chọn này BẬT - việc vào và thoát trong cùng một thanh là cho phép, khi nó TẮT thì việc thoát chỉ có thể xảy ra trên các thanh theo sau thanh vào. Bạn chỉ có thể bật tùy chọn "Cho phép thoát trên cùng một thanh" nếu bạn đang vào giao dịch tại mức MỞ. Nếu bạn vào giao dịch vào bất kỳ thời điểm nào khác ngoài mức mở của thanh, tùy chọn này nên được tắt để tránh việc nhìn vào tương lai.

Sử dụng QuickAFL

QuickAFL(tm) là một tính năng cho phép tính toán AFL nhanh hơn trong điều kiện nhất định. Ban đầu (từ 2003), nó chỉ có sẵn cho các chỉ báo, nhưng từ phiên bản 5.14+ nó cũng có sẵn trong Phân tích tự động.

Ý tưởng ban đầu là cho phép vẽ lại biểu đồ nhanh hơn thông qua việc tính toán công thức AFL chỉ cho phần hiển thị trên biểu đồ. Tương tự như vậy, cửa sổ phân tích tự động có thể sử dụng tập con của các bảng báo giá hiện có để tính AFL, nếu "range" được chọn nhỏ hơn "Tất cả bảng báo giá".

Bạn có thể xem thêm thông tin chi tiết về cách QuickAFL hoạt động và cách điều khiển nó tại bài viết trong Cơ sở Kiến thức này: http://www.amibroker.com/kb/2008/07/03/quickafl/

Lưu ý rằng tùy chọn này hoạt động trong backtester/optimizer, khám phá và quét.

Tab Giao dịch

Các trường giá mua/bán/khống chế/mua để bảo vệ - cho phép người dùng xác định giá mua/bán/khống/bù lỗ trong quá trình thử nghiệm hệ thống

delays mua/bán/khống chế/mua để bảo vệ - cho phép định nghĩa sự trễ tùy chỉnh giữa tín hiệu và giao dịch

Tab Dừng

dừng khi lỗ tối đa

dừng khi mục tiêu lợi nhuận

dừng theo dõi

dừng N-bar.

Xem chức năng **APPLYSTOP** để biết thêm chi tiết về các cài đặt dừng khác nhau.

Tab Báo cáo

Danh sách kết quả hiển thị

Loại danh sách kết quả mà backtester mới sử dụng. Các lựa chọn có thể:

Danh sách giao dịch (mặc định) - mỗi giao dịch được liệt kê trên một dòng riêng biệt. Giao dịch được sắp xếp theo ngày thoát mặc định

Nhật ký chi tiết - mỗi thanh dữ liệu được liệt kê riêng biệt. Nhật ký hiển thị điểm số, vị trí và thông tin chi tiết khác rất hữu ích cho việc gỡ lỗi hệ thống giao dịch/quy mô vị trí/chiến lược ghi điểm

Tóm tắt - một dòng cho mỗi backtest được tạo ra. Dòng này chứa tóm tắt/thống kê backtest (như báo cáo)

Lãi suất không rủi ro

Định nghĩa lãi suất không rủi ro cho Sharpe và UPI

Khoảng cách biểu đồ phân phối

Định khoảng cách của lợi nhuận, biểu đồ phân phối MAE và MFE. Khoảng cách là % lượng lợi nhuận/MAE/MFE cho mỗi thanh trong biểu đồ.

Tạo báo cáo chi tiết cho các backtest cá nhân.

Điều này dẫn đến việc trong chế độ kiểm tra lại cá nhân, báo cáo đầy đủ được tạo ra và lưu trữ cho mỗi chứng khoán đang được kiểm tra. Xin lưu ý rằng điều này sẽ làm chậm quá trình kiểm tra và tiêu tốn khá nhiều dung lượng ổ cứng.

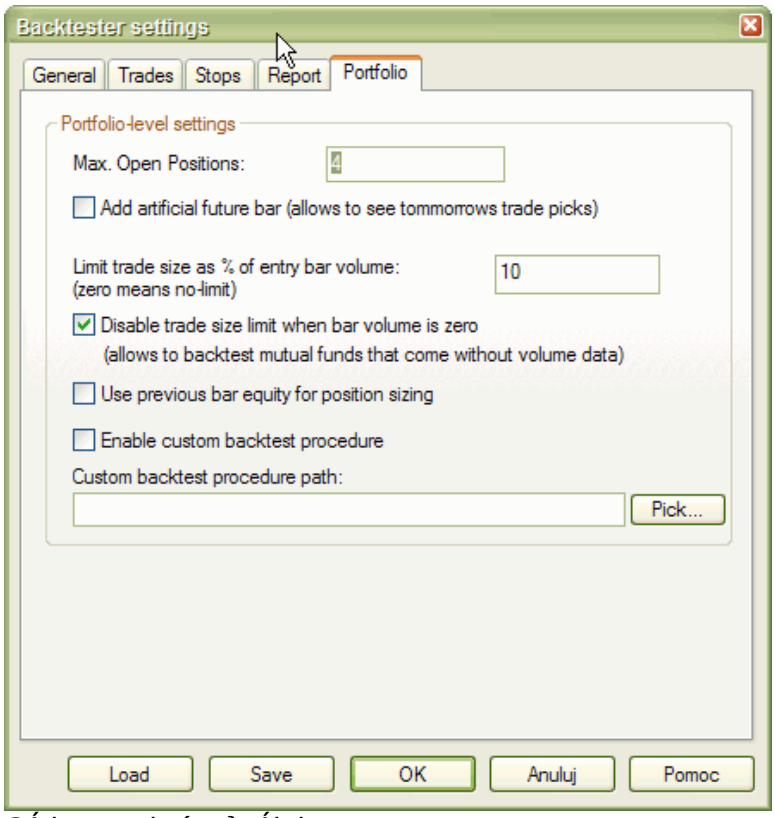
Bao gồm danh sách giao dịch trong báo cáo

Khi tùy chọn này được BẬT (mặc định), báo cáo kiểm tra lại cũng bao gồm danh sách giao dịch. Xin lưu ý rằng danh sách giao dịch có thể rất lớn và tiêu tốn nhiều dung lượng đĩa.

Cảnh báo trước các tối ưu hóa tốn thời gian

Khi tùy chọn này được BẬT (mặc định), AmiBroker sẽ hiển thị hộp thoại xác nhận khi tối ưu hóa của bạn có hơn 300 bước.

Tab danh mục đầu tư



Số lượng vị trí mở tối đa

Số lượng vị trí mở tối đa - số lượng tối đa các vị trí mở cùng một lúc. Có thể đặt cũng bằng cách sử dụng hàm `SetOption("MaxOpenPositions", number)`.

Thêm thanh tương lai giả

Khi được chọn, AmiBroker sẽ thêm thanh của ngày mai, cho phép bạn thấy các khuyến nghị giao dịch của ngày mai (hoặc thanh tiếp theo) khi hệ thống của bạn sử dụng độ trễ một thanh. Thanh tương lai giả có ngày tăng lên, khối lượng được đặt là không và tất cả các trường giá (OHLC) được đặt theo giá ĐÓNG của thanh dữ liệu cuối cùng.

Giới hạn kích thước giao dịch là % khối lượng thanh vào

Điều này ngăn không cho nhập các giao dịch lớn hơn tỷ lệ phần trăm đã cho của khối lượng thanh điều khiển. Ví dụ: nếu backtest dữ liệu hàng ngày và khối lượng hôm nay cho cổ phiếu giao dịch ít là 177.000 cổ phiếu, đặt giá trị này là 10% sẽ giới hạn kích thước giao dịch tối đa là 17.700 cổ phiếu (10% của tổng khối lượng hàng ngày). Điều này ngăn không cho 'ảnh hưởng đến thị trường' bởi các lệnh lớn.

LƯU Ý QUAN TRỌNG

Một số công cụ như QUỸ TƯƠNG HỢP không có dữ liệu VOLUME. Để backtest những công cụ như vậy, hãy đặt trường này thành ZERO (0) hoặc đánh dấu vào hộp "Tắt giới hạn kích thước giao dịch khi khối lượng thanh là không". Điều này hiệu quả tắt tính năng này. Nếu không, bạn sẽ không thể nhập bất kỳ giao dịch nào.

Tắt giới hạn kích thước giao dịch khi khối lượng thanh là không.

Khi nó được bật và khối lượng thanh nhập là không, backtester sẽ không áp dụng "giới hạn kích thước giao dịch như % của khối lượng thanh nhập" - điều này là để cho phép backtest quỹ tương hợp không có dữ liệu khối lượng. Khi nó được tắt và khối lượng thanh nhập là không, backtester sẽ không cho phép nhập giao dịch trên thanh đó.

Sử dụng cổ phiếu thanh trước cho việc xác định kích thước vị trí.

Ảnh hưởng đến việc xác định kích thước vị trí theo phần trăm của cổ tức hiện tại. Không đánh dấu (giá trị mặc định) có nghĩa là: sử dụng cổ tức hiện tại (trong ngày) để xác định kích thước vị trí, đánh dấu có nghĩa là: sử dụng cổ tức cuối cùng của thanh trước để xác định kích thước vị trí.

Kích hoạt thủ tục backtest tùy chỉnh.

Khi được chọn, AmiBroker sẽ áp dụng công thức backtest tùy chỉnh được chỉ định trong trường dưới đây cho mỗi backtest bạn thực hiện. Điều này hữu ích nếu bạn muốn thêm vĩnh viễn các số liệu tùy chỉnh của bạn vào tất cả các backtest mà không cần copy paste cùng một dòng mã.

Đường dẫn thủ tục backtest tùy chỉnh

Đường dẫn đầy đủ đến công thức backtest tùy chỉnh (xem phía trên).

Tab cũ

Các số liệu Drawdown dựa trên...

Các chỉ số giảm giá trong báo cáo kiểm tra lùi đo lường sự giảm giá vốn chủ sở hữu trong quá trình giao dịch. Để tính toán mức giảm, bạn có thể sử dụng kịch bản tồi tệ nhất: giá thấp cho giao dịch mua và giá cao cho giao dịch bán, hoặc giá đơn (giá mở hoặc giá đóng) cho cả giao dịch mua và bán. Tùy chọn "Các chỉ số giảm giá dựa trên..." (hình 2) cho phép bạn chọn giá được sử dụng để tính toán mức giảm giá. Sử dụng kịch bản tồi tệ nhất sẽ cho bạn các mức giảm lớn hơn vài phần trăm so với việc sử dụng giá đóng hoặc giá mở. Mặt khác, hàm Equity() luôn sử dụng mảng giá bán/giá bù nên bạn có thể chọn trường giá mở hoặc giá đóng ở đây để khớp với mức giảm như quan sát trong đường vốn chủ sở hữu.

Công thức - đánh dấu ô này để bao gồm công thức AFL trong báo cáo kiểm tra lùi.

Cài đặt - đánh dấu ô này để bao gồm cài đặt trong báo cáo kiểm tra lùi.

Incl. pos ngoại thị trường - đánh dấu vào ô này để bao gồm các vị trí ngoại thị trường trong báo cáo backtest

Tổng kết chung - đánh dấu vào ô này để bao gồm tổng số kết quả backtest của từng biểu tượng

Tổng kết biểu tượng - đánh dấu vào ô này để bao gồm các tổng kết theo biểu tượng

Danh sách giao dịch - chọn định dạng của danh sách giao dịch được bao gồm trong báo cáo

% Tiếp xúc - 'Phần trăm tiếp xúc thị trường của hệ thống giao dịch được tính trên cơ sở từng thanh. Tổng của phần trăm tiếp xúc thanh chia cho số lượng thanh. Phần trăm tiếp xúc của một thanh là giá trị của các vị trí mở chia cho tổng số vốn.

Phần trăm lợi nhuận được điều chỉnh theo rủi ro Net - Phần trăm lợi nhuận net chia cho % Tiếp xúc %

Lợi nhuận hàng năm - Lợi nhuận hàng năm đầu tư tăng trưởng (CAR) - đây là %

Lợi nhuận được điều chỉnh theo rủi ro - Phần trăm lợi nhuận hàng năm chia cho % Tiếp xúc %.

Lãi/Lỗ trung bình - (Lợi nhuận của người thắng cuộc + Lỗ của người thua)/(số giao dịch) %.

Lãi/Lỗ trung bình - '(% Lợi nhuận của người thắng cuộc + % Lỗ của người thua)/(số giao dịch)

Thanh trung bình được giữ - tổng số thanh trong giao dịch / số giao dịch

Rớt giá giao dịch tối đa - Sự giảm từ đỉnh đến thung lũng lớn nhất xảy ra trong bất kỳ một giao dịch nào

Rớt giá % giao dịch tối đa - Sự giảm từ đỉnh đến thung lũng lớn nhất theo phần trăm xảy ra trong bất kỳ một giao dịch nào

Rớt giá hệ thống tối đa - Sự giảm từ đỉnh đến thung lũng lớn nhất xảy ra trong tổng số vốn

Rớt giá % hệ thống tối đa - Sự giảm từ đỉnh đến thung lũng lớn nhất theo phần trăm xảy ra trong tổng số vốn

Hệ số phục hồi - Lợi nhuận ròng chia cho rớt giá hệ thống tối đa

CAR/MaxDD - Lợi nhuận hàng năm đầu tư tăng trưởng phần trăm chia cho rớt giá % hệ thống tối đa

RAR/MaxDD - Lợi nhuận được điều chỉnh theo rủi ro chia cho rớt giá % hệ thống tối đa

Hệ số lợi nhuận - Lợi nhuận của người thắng cuộc chia cho lỗ của người thua

Tỉ lệ Thanh toán - Tỉ lệ lợi nhuận trung bình / lỗ trung bình

Lỗi chuẩn - Lỗi chuẩn đo sự không ổn định của dòng vốn. Càng thấp càng tốt.

Tỉ lệ Rủi ro-Thưởng - Đo lường mối quan hệ giữa rủi ro tiềm ẩn trong một hệ thống giao dịch so với lợi nhuận tiềm năng của nó. Càng cao càng tốt. Được tính như là độ dốc của đường vốn (lợi nhuận hàng năm dự kiến) chia cho lỗi chuẩn của nó.

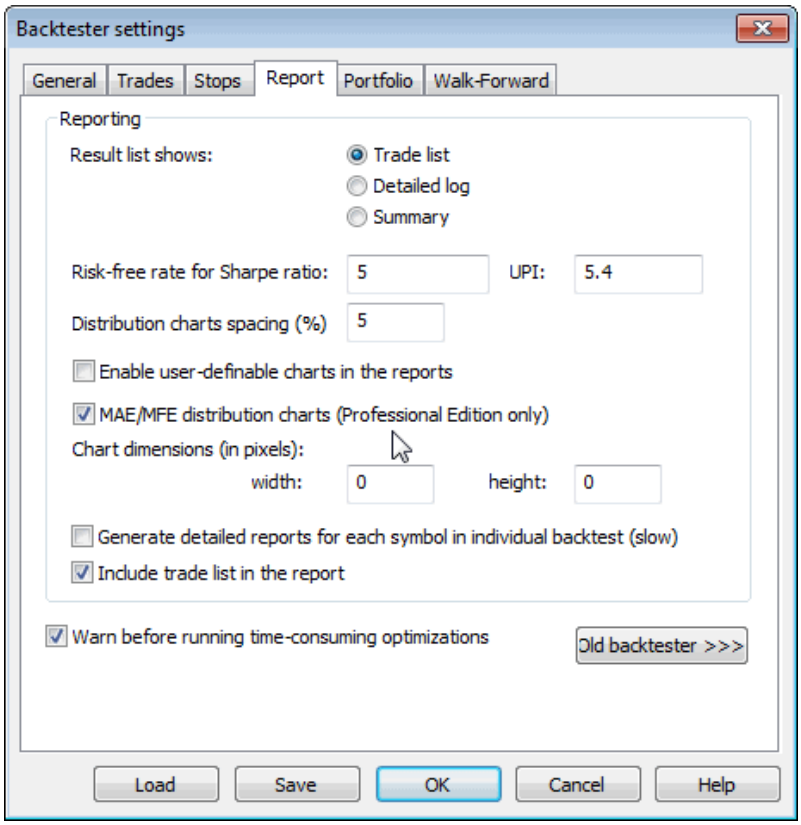
Chỉ số Ulcer - Căn bậc hai của tổng các mức giảm giá bình phương chia cho số lượng thanh

Chỉ số Hiệu suất Ulcer - (Lợi nhuận hàng năm - Lợi nhuận từ trái phiếu kho bạc) / Chỉ số Ulcer. Hiện tại, lợi nhuận từ trái phiếu kho bạc được mã hóa cứng ở mức 5.4. Trong phiên bản tương lai, sẽ có cài đặt cho người dùng về điều này.

Tỷ lệ Sharpe của các giao dịch - Đo lường lợi nhuận điều chỉnh theo rủi ro của khoản đầu tư. Trên 1.0 là tốt, trên 2.0 là rất tốt. Thông tin thêm: http://www.stanford.edu/~wfsharpe/art/sr/sr.htm. Tính toán: trước tiên, tỷ lệ lợi nhuận trung bình và độ lệch chuẩn của các lợi nhuận được tính toán. Sau đó, hai chỉ số này được annual hóa bằng cách nhân chúng với tỷ lệ (NumberOfBarsPerYear)/(AvgNumberOfBarsPerTrade)(\text{NumberOfBarsPerYear}) / (\text{AvgNumberOfBarsPerTrade})(NumberOfBarsPerYear)/(AvgNumberOfBarsPerTrade). Sau đó, tỷ lệ sinh lời không rủi ro (hiện tại được mã hóa cứng ở mức 5) sẽ được trừ đi từ lợi nhuận trung bình hàng năm và sau đó chia cho độ lệch chuẩn hàng năm của các lợi nhuận.

Tỷ lệ K - Phát hiện sự không nhất quán trong lợi nhuận. Nên có giá trị từ 1.0 trở lên. Tỷ lệ K càng cao thì lợi nhuận mà bạn có thể mong đợi từ hệ thống càng nhất quán. Đường xu hướng hồi quy tuyến tính của đường vốn chủ sở hữu nhân với căn bậc hai của tổng các độ lệch bình phương của số lượng thanh chia cho sai số chuẩn của đường vốn chủ sở hữu nhân với căn bậc hai của số lượng thanh. Thông tin thêm: Stocks & Commodities V14:3 (115-118): Đo lường Hiệu suất Hệ thống của Lars N. Kestner.

BÁO CÁO KIỂM TRA LÙI CŨ



Cửa sổ này (truy cập từ nút **Báo cáo** trong cửa sổ Phân tích Tự động) cung cấp thông tin rất hữu ích về hiệu suất của một hệ thống giao dịch đang được kiểm tra. Thông tin ở đây có thể được tùy chỉnh bằng cách sử dụng hộp thoại cài đặt kiểm tra hệ thống.

Giải thích các giá trị:

Lợi nhuận ròng tổng: Đây là tổng lợi nhuận/lỗ thực hiện bởi cuộc kiểm tra. Bao gồm giá trị đã đóng của vị thế mở (nếu có).

Lợi nhuận trên tài khoản: Đây là tổng lợi nhuận/lỗ dưới dạng phần trăm của khoản đầu tư ban đầu.

Tổng hoa hồng đã trả: Số tiền hoa hồng đã trả trong các giao dịch.

Lợi nhuận/lỗ từ vị thế mở: Giá trị đã đóng của vị thế mở tồn tại vào cuối cuộc kiểm tra.

Lợi nhuận mua và giữ: Tổng lợi nhuận/lỗ thực hiện bởi chiến lược mua và giữ (bao gồm cả hoa hồng).

Tỷ lệ lợi nhuận mua và giữ: Tổng lợi nhuận từ chiến lược mua và giữ dưới dạng phần trăm của khoản đầu tư ban đầu.

Số thanh trong kiểm tra: Số lượng thanh đã được kiểm tra (tóm tắt tổng thể cho thấy tổng số thanh của tất cả các ký hiệu).

Số ngày trong kiểm tra: Số ngày giữa ngày của thanh đầu tiên và ngày của thanh cuối cùng (tóm tắt tổng thể cho thấy giá trị trung bình số ngày qua số lượng biểu tượng được kiểm tra)

Hệ thống mua và giữ chỉ số (System to buy-and-hold index): Chỉ số này cho thấy hệ thống hoạt động tốt hay kém hơn so với chiến lược mua và giữ. Giá trị 0% có nghĩa là hệ thống mang lại lợi nhuận tương đương với chiến lược mua và giữ. Giá trị 200% có nghĩa là hệ thống mang lại lợi nhuận cao hơn 200% so với chiến lược mua và giữ. Một giá trị -50% có nghĩa là hệ thống chỉ mang lại một nửa lợi nhuận so với chiến lược mua và giữ.

Lợi tức hàng năm của hệ thống (%): Ít được tính dựa trên lợi tức phần trăm hàng năm hợp thành của hệ thống (xem ghi chú)

Lợi tức hàng năm của chiến lược mua và giữ (%): Ít được tính dựa trên lợi tức phần trăm hàng năm hợp thành của chiến lược mua và giữ (xem ghi chú)

Lượng tiền lớn nhất mà hệ thống có thể mất (System drawdown): Mức độ mà quỹ đầu tư giảm sút nhiều nhất (so với số vốn đầu tư ban đầu).

Lượng tiền lớn nhất mà chiến lược mua và giữ có thể mất (B&H drawdown): Mức độ mà quỹ đầu tư giảm sút nhiều nhất (so với số vốn đầu tư ban đầu).

Giảm giá tối đa của hệ thống (Max. system drawdown): Khoảng cách điểm lớn nhất từ đỉnh cao cho đến đáy thấp tiếp theo mà hệ thống trải qua.

Giảm giá tối đa theo phần trăm của hệ thống (Max. system % drawdown): Khoảng cách phần trăm lớn nhất từ đỉnh cao cho đến đáy thấp tiếp theo mà hệ thống…

Giảm giá tối đa của chiến lược mua và giữ (Max. B&H drawdown): Khoảng cách điểm lớn nhất từ đỉnh cao cho đến đáy thấp tiếp theo mà chiến lược mua và giữ trải qua.

Giảm giá giao dịch: Mức giảm vốn chủ sở hữu lớn nhất mà bất kỳ giao dịch đơn lẻ nào trải qua (so với giá vào lệnh của giao dịch đó).

Giảm giá giao dịch tối đa: Khoảng cách điểm lớn nhất giữa giá trị đỉnh vốn chủ sở hữu và giá trị đáy tiếp theo mà bất kỳ giao dịch đơn lẻ nào trải qua.

Giảm giá % giao dịch tối đa: Khoảng cách phần trăm lớn nhất giữa giá trị đỉnh vốn chủ sở hữu và giá trị đáy tiếp theo mà bất kỳ giao dịch đơn lẻ nào trải qua.

Tổng số giao dịch: Số lượng giao dịch (thắng + thua).

Phần trăm giao dịch có lợi: Số lượng giao dịch thắng so với tổng số giao dịch, được thể hiện dưới dạng phần trăm.

Lợi nhuận của giao dịch thắng/Loss của giao dịch thua: Tổng số tiền kiếm được từ các giao dịch thắng/thua.

Tổng số thanh trong giao dịch thắng/thua: Số lượng thanh được sử dụng trong các giao dịch thắng/thua.

Giao dịch thắng/thua lớn nhất: Số tiền của giao dịch thắng/thua lớn nhất.

Số thanh trong giao dịch thắng/thua lớn nhất: Số lượng thanh trong giao dịch thắng/thua lớn nhất.

Giao dịch thắng/thua trung bình: Trung bình của các giao dịch thắng/thua (tổng số tiền thắng/thua chia cho số lượng giao dịch thắng/thua).

Số thanh trung bình trong giao dịch thắng/thua: Trung bình số thanh trong các giao dịch thắng/thua (tổng số thanh trong các giao dịch thắng/thua chia cho số lượng giao dịch thắng/thua).

Số lượng giao dịch thắng/thua liên tiếp lớn nhất: Số lượng giao dịch thắng/thua liên tiếp lớn nhất.

Số thanh ngoài thị trường: Số thanh mà hệ thống hoàn toàn không tham gia thị trường (không có vị thế mua hoặc bán). Nếu bạn mở và đóng vị thế trong cùng một ngày, ngay cả khi không có vị thế mở lúc thị trường mở và không có vị thế vào lúc đóng, ngày này KHÔNG được coi là ra ngoài thị trường.

Lãi suất kiếm được: Tổng lãi suất kiếm được giữa các giao dịch. Lưu ý rằng AmiBroker mô phỏng tiền gửi qua đêm (O/N). Điều này có nghĩa là nếu bạn đóng vị thế vào thứ Hai và mở vị thế tiếp theo vào thứ Ba, bạn sẽ kiếm được lãi cho một khoản tiền gửi qua đêm.

Rủi ro tiếp xúc (Exposure): Cho biết bạn tiếp xúc với thị trường ở mức độ nào. Đó là tỷ lệ số thanh trong thị trường chia cho tổng số thanh đang được kiểm tra. (Số lượng thanh trong thị trường được tính bằng tổng số thanh trừ đi số thanh ngoài thị trường)

Lợi nhuận điều chỉnh theo rủi ro hàng năm: Thể hiện lợi nhuận hàng năm của hệ thống (*xem ghi chú) đã được điều chỉnh (chia) theo phơi nhiễm thị trường. Nếu hệ thống của bạn tăng 10% trong một năm với mức phơi nhiễm là 50%, suất lợi nhuận được điều chỉnh sẽ là 20% (10%/0.5).

Tỉ lệ trung bình thắng/trung bình thua: Giá trị tuyệt đối của tỷ lệ giao dịch trung bình thắng so với giao dịch trung bình thua.

Hệ số lợi nhuận: Giá trị tuyệt đối của tỷ lệ lợi nhuận từ giao dịch thắng so với lỗ từ giao dịch thua.

Giao dịch trung bình (thắng & thua): Lợi nhuận giao dịch trung bình được tính là tổng lợi nhuận từ giao dịch thắng và thua chia cho tổng số giao dịch.

*Ghi chú: Phương pháp tính lợi nhuận hàng năm theo tỷ lệ phần trăm sử dụng:

Phần lớn phần mềm (bao gồm cả hai gói phần mềm được gọi là chuyên nghiệp phổ biến nhất) sử dụng phương pháp cung cấp tỷ lệ theo năm rất đơn giản dựa trên công thức sau:

simple_annualized_percentage_return = percentage_return * (365 / days_in_test);

thật không may, phương pháp này là sai và gây hiểu lầm rất lớn vì nó sẽ cho bạn biết lợi nhuận hằng năm là 22% khi hệ thống của bạn kiếm được 44% trong hai năm. Giá trị này quá lạc quan. Trên thực tế, lợi nhuận hàng năm trong trường hợp này chỉ là 20%: nếu vốn ban đầu của bạn là 10000, bạn kiếm được 20% trong năm đầu tiên vì vậy bạn sau đó nhận được 12000 và 20% trong năm thứ hai chúng ta thu được 14400 = (12000 * 120%).Sau hai năm, bạn kiếm được 44% nhưng hàng năm chỉ là 20%.

AmiBroker là một trong số ít các chương trình tính toán lợi nhuận hàng năm một cách chính xác và sẽ cho bạn giá trị chính xác là 20% như hình ví dụ trên. Công thức mà AmiBroker sử dụng để tính lượt xem hàng năm như sau:

correctly_annualized_perc_return = 100%*((final_value/initial_value)^(365/days_in_test) - 1).

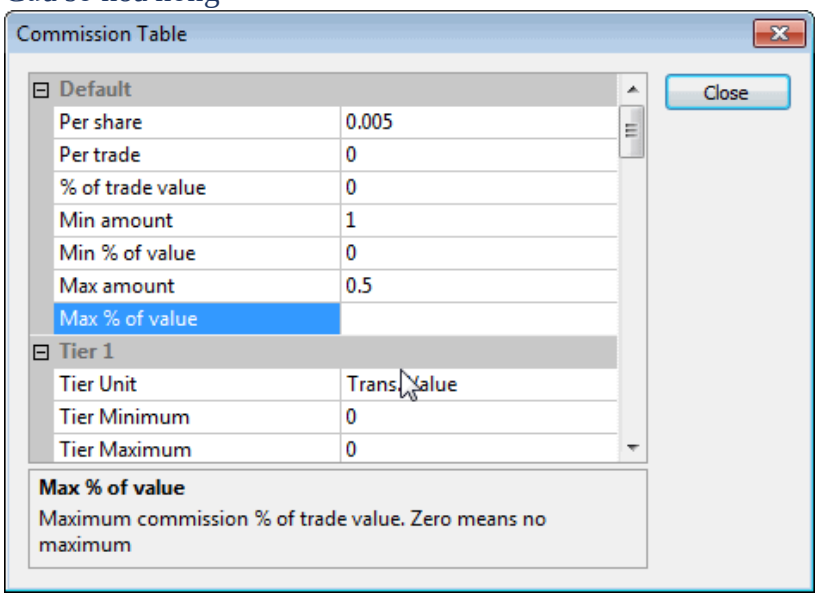
Trong đó x^y có nghĩa là lũy thừa của x lên y.

Dưới đây là bảng so sánh những khác biệt giữa thống kê được sản xuất bởi "kiểm tra lùi cũ" và "kiểm tra lùi mới (danh mục đầu tư)":

	Kiểm tra lùi cũ	Kiểm tra lùi mới (danh mục đầu tư)
Hệ thống và gao dịch tính toán giảm giả	Giá mở/Đóng/Phạm vi H-L (kịch bản tồi tệ nhất) có thể chọn trong cài đặt.	Chỉ dựa trên giá đóng (không phụ thuộc vào cài đặt) - có thể thay đổi.

Giảm giá % giao dịch tối đa	Tính toán dựa trên giá trị tổng	Tính toán dựa trên giá trị thực tế của giao dịch tại điểm vào.
Thống kê có sẵn	Cho tất cả các giao dịch chỉ	Phân tích riêng cho giao dịch dài, ngắn và tất cả
Kích thước vị trí	Dựa trên vốn của từng mã	Dựa trên vốn của danh mục đầu tư Kích thước vị trí = -25 Sẽ vào 25% vốn hiện tại của danh mục đầu tư
Thống kê giao dịch	Chỉ bao gồm các giao dịch đã đóng, gioa dịch mở được báo cáo riêng biệt	Bao gồm tất cả giao dịch(đã đóng và còn mở tại cuối kỳ phân tích). bất kỳ giao dịch mở nào cũng sẽ được đóng ở giá “đóng”
Rủi ro	Tính toán không phụ thuộc vào kích thước vị trí (không quan tâm đến kích thước vị trí, nếu giao dịch được thực hiện cho thanh cụ thể thì giả định 100% rủi ro tại thanh đó)	Tính toán bây giờ (trong phiên bản 4.43.0) bao gồm tổng số vị trí mở so với tổng vốn danh mục đầu tư. Rủi ro được tính toán theo từng thanh, vì vậy nếu chỉ có 50% vốn trong giao dịch mở, thì rủi ro cho thanh đó là 0.5. Sau đó, các rủi ro của từng thanh được cộng lại và chia cho số thanh để tạo ra chỉ số rủi ro. Bằng cách này, rủi ro thực tế trên thị trường được tính toán.
Kiểm tra nhiều chứng khoán	N tài khoản độc lập (nhiều vốn riêng lẻ)	Vốn danh mục chung cho tất cả các mã được thử nghiệm

Cửa sổ hoa hồng



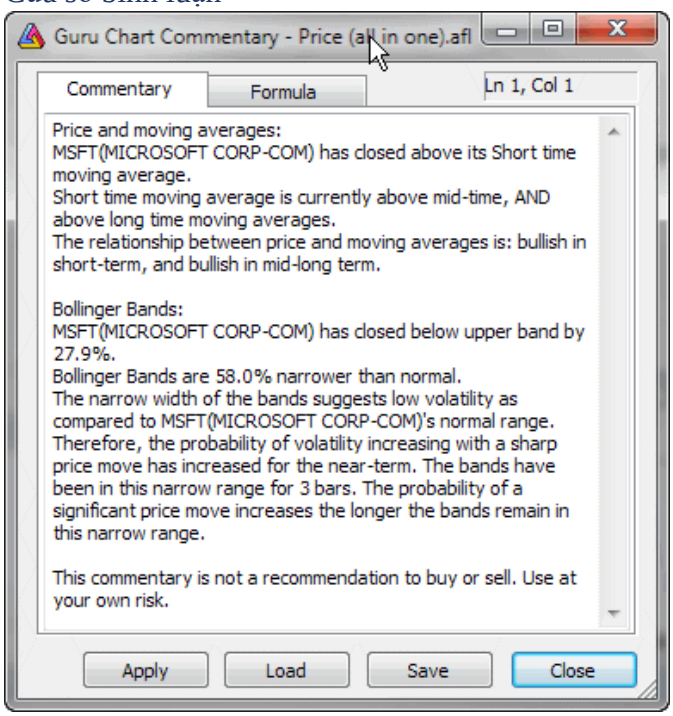
Bảng phí có sẵn trong Trình quản lý Tài khoản và trong Phân tích Tự động -> Cửa sổ Cài đặt, tab "Chung", "Phí và tỷ lệ: Định nghĩa...".

Trong cửa sổ này, bạn có thể nhập phí áp dụng cho các giao dịch mua/bán.

Có 5 cấp độ trong bảng lịch phí cộng thêm một cấp độ "mặc định" được sử dụng khi các cấp độ khác không được định nghĩa hoặc giao dịch không khớp với bất kỳ cấp độ nào đã được định nghĩa. Các cấp độ có thể được định nghĩa dựa trên giá trị giao dịch hoặc số lượng cổ phiếu/hợp đồng được giao dịch. Mỗi cấp độ có thể định nghĩa giới hạn tối thiểu và tối đa do người dùng xác định. Nếu giới hạn tối thiểu/tối đa không được định nghĩa hoặc đặt bằng không, cấp độ đó sẽ không hoạt động.

Mỗi cấp độ cho phép định nghĩa phí dựa trên mỗi cổ phiếu, mỗi giao dịch, hoặc tỷ lệ phần trăm của khối lượng giao dịch và cho phép xác định các giá trị phí tối thiểu và tối đa dựa trên giá trị đô la hoặc tỷ lệ phần trăm.

Cửa sổ bình luận



Cửa sổ bình luận cho phép bạn xem các mô tả văn bản về tình hình kỹ thuật thực tế trên thị trường cụ thể.

Các bình luận được tạo ra bằng cách sử dụng các công thức viết bằng ngôn ngữ công thức riêng của AmiBroker. Bạn có thể tìm thấy mô tả về ngôn ngữ này trong Hướng dẫn Tham khảo Ngôn ngữ Công thức AmiBroker.

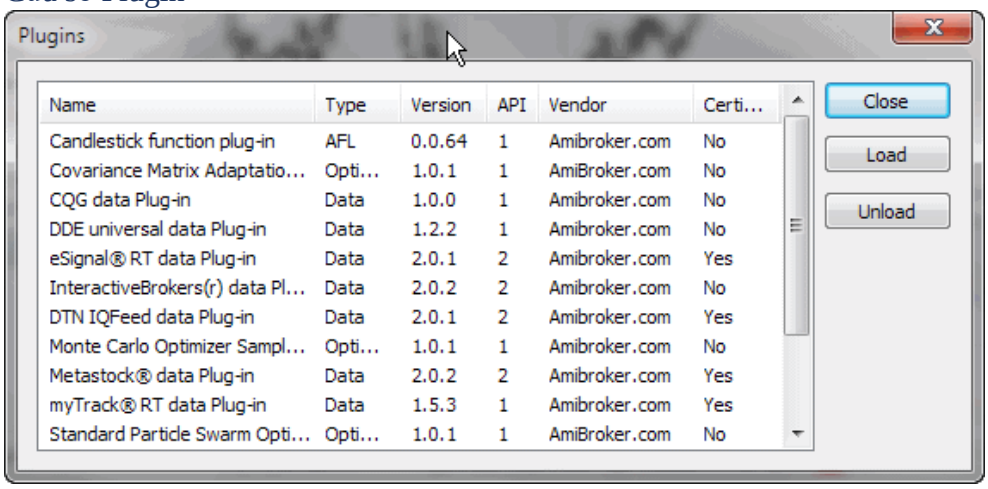
Hơn nữa, tính năng bình luận còn cung cấp cả hình ảnh biểu diễn các tín hiệu mua và bán bằng cách đặt các dấu hiệu (mũi tên) trên biểu đồ giá.

Người mới bắt đầu nên đọc "Hướng dẫn: Cách viết bình luận của riêng bạn" để có hướng dẫn từng bước và làm việc với trình chỉnh sửa AFL.

Nút "Làm mới" khiến AmiBroker diễn giải lại bình luận sử dụng ký hiệu/ngày hiện tại đã chọn. Các nút "Tải" và "Lưu" cho phép tải/lưu các công thức bình luận. Nút "Đóng" đóng cửa sổ bình luận.

Bây giờ, cửa sổ bình luận biểu đồ Guru được tự động cập nhật và đồng bộ với ngày được chọn trên biểu đồ bằng công cụ chọn "Pick". Bằng cách này, bạn có thể dễ dàng đọc bất kỳ giá trị chỉ báo nào vào bất kỳ ngày đã chọn nào ngay từ cửa sổ bình luận biểu đồ.

Cửa sổ Plugin



Cửa sổ Plugin liệt kê tất cả các DLL plugin đã được tải. Nó hữu ích để kiểm tra các plugin nào đang hoạt động.

Ngoài việc chỉ hiển thị danh sách các plugin, bạn có thể giải phóng tất cả các DLL bằng cách nhấn nút "Unload" và tải lại chúng bằng cách nhấn nút "Load". Xin lưu ý rằng một DLL phải được đặt trong thư mục con "Plugins" của thư mục chính AmiBroker để có thể nhìn thấy.

Khi khởi động, AmiBroker quét thư mục "Plugins" và tải các DLL phù hợp với thông số kỹ thuật của plugin AmiBroker. Nếu một DLL được tải, nó sẽ "bị khóa" để không thể ghi đè hoặc sửa đổi.

Trong quá trình phát triển, cần thiết phải ghi đè/sửa đổi mã DLL - vì khi bạn áp dụng các thay đổi cho mã nguồn, các thay đổi này phải được biên dịch lại và lưu vào tệp DLL. Để cho phép nhà phát triển ghi đè DLL được sử dụng bởi AmiBroker, chức năng "Unload" có sẵn trong cửa sổ này. Giải phóng DLL cho phép nó được ghi đè mà không cần khởi động lại AmiBroker. Sau đó, sau khi sửa đổi mã DLL, bạn có thể tải lại DLL bằng cách sử dụng chức năng "Load".

LƯU Ý QUAN TRỌNG: AmiBroker không đảm bảo về tính năng và hiệu suất của các plugin bên thứ ba không được chứng nhận. Cụ thể, một số plugin có thể gây ra sự không ổn định hoặc thậm chí lỗi hệ thống. Việc sử dụng hoàn toàn các plugin bên thứ ba không được chứng nhận là rủi ro của riêng bạn.

Hướng dẫn bảo dưỡng Indicator

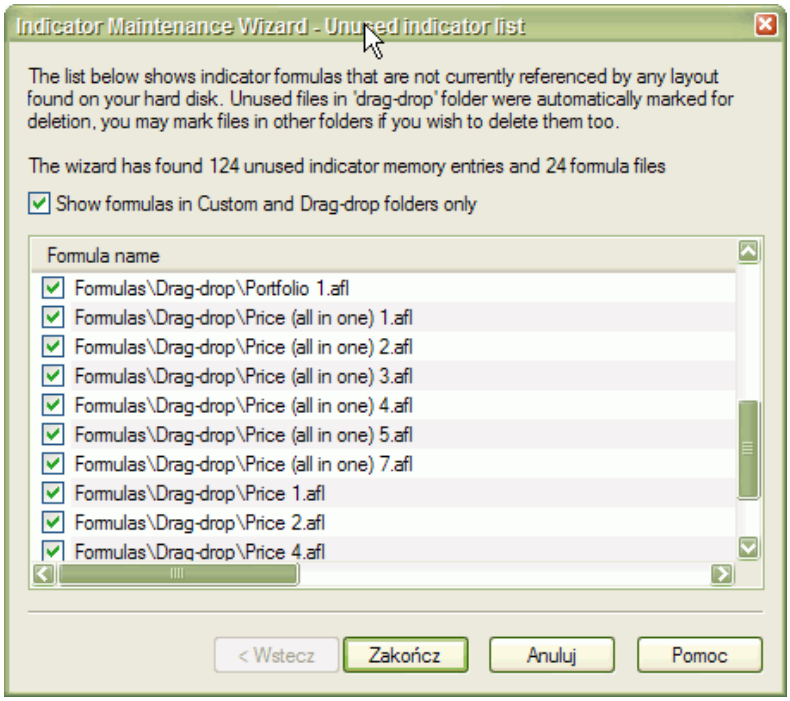
Hệ thống bảo dưỡng Indicator kiểm tra bất kỳ indicator nào đã bị xóa khỏi các layout trên ổ đĩa cứng của bạn và giải phóng bảng khỏi các mục đã được phân bổ cho các indicator đã bị xóa. Quy trình này hiếm khi nhưng vẫn cần thiết vì nếu bạn xóa indicator khỏi một layout, không có bảo đảm rằng không có tệp layout khác nằm ở đâu đó trên ổ đĩa cứng của bạn vẫn tham chiếu đến indicator đã cho.

Vì vậy, hệ thống bảo dưỡng Indicator quét tất cả các ổ đĩa cứng và tất cả các phân vùng để tìm các tệp layout và phân tích chúng để xây dựng bảng "thực sự được sử dụng" của các indicator.



Những cái không được tham chiếu bởi bất kỳ layout nào có thể được xóa khỏi bảng nội bộ.

Tùy thuộc vào lựa chọn của bạn, bạn có thể để mặc định hành vi (chỉ làm sạch bảng nội bộ) hoặc xóa các tệp công thức thực tế không được tham chiếu. Điều này tùy thuộc vào bạn. Nếu bạn không sử dụng công thức cụ thể cho AA Scan/Backtest/Optimization bạn có thể xóa nó. Nếu bạn sử dụng nó hoặc cần nó cho mục đích lưu trữ nào đó - hãy để nó không được kiểm tra.



Nếu bạn không chắc chắn nên chọn tùy chọn nào, chỉ cần nhấn "Tiếp theo" liên tục là bạn có thể hoàn tất quy trình một cách an toàn mà không cần thay đổi bất kỳ cài đặt nào.

Cửa sổ Nhật ký

Cửa sổ Nhật ký (có sẵn từ menu Window -> Log) cho phép bạn xem:

- Các lỗi thời gian chỉnh sửa được hiển thị trong quá trình kiểm tra công thức.
- Các lỗi thời gian chạy xảy ra khi công thức đang chạy (không phải khi chỉnh sửa).
- Đầu ra của lệnh _trace trong AmiBroker (không sử dụng trình gỡ lỗi bên thứ ba).

Để thực hiện các tác vụ như xóa đầu ra, sao chép, thay đổi cài đặt, bạn sử dụng chuột phải nhấp vào danh sách trong cửa sổ nhật ký. Nhấp đúp vào dòng lỗi sẽ mở trình chỉnh sửa, giúp bạn sửa lỗi dễ dàng.

Trong khi danh sách lỗi "thời gian chỉnh sửa" được xóa tự động mỗi khi bạn kiểm tra cú pháp trong trình chỉnh sửa, danh sách lỗi thời gian chạy KHÔNG được xóa, vì vậy tất cả các lỗi vẫn được liệt kê, ngay cả khi chúng đã được sửa, trừ khi bạn tự tay xóa danh sách.

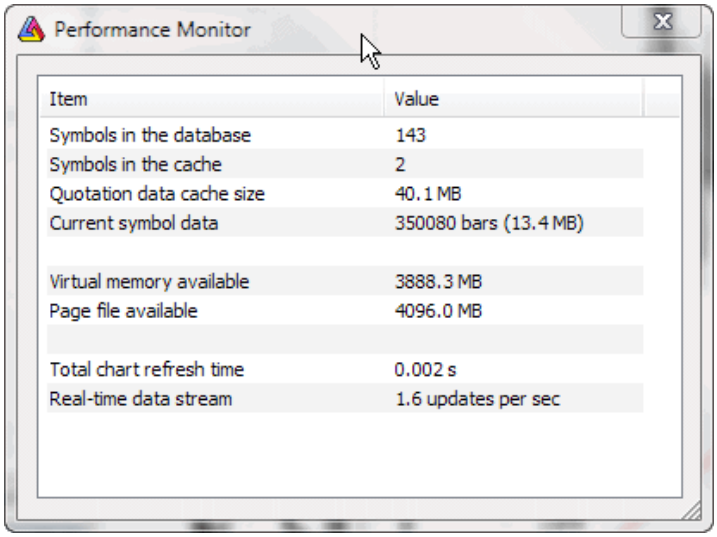
Lưu ý rằng đầu ra _TRACE theo mặc định được chuyển đến trình gỡ lỗi bên ngoài (như DebugView), để hiển thị nội bộ, bạn cần chuyển đổi tùy chọn thích hợp trong Tools -> Preferences -> AFL. Bạn có thể chọn hiển thị nội bộ, bên ngoài hoặc ở cả hai nơi.

Nhật ký nội bộ _trace có mức phí hiệu suất thấp hơn nhiều (được đo theo bậc) so với bên ngoài, nhưng điều này đạt được nhờ vào việc cửa sổ nhật ký nội bộ chỉ được làm mới khi ứng dụng không bận rộn. Điều này thích hợp cho một số mục đích, nhưng bạn có thể thích sự làm mới tức thì mà DebugView cung cấp hơn.

Lưu ý rằng cửa sổ nhật ký nội bộ chấp nhận chuỗi đặc biệt "!CLEAR!" mà sẽ xóa nội dung của cửa sổ nhật ký, như được trình bày trong ví dụ dưới đây:

```
_TRACE("!CLEAR!"); // this clears the internal log window
_TRACE("First line after clear)
```

Cửa sổ Giám sát Hiệu năng



Cửa sổ giám sát hiệu năng có thể truy cập từ menu Tools->Performance Monitor và nó hiển thị một số số liệu thống kê về bộ nhớ và sử dụng:

- số lượng biểu tượng trong cơ sở dữ liệu

- số lượng biểu tượng được lưu trữ trong RAM

- bộ nhớ sử dụng dữ liệu báo giá

- sử dụng bộ nhớ biểu tượng hiện tại

- tổng thời gian làm mới biểu đồ

- tần suất cập nhật luồng dữ liệu thời gian thực

Nội dung của cửa sổ được cập nhật tự động mỗi 3 giây.

Công cụ này dự định được sử dụng ngay bây giờ cho hai mục đích:

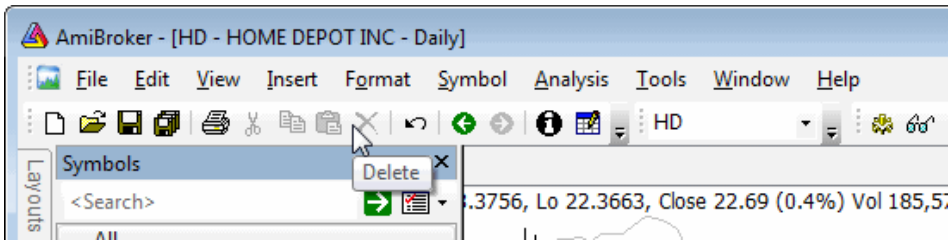
a) tinh chỉnh cài đặt bộ nhớ đệm để sử dụng RAM tốt nhất (ví dụ các tối ưu hóa sẽ chạy nhanh hơn nếu tất cả dữ liệu báo giá có thể được giữ trong RAM)

b) giám sát hiệu suất thời gian thực

Nhiều ứng dụng khác có thể sẽ xuất hiện trong tương lai.

Các Menu

Chương này mô tả các menu của AmiBroker.



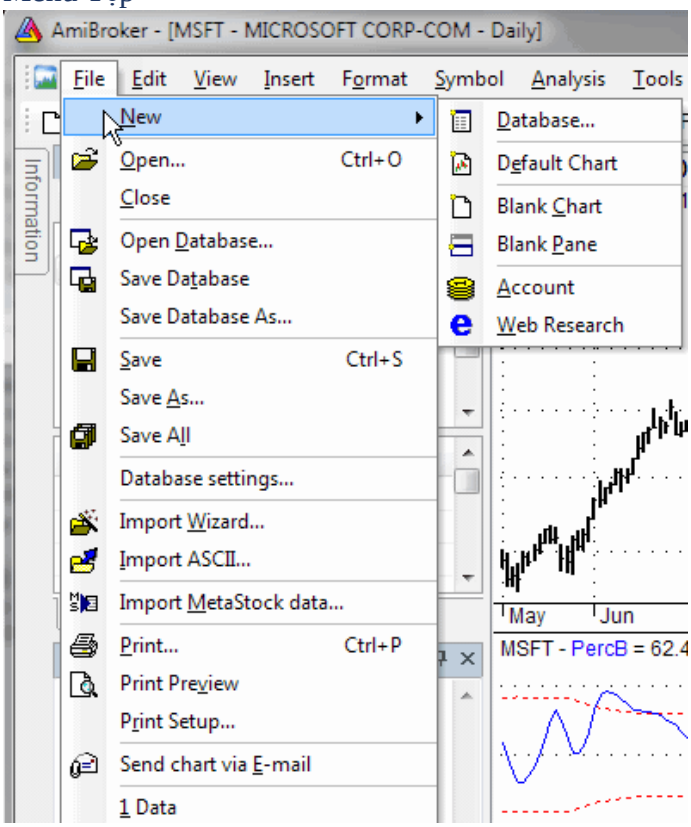
Có các menu chính sau:

- **Tệp**
- **Chỉnh sửa**
- **Xem**
- **Chèn**
- **Định dạng**
- **Ký hiệu**
- **Phân tích**
- **Công cụ**
- **Cửa sổ**
- **Trợ giúp**

Và các menu ngữ cảnh sau:

- Menu ngữ cảnh của Trình chỉnh sửa AFL (có sẵn khi bạn nhấp chuột phải trong trình chỉnh sửa AFL)
- Menu ngữ cảnh của cửa sổ Phân tích Tự động (có sẵn khi bạn nhấp chuột phải vào DANH SÁCH KẾT QUẢ Phân tích Tự động)
- Menu ngữ cảnh của đầu ra Cảnh báo (có sẵn khi bạn nhấp chuột phải trong cửa sổ đầu ra Cảnh báo)
- Menu ngữ cảnh của khu vực biểu đồ (có sẵn khi bạn nhấp chuột phải trong khu vực biểu đồ)
- Menu ngữ cảnh của bố cục (có sẵn khi bạn nhấp chuột phải trong cây Bố cục của không gian làm việc)
- Menu ngữ cảnh của công thức (có sẵn khi bạn nhấp chuột phải trong cây Biểu đồ của không gian làm việc)
- Menu ngữ cảnh của lớp (có sẵn khi bạn nhấp chuột phải trong danh sách Lớp của không gian làm việc)
- Menu ngữ cảnh của báo giá thời gian thực (có sẵn khi bạn nhấp chuột phải trong danh sách báo giá thời gian thực)

Menu Tệp



Mới

- **Cơ sở dữ liệu**
  Tạo một cơ sở dữ liệu AmiBroker mới và mở cửa sổ cài đặt cơ sở dữ liệu.
- **Biểu đồ Mặc định**
  Tạo cửa sổ biểu đồ mới sử dụng mẫu mặc định. Có thể chọn các ký hiệu và khung thời gian độc lập trong từng cửa sổ mở ra.
- **Biểu đồ Liên kết**
  Tạo cửa sổ biểu đồ liên kết dựa trên mẫu hiện tại và biểu đồ đang hoạt động. Các cửa sổ liên kết sử dụng cùng một lựa chọn ký hiệu, vì vậy nếu bạn thay đổi ký hiệu đã chọn cho một trong số đó, cái còn lại sẽ tự động đồng bộ. Các cửa sổ liên kết có thể có khung thời gian xem KHÁC NHAU được chọn. Chỉ cần kích hoạt cửa sổ và chọn khoảng thời gian mong muốn từ menu Xem cho một cửa sổ, sau đó chuyển sang cửa sổ khác và chọn khoảng thời gian khác cho nó. Tùy chọn này cho phép bạn chọn khung thời gian hoặc tập hợp chỉ báo khác nhau trong mỗi cửa sổ và dễ dàng di chuyển qua cơ sở dữ liệu.
- **Biểu đồ Trống**
  Tạo cửa sổ biểu đồ mới (trống). Điều này hữu ích nếu bạn muốn tạo hoàn toàn thiết lập mới cho các biểu đồ không chia sẻ cùng ID biểu đồ. Điều này quan trọng nếu bạn muốn có các chỉ báo với các tham số độc lập so với các cửa sổ khác mà bạn đã tạo.
- **Khu vực Trống**
  Tạo khu vực biểu đồ mới (trống).

- **Tài khoản**
  Tạo Tài khoản Mới (Trình quản lý Tài khoản).
- **Nghiên cứu Web**
  Tạo cửa sổ Nghiên cứu Web Mới.

Mở
Mở tài liệu (tài khoản, cơ sở dữ liệu hoặc tệp HTML - bạn có thể chọn loại tài liệu từ danh sách "Tệp loại" trong cửa sổ chọn tệp).

Đóng
Đóng cửa sổ tài liệu hiện tại (biểu đồ, tài khoản, nghiên cứu web).

Mở Cơ sở dữ liệu
Cho phép bạn mở một cơ sở dữ liệu AmiBroker hiện có. Vui lòng chọn thư mục cơ sở dữ liệu và nhấn OK.

Lưu Cơ sở dữ liệu
Lưu cơ sở dữ liệu đang được sử dụng hiện tại.

Lưu Cơ sở dữ liệu với tên khác...
Lưu cơ sở dữ liệu vào vị trí mới.

Lưu
Lưu tài liệu hiện tại (tài khoản, tệp HTML).

Lưu với tên khác...
Lưu tài liệu hiện tại (tài khoản, tệp HTML) với tên mới.

Lưu Tất cả
Lưu tất cả tài liệu hiện đang mở.

Cài đặt Cơ sở dữ liệu
Mở hộp thoại cài đặt cơ sở dữ liệu cho phép bạn thay đổi các tham số hoặc cài đặt intraday của cơ sở dữ liệu.

Trình hướng dẫn Nhập khẩu
Khởi động cửa sổ Trình hướng dẫn Nhập khẩu ASCII, cho phép bạn dễ dàng nhập các tệp ASCII (văn bản) vào cơ sở dữ liệu của mình.

Nhập khẩu ASCII
Cho phép bạn nhập các tệp ASCII bằng cách sử dụng các định dạng nhập khẩu đã định nghĩa trước. Để tìm hiểu thêm về cách sử dụng trình nhập khẩu ASCII, vui lòng đọc chương tham khảo về Trình nhập khẩu ASCII.

Nhập khẩu dữ liệu MetaStock
Khởi động cửa sổ trình nhập khẩu MetaStock. **LƯU Ý QUAN TRỌNG:** Trình nhập khẩu MetaStock chỉ nên được sử dụng NẾU bạn muốn nhập dữ liệu MS vào cơ sở dữ liệu AmiBroker cục bộ một lần. Nếu bạn muốn AmiBroker chỉ đọc cơ sở dữ liệu MetaStock TRỰC TIẾP mà không cần nhập dữ liệu mới lặp đi lặp lại, vui lòng thiết lập cơ sở dữ liệu của bạn VỚI PLUGIN METASTOCK như được mô tả trong Hướng dẫn.

In
Cho phép bạn in các biểu đồ đang hiển thị hiện tại.

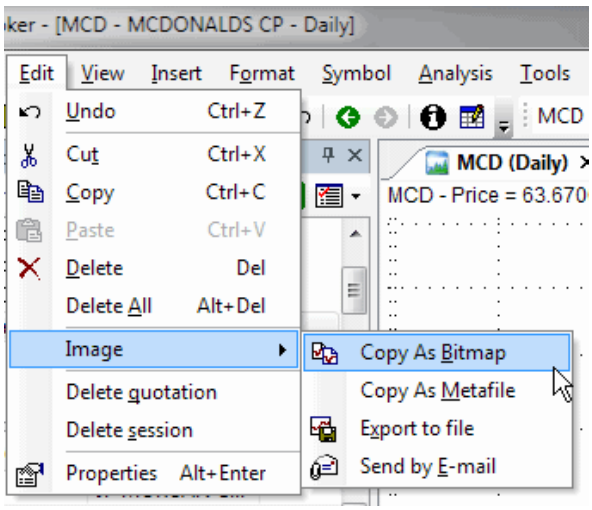
Xem Trước khi In
Xem trước các biểu đồ đang hiển thị hiện tại (bạn có thể kiểm tra sự xuất hiện của tài liệu trước khi in).

Cài đặt In
Mở hộp thoại cài đặt in.

Gửi Biểu đồ qua E-mail
AmiBroker tạo một hình ảnh .png (với biểu đồ hiện đang hiển thị) và sử dụng chương trình gửi mail mặc định của bạn (ví dụ: Outlook Express) để gửi tệp dưới dạng tệp đính kèm.

Thoát
Đóng chương trình AmiBroker.

Chỉnh sửa menu



Hoàn tác
Cho phép hoàn tác thao tác cuối cùng đã thực hiện trên các nghiên cứu biểu đồ (đường xu hướng, v.v.). Tùy chọn này sẽ không khả dụng nếu không có nghiên cứu nào được vẽ hoặc di chuyển.

Cắt, Sao chép, Dán, Xóa
Những tùy chọn này có thể được sử dụng để cắt, sao chép, dán hoặc xóa các nghiên cứu từ biểu đồ. Cắt, sao chép và xóa sẽ bị mờ nếu không có đối tượng nào trên biểu đồ được chọn. Để dán đối tượng, cần sử dụng tùy chọn 'sao chép' hoặc 'cắt' trước.
Để tìm hiểu thêm về các công cụ vẽ trong AmiBroker, vui lòng đọc chương tham khảo về công cụ vẽ.

Xóa Tất cả
Xóa tất cả các đối tượng khỏi cửa sổ biểu đồ đang mở.

Hình ảnh

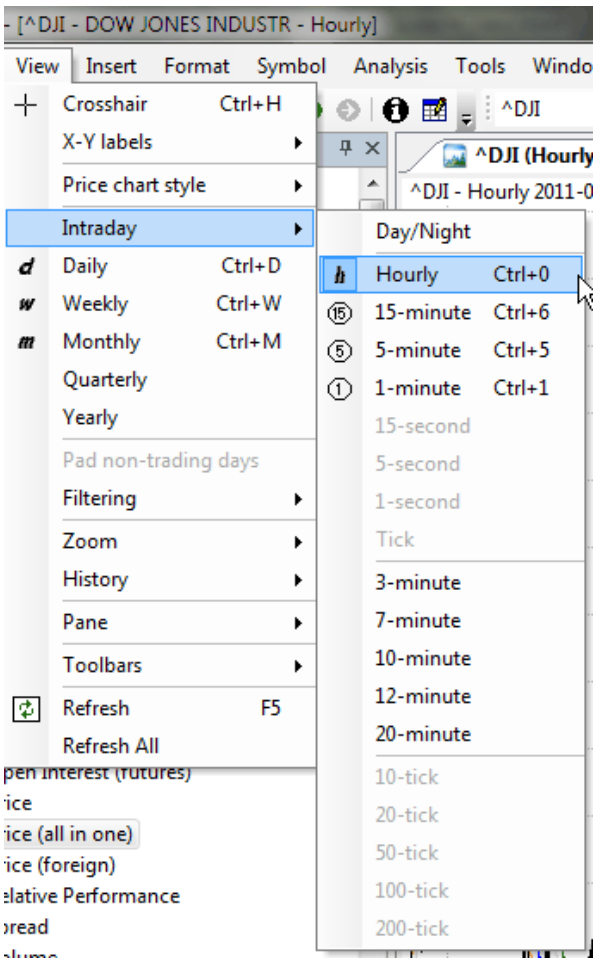
- **Sao chép Dưới dạng Bitmap** - sao chép biểu đồ hiện đang mở vào clipboard của hệ thống dưới dạng hình ảnh .BMP. Bạn có thể dán nội dung clipboard vào ứng dụng 'Paint', chẳng hạn.
- **Sao chép Dưới dạng Metafile** - sao chép biểu đồ hiện đang mở vào clipboard của hệ thống dưới dạng metafile.
- **Xuất ra tệp** - lưu biểu đồ hiện đang hiển thị dưới dạng tệp .PNG.
- **Gửi qua E-mail** - AmiBroker tạo một hình ảnh .png (với biểu đồ hiện đang hiển thị) và sử dụng chương trình gửi mail mặc định của bạn (ví dụ: Outlook Express) để gửi tệp dưới dạng tệp đính kèm.

Xóa báo giá
Xóa thanh hiện đang được chọn.

Xóa Phiên
Xóa thanh hiện đang được chọn từ TẤT CẢ các ký hiệu trong cơ sở dữ liệu.

Thuộc tính
Mở hộp thoại thuộc tính nghiên cứu.

Menu xem



Con trỏ chéo
Bật/tắt con trỏ chéo.

Nhãn X-Y
Điều khiển việc hiển thị các nhãn giá trị X-Y.

- **Tắt**
- **Chỉ với con trỏ chéo** - hiển thị nhãn giá trị X-Y khi con trỏ chéo được kích hoạt.
- **Luôn hiển thị** - luôn hiển thị nhãn giá trị X-Y.

Phong cách Biểu đồ Giá
Thay đổi kiểu của biểu đồ giá mặc định.

- **Tự động** - sử dụng cài đặt đã định nghĩa trong Công cụ -> Tùy chọn.
- **Đường** - biểu đồ đường.
- **Nến** - biểu đồ nến.
- **Cột** - biểu đồ cột truyền thống.

Intraday
Cho phép bạn chọn một trong các khoảng thời gian intraday và quyết định xem có hiển thị phiên ngày hay đêm hay không. Giờ của phiên ngày và đêm có thể được đặt trong: cửa sổ

cài đặt cơ sở dữ liệu (Tệp -> Cài đặt cơ sở dữ liệu -> Cài đặt intraday) hoặc riêng biệt cho nhóm trong cửa sổ Danh mục (Ký hiệu -> Danh mục).

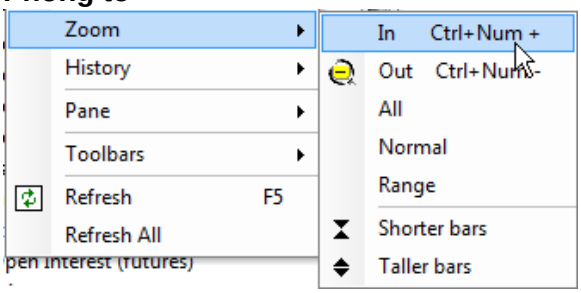
- **Ngày / Đêm** - hiển thị hai thanh (ngày và đêm) mỗi ngày.
- **Hiển thị giao dịch 24 giờ** - không áp dụng bộ lọc và tất cả dữ liệu trong cơ sở dữ liệu được đưa vào biểu đồ.
- **Chỉ hiển thị phiên ngày** - chỉ hiển thị phiên ngày.
- **Chỉ hiển thị phiên đêm** - chỉ hiển thị phiên đêm.
- **Hiển thị cả phiên ngày và đêm** - hiển thị cả phiên ngày và đêm.

Hàng ngày, Hàng tuần, Hàng tháng, Hàng quý, Hàng năm
Cho phép thay đổi khoảng thời gian hiển thị.

Đệm cho các ngày không giao dịch
Bật đệm cho các ngày thứ Bảy, Chủ Nhật và các ngày không giao dịch khác với giá đóng cửa của phiên trước.

Lọc
Cho phép chọn giữa không lọc (hiển thị giao dịch 24 giờ), chỉ giao dịch thông thường, hoặc chỉ giao dịch mở rộng.

Phóng to

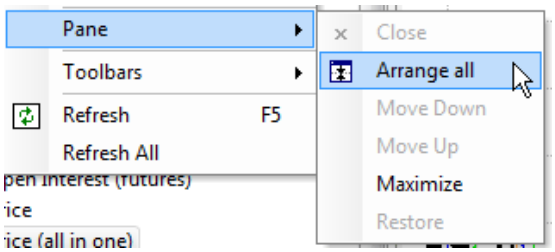


- Điều khiển mức độ phóng to của biểu đồ.
- **Vào** - giảm số lượng thanh hiển thị.
- **Ra** - tăng số lượng thanh hiển thị.
- **Tất cả** - hiển thị tất cả các thanh có sẵn cho ký hiệu hiện tại.
- **Bình thường** - hiển thị số lượng thanh mặc định (được định nghĩa trong Công cụ -> Tùy chọn -> Biểu đồ).
- **Phạm vi** - hiển thị các thanh trong phạm vi đã chọn.
- **Thanh ngắn hơn** - giảm kích thước chiều dọc của các thanh.
- **Thanh dài hơn** - tăng kích thước chiều dọc của các thanh.

Lịch sử
Cho phép di chuyển Lùi/Tiến theo cách giống như trình duyệt.

- **Trước** - di chuyển đến ký hiệu trước (phím tắt: Ctrl+Alt+LEFT).
- **Tiếp theo** - di chuyển đến ký hiệu tiếp theo (phím tắt: Ctrl+Alt+RIGHT).

Khu vực



- **Đóng** - đóng khu vực biểu đồ hiện đang được chọn.
- **Sắp xếp Tất cả** - sắp xếp tất cả các biểu đồ đang hiển thị.
- **Di chuyển Xuống** - di chuyển khu vực biểu đồ hiện đang được chọn xuống một vị trí.
- **Di chuyển Lên** - di chuyển khu vực biểu đồ hiện đang được chọn lên một vị trí.
- **Tối đa hóa** - tối đa hóa khu vực biểu đồ hiện đang được chọn.
- **Khôi phục** - khôi phục bố cục biểu đồ sau khi sử dụng Tối đa hóa.

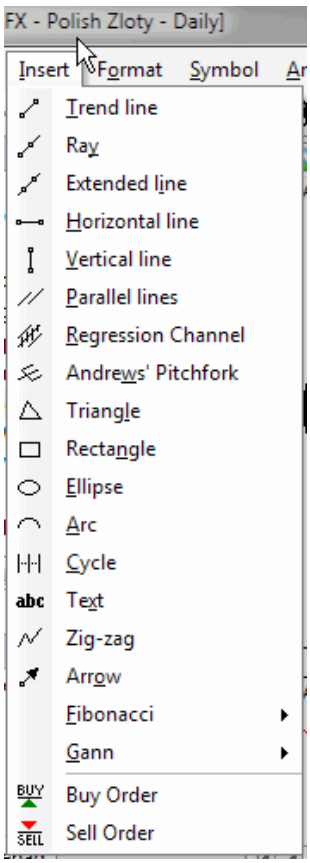
Thanh công cụ
Cho phép bạn hiển thị/ẩn các thanh công cụ.

Làm mới
Làm mới cửa sổ biểu đồ.

Làm mới Tất cả
Làm mới cửa sổ biểu đồ và đọc lại nội dung của tất cả các danh mục trong cây ký hiệu trong cửa sổ Khu vực làm việc.

Lưu ý cho người dùng các phiên bản trước: Các mục điều khiển Ký hiệu, Bố cục, Lớp, Biểu đồ, Thông tin và các cửa sổ khác đã được chuyển đến menu Cửa sổ

Chèn menu



Đường xu hướng
Vẽ một đường xu hướng.
Để vẽ đường xu hướng trên biểu đồ, bắt đầu bằng cách di chuyển chuột đến vị trí bạn muốn bắt đầu vẽ và nhấn nút chuột trái. Sau đó, di chuyển chuột và đường theo dõi sẽ xuất hiện. Nhả nút chuột trái khi bạn muốn kết thúc việc vẽ. Ngoài ra, bạn có thể nhấp một lần vào vị trí mà bạn muốn bắt đầu vẽ đường xu hướng, di chuyển chuột và nhấp một lần nữa để hoàn thành việc vẽ. Bạn cũng có thể hủy việc vẽ bằng cách nhấn phím ESC.

Đường tia
Vẽ một đường tia. Đường tia là một đường xu hướng được kéo dài sang bên phải.

Đường kéo dài
Vẽ một đường kéo dài. Đường kéo dài là một đường xu hướng được tự động mở rộng từ cả hai bên trái và phải.

Đường ngang
Vẽ một đường ngang. Đường ngang tự mở rộng, vì vậy chỉ cần nhấp vào mức giá đã chọn.

Đường dọc
Vẽ một đường dọc. Đường dọc tự mở rộng, vì vậy chỉ cần nhấp vào thanh đã chọn.

Đường song song
Vẽ các đường xu hướng song song.
Công cụ này cho phép vẽ một chuỗi các đoạn đường xu hướng song song. Đầu tiên, bạn vẽ

một đường xu hướng như bình thường, sau đó một đường song song với đường đầu tiên sẽ được tự động tạo ra và bạn có thể di chuyển chúng bằng chuột. Khi bạn nhấp vào biểu đồ, nó sẽ được đặt ở vị trí đã cho. Sau đó, một đường song song khác sẽ xuất hiện và có thể được đặt ở một vị trí khác. Bạn có thể tiếp tục như vậy. Để dừng lại, hãy nhấn phím ESC hoặc chọn công cụ "Chọn".

Kênh hồi quy
Vẽ các kênh Raff, độ lệch chuẩn, độ sai số chuẩn. Để biết thông tin chi tiết về công cụ này, hãy đọc chương tham khảo công cụ vẽ.

Công cụ Andrews' pitchfork
Vẽ một công cụ Andrews' pitchfork. Đọc chương tham khảo công cụ vẽ để biết thêm thông tin chi tiết.

Tam giác
Vẽ một tam giác. Nhấp chuột trái vào điểm đầu tiên, di chuyển đến điểm thứ hai rồi nhấp một lần nữa, sau đó di chuyển đến điểm thứ ba và nhấp một lần nữa.

Hình chữ nhật
Vẽ một hình chữ nhật. Nhấp chuột trái vào điểm đầu tiên, di chuyển đến vị trí bạn muốn đặt góc đối diện và nhấp một lần nữa.

Elip
Vẽ một hình elip. Hình elip được kết nối với tọa độ ngày/giá (giống như đường xu hướng) thay vì với các điểm ảnh trên màn hình, vì vậy nó có thể thay đổi hình dạng trực quan khi hiển thị ở các mức phóng to hoặc kích thước màn hình khác nhau. Để xem thuộc tính của hình elip, bạn nên nhấp đúp vào các vị trí giờ 3, 6, 9 hoặc 12.

Cung
Vẽ một cung. Cung, giống như hình elip, được kết nối với tọa độ ngày/giá (giống như đường xu hướng) thay vì với các điểm ảnh trên màn hình, vì vậy nó có thể thay đổi hình dạng trực quan khi hiển thị ở các mức phóng to hoặc kích thước màn hình khác nhau. Để xem thuộc tính của hình elip, bạn nên nhấp đúp vào các vị trí giờ 3, 6, 9 hoặc 12.

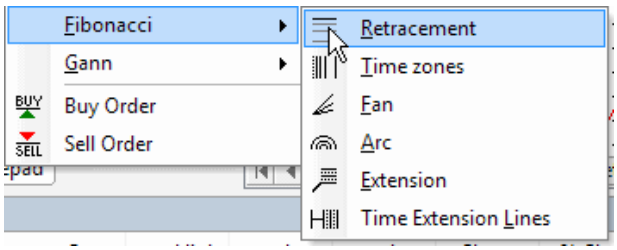
Chu kỳ
Vẽ các chu kỳ thời gian. Để sử dụng công cụ chu kỳ thời gian, nhấp vào nút công cụ vẽ chu kỳ trên thanh công cụ, sau đó nhấp vào điểm bắt đầu của chu kỳ và kéo đến điểm kết thúc của chu kỳ. Hai điểm điều khiển này kiểm soát khoảng cách giữa các đường chu kỳ. Khi bạn nhả nút chuột, bạn sẽ có một chuỗi các đường song song với khoảng cách bằng nhau giữa chúng.

Văn bản
Cho phép đặt văn bản tùy chỉnh trên biểu đồ. Nhấp chuột trái vào biểu đồ để bắt đầu gõ. Để hoàn tất - nhấp một lần nữa vào biểu đồ, bên ngoài ô văn bản. Bạn cũng có thể hủy việc gõ bằng cách nhấn phím ESC.

Zig-zag
Vẽ một chuỗi các đường xu hướng nối tiếp. Để hoàn tất chuỗi, hãy nhấp đúp hoặc nhấn phím ESC.

Mũi tên
Vẽ một đường kết thúc bằng một mũi tên. Kỹ thuật vẽ hoàn toàn giống như vẽ một đường xu hướng.

Fibonacci
Nhóm các công cụ vẽ Fibonacci. Đọc chương tham khảo công cụ vẽ để biết thêm thông tin chi tiết.



- Nghiên cứu Fibonacci Retracement
- Nghiên cứu Fibonacci Time Zones
- Fibonacci Fan
- Fibonacci Arc
- Fibonacci Extensions
- Fibonacci Time Extension Lines

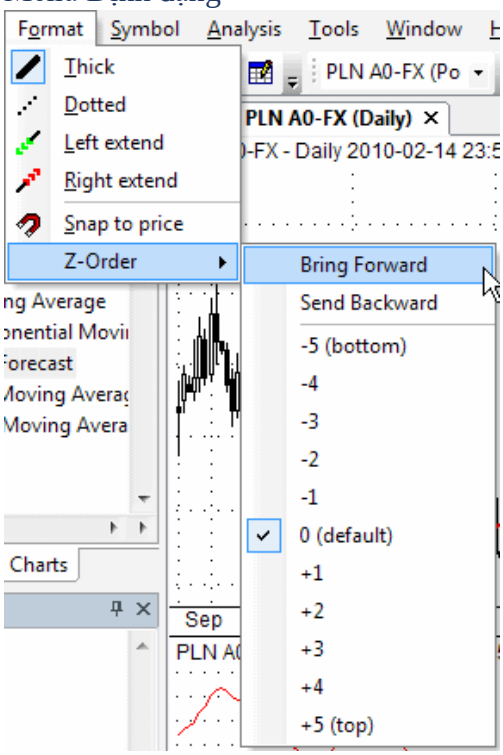
Gann
Nhóm các công cụ vẽ Gann.

Gann Fan

Gann Square

Đọc chương tham khảo công cụ vẽ để biết thêm thông tin chi tiết.

Menu Định dạng



Những tùy chọn này cho phép bạn áp dụng màu sắc hoặc kiểu dáng cho các đối tượng. Lưu ý rằng bạn cũng có thể chọn màu sắc và kiểu dáng của đối tượng trước khi vẽ đối tượng mới: chỉ cần bỏ chọn đối tượng trước đó (nếu có), thay đổi các lựa chọn kiểu dáng và vẽ đối tượng mới.

Đậm
Thay đổi định dạng của đối tượng đã vẽ thành kiểu đậm.

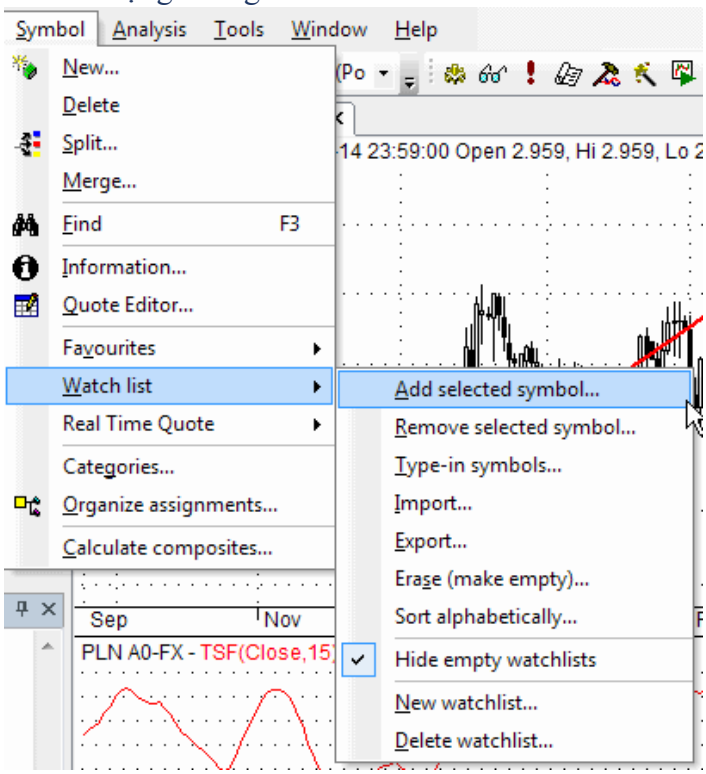
Chấm
Thay đổi định dạng của đối tượng thành kiểu chấm.

Mở rộng trái
Kéo dài đường xu hướng sang bên trái.

Mở rộng phải
Kéo dài đường xu hướng sang bên phải.

Gắn vào giá
Bật chế độ nam châm, giúp các nghiên cứu đã vẽ gắn vào giá. Ngưỡng % gắn vào giá có thể được thiết lập trong cửa sổ Tùy chọn. Ngưỡng % gắn vào giá xác định khoảng cách mà "nam châm" giá hoạt động, nó sẽ gắn vào giá khi chuột gần hơn ngưỡng % từ giá H/L/C

Menu Tượng Trưng



Mới

Cho phép bạn thêm các ký hiệu mới vào cơ sở dữ liệu. Sau khi chọn chức năng này, bạn sẽ được nhắc nhập ký hiệu mới. Hãy cố gắng không vượt quá 26 ký tự. Để việc nhập khẩu hoạt động đúng, bạn nên nhập ký hiệu bằng CHỮ HOA.

Xóa

Xóa ký hiệu hiện tại được chọn khỏi cơ sở dữ liệu. Sau khi chọn chức năng này, bạn sẽ được yêu cầu xác nhận việc xóa ký hiệu. Lưu ý rằng thao tác này không thể hoàn tác.

Chia tách

Cho phép bạn thực hiện việc chia tách cổ phiếu. AmiBroker cung cấp cách dễ dàng để xử lý các cuộc chia tách cổ phiếu. Chương trình sẽ cố gắng đoán ngày chia tách và tỷ lệ bằng cách phân tích các báo giá. Nếu có chỉ một báo giá sau khi chia tách, điều này sẽ hoạt động, nếu không bạn sẽ được yêu cầu nhập ngày chia tách và tỷ lệ. Bạn có thể xác định một cuộc chia tách bằng cách sử dụng biểu thức sau: x->y, có nghĩa là x cổ phiếu trước khi chia tách trở thành y sau đó. Ví dụ, 2->3 có nghĩa là 2 cổ phiếu trở thành 3 sau khi chia tách. Cũng có thể thực hiện chia tách ngược, ví dụ, 2->1, có nghĩa là 2 cổ phiếu được gộp thành 1 cổ phiếu.

Gộp

Chức năng này cho phép bạn gộp hai ký hiệu, khi ký hiệu của biểu tượng được thay đổi và trong cơ sở dữ liệu của bạn - một ký hiệu giữ các báo giá lịch sử và ký hiệu thứ hai giữ các báo giá mới nhất (sau khi thay đổi tên). Bạn chỉ cần chọn ký hiệu mới (sau khi thay đổi tên)

và sử dụng Symbol->Merge. Sau đó từ danh sách kết hợp, bạn nên chọn ký hiệu gốc ("merge with") và tùy chọn đánh dấu các trường sau:

- Ghi đè báo giá trùng lặp - đánh dấu tùy chọn này sẽ ghi đè các báo giá đã có trong ký hiệu "mới" bằng những báo giá hiện có trong ký hiệu "cũ" (điều này thực sự không nên xảy ra, nhưng có thể xảy ra).
- Xóa "merge with" sau - đánh dấu tùy chọn này sẽ xóa ký hiệu "cũ" sau khi gộp.
- Gán tên bí danh - đánh dấu tùy chọn này sẽ sao chép ký hiệu "cũ" vào trường bí danh của ký hiệu "mới".

Tìm

Mở cửa sổ tìm kiếm ký hiệu cho phép bạn nhanh chóng tìm kiếm cơ sở dữ liệu cho một ký hiệu bằng cách gõ những chữ cái đầu tiên của tên đầy đủ hoặc ký hiệu của nó.

Thông tin

Mở cửa sổ thông tin cho ký hiệu, cho phép bạn thay đổi thuộc tính của ký hiệu.

Tài chính

Cửa sổ tài chính cho phép bạn nhập một số dữ liệu cơ bản cho ký hiệu (doanh thu, thu nhập trước thuế (EBT), thu nhập sau thuế (EAT)). AmiBroker sẽ tính toán các chỉ số P/E (Tỷ lệ Giá trên Thu nhập) và EPS (Lợi nhuận trên Cổ phiếu) dựa trên dữ liệu đã nhập.

Trình chỉnh sửa báo giá

Mở cửa sổ Trình chỉnh sửa báo giá cho phép bạn chỉnh sửa, xóa và thêm báo giá vào cơ sở dữ liệu của bạn.

Danh sách theo dõi

Những tùy chọn này cho phép bạn quản lý danh sách theo dõi của mình. Chương trình làm việc với danh sách theo dõi giải thích chi tiết hơn cách bạn có thể sử dụng các tùy chọn dưới đây.

- Thêm Ký hiệu Đã Chọn - thêm ký hiệu hiện tại được chọn vào danh sách theo dõi chỉ định.
- Xóa Ký hiệu Đã Chọn - xóa ký hiệu hiện tại được chọn khỏi danh sách theo dõi chỉ định.
- Nhập Ký hiệu - cho phép bạn nhập các ký hiệu vào danh sách theo dõi.
- Nhập khẩu - cho phép nhập khẩu danh sách theo dõi từ tệp .TLS.
- Xuất khẩu - xuất các ký hiệu thuộc danh sách theo dõi sang tệp .TLS (danh sách ký hiệu).
- Xóa (làm trống) - xóa tất cả các ký hiệu khỏi danh sách theo dõi chỉ định.
- Sắp xếp theo thứ tự chữ cái - sắp xếp các ký hiệu theo thứ tự chữ cái trong danh sách theo dõi chỉ định.
- Ẩn danh sách theo dõi trống - ẩn các danh sách theo dõi không có ký hiệu trong cây ký hiệu.
- Danh sách theo dõi mới - tạo danh sách theo dõi mới.

- Xóa danh sách theo dõi - xóa danh sách theo dõi đã chọn (nó không xóa ký hiệu khỏi cơ sở dữ liệu).

Danh mục

Cửa sổ danh mục cho phép bạn định nghĩa tên của các thị trường, nhóm, lĩnh vực và ngành nghề. Đối với mỗi thị trường, bạn cũng có thể xác định các chỉ số cơ sở để tính toán sức mạnh tương đối, dữ liệu tổng hợp, beta hoặc URL hồ sơ web.

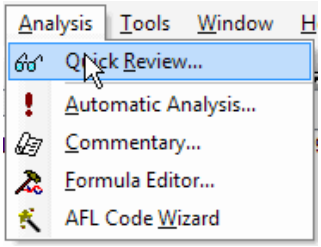
Tổ chức phân công

Cửa sổ tổ chức phân công cho phép bạn dễ dàng thay đổi phân công danh mục cho các ký hiệu hoặc xóa nhiều ký hiệu khỏi cơ sở dữ liệu.

Tính toán Tổng hợp

Mở cửa sổ tính toán tổng hợp cho phép tính toán tự động số lượng và khối lượng các vấn đề đang tăng/giảm/không thay đổi hoặc số liệu khối lượng cho các chỉ số.

Menu Phân Tích



Xem Nhanh

Mở cửa sổ xem nhanh cung cấp thông tin tổng quan về thị trường như: báo giá ký hiệu hàng ngày, bảng so sánh lợi tức hàng ngày/tuần/tháng/quý/năm hoặc so sánh giá trên thu nhập và giá trên giá trị sổ sách.

Phân Tích Tự Động

Mở cửa sổ phân tích tự động cho phép bạn kiểm tra các báo giá của mình theo các quy tắc mua/bán đã định nghĩa hoặc khám phá cơ sở dữ liệu của bạn. AmiBroker có thể tạo báo cáo cho biết liệu các tín hiệu mua/bán đã xảy ra trên ký hiệu nhất định trong khoảng thời gian xác định hay không, mô phỏng giao dịch, cung cấp cho bạn một ý tưởng về hiệu suất của hệ thống của bạn hoặc tối ưu hóa hệ thống giao dịch mà bạn đang sử dụng để cải thiện hiệu suất của nó.

Bình luận

Hiển thị cửa sổ bình luận cho phép bạn xem các mô tả bằng văn bản về tình hình kỹ thuật hiện tại trên thị trường nhất định.

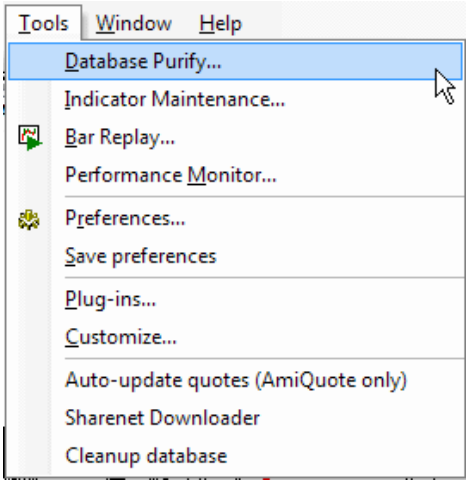
Trình chỉnh sửa Công thức

Mở cửa sổ Trình chỉnh sửa công thức cho phép bạn viết các công thức của riêng mình.

Trợ lý Mã AFL

Mở Trợ lý Mã AFL - chương trình bổ sung tạo mã hệ thống giao dịch AFL từ các câu tiếng Anh đơn giản. Xem video giới thiệu về Trợ lý Mã AFL.

Menu Công Cụ



Làm sạch Cơ sở Dữ liệu

Công cụ làm sạch cơ sở dữ liệu cho phép phát hiện các báo giá thiếu/thừa, các cuộc chia tách có thể xảy ra hoặc mối quan hệ OHLC không hợp lệ.

Bảo trì Chỉ báo

Mở trình hướng dẫn bảo trì chỉ báo, giúp bạn dọn dẹp không gian chỉ báo không sử dụng.

Phát lại Thanh

Mở công cụ phát lại thanh, cho phép phát lại dữ liệu lịch sử.

Tùy chọn

Mở cửa sổ tùy chọn cho phép bạn cấu hình chương trình.

Lưu Tùy chọn

Lưu tất cả các thay đổi tùy chọn (thông tin được lưu trong tệp broker.prefs).

Plugin

Mở cửa sổ plugin. Nó chứa danh sách tất cả các DLL plugin đã tải và có thể được sử dụng để kiểm tra những plugin nào đang hoạt động. Cũng có thể tải xuống các plugin.

Tùy chỉnh

Hộp thoại tùy chỉnh công cụ cho phép bạn định nghĩa các công cụ tùy chỉnh có thể được gọi từ menu Công cụ.

Cập nhật tự động báo giá

Tùy chọn cập nhật tự động báo giá cập nhật báo giá lịch sử từ ngày cuối cùng có mặt trong AmiBroker đến hôm nay bằng cách sử dụng AmiQuote Downloader. Mô tả chi tiết về cách sử dụng AmiQuote để lấy báo giá miễn phí có thể được tìm thấy trong chương trình hướng dẫn cập nhật tự động báo giá EOD.

Trình tải Sharenet

Khởi động tập lệnh tải báo giá từ Sharenet (chỉ tại Nam Phi).

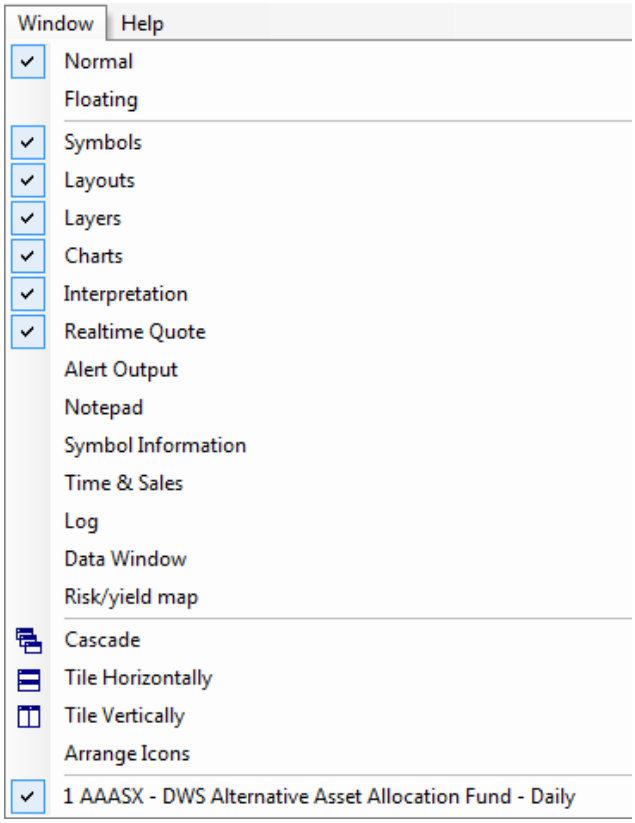
Xuất sang tệp CSV

Chạy một kịch bản để xuất cơ sở dữ liệu ra tệp CSV. Lưu ý rằng bạn có thể sử dụng cửa sổ Phân tích Tự động để xuất các báo giá nhanh hơn nhiều so với việc sử dụng kịch bản này.

Dọn dẹp cơ sở dữ liệu

Khởi chạy kịch bản cho phép bạn tìm các cổ phiếu không giao dịch trong cơ sở dữ liệu. Kịch bản sẽ tự động quét cơ sở dữ liệu và kiểm tra ngày báo giá mới nhất. Nếu ngày này quá cũ, kịch bản sẽ hiển thị thông báo cảnh báo và cho phép bạn quyết định xem cổ phiếu đó có nên bị xóa hay không. Ngoài ra, kịch bản còn có thể tạo danh sách các cổ phiếu "cũ" và lưu vào tệp văn bản. Thông tin chi tiết có sẵn trong số bản tin tháng 05-2000.

Cửa sổ menu



LƯU Ý QUAN TRỌNG cho người dùng phiên bản cũ: Các tùy chọn Window -> New và Window -> New Linked đã được chuyển sang menu File->New->Default Chart và File->New->Linked Chart.

Tab Symbols - cây biểu tượng với các danh mục (Xem: Hiểu về các danh mục).

Tab Layouts - danh sách các bố cục toàn cục và địa phương có sẵn (Xem: Làm việc với bảng biểu đồ và bố cục cửa sổ).

Tab Layers - danh sách các lớp biểu đồ (Xem: Làm việc với các lớp).

Tab Charts - cửa sổ hiển thị danh sách công thức biểu đồ (Xem: Làm việc với giao diện kéo-thả biểu đồ).

Interpretation
Hiển thị/ẩn cửa sổ Giải thích.

Realtime Quote
Hiển thị/ẩn cửa sổ Báo giá Thời gian thực. Cửa sổ báo giá RT cung cấp báo giá trực tiếp và một số dữ liệu cơ bản. Để tìm hiểu thêm, đọc: Cách sử dụng AmiBroker ở chế độ Thời gian thực.

Alert Output
Hiển thị/ẩn cửa sổ Đầu ra Cảnh báo. Cửa sổ này hiển thị văn bản được tạo ra bởi cảnh báo dựa trên công thức. Thông tin chi tiết về cách sử dụng cảnh báo có sẵn trong: Sử dụng cảnh báo dựa trên công thức trong Hướng dẫn người dùng.

Notepad
Hiển thị/ẩn cửa sổ Ghi chú, cho phép lưu trữ ghi chú tự do về một chứng khoán cụ thể. Chỉ cần nhập bất kỳ văn bản nào và nó sẽ được tự động lưu/truy xuất khi bạn duyệt qua các biểu tượng. Ghi chú là toàn cầu và được lưu trong thư mục con "Notes" dưới dạng các tệp văn bản thông thường.

Thông tin Biểu tượng
Hiển thị cửa sổ thông tin biểu tượng với dữ liệu cơ bản.

Time & Sales
Hiển thị cửa sổ Thời gian và Doanh số thời gian thực.

Log
Hiển thị cửa sổ nhật ký hiển thị thông báo lỗi AFL, lỗi thời gian chạy và đầu ra _TRACE.

Cửa sổ Dữ liệu
Hiển thị cửa sổ dữ liệu hiển thị các giá trị của các chỉ báo biểu đồ.

Bản đồ Rủi ro/Lợi nhuận
Hiển thị bản đồ Rủi ro/Lợi nhuận của tất cả các biểu tượng trong cơ sở dữ liệu. Bản đồ rủi ro/lợi nhuận tính toán lợi suất trung bình hàng tuần (lợi nhuận) và độ lệch chuẩn của lợi suất hàng tuần (rủi ro) trong ít nhất 12 tuần. Nó yêu cầu ít nhất 60 thanh dữ liệu cho mỗi cổ phiếu. Để phóng to - đánh dấu khu vực bằng chuột. Để thu nhỏ, chỉ cần nhấp vào bản đồ.

Cascade
Xếp chồng các cửa sổ biểu đồ đã mở.

Tile Horizontally
Xếp các cửa sổ biểu đồ đã mở theo chiều ngang.

Tile Vertically
Xếp các cửa sổ biểu đồ đã mở theo chiều dọc.

Normal
Chuyển cửa sổ biểu đồ về trạng thái "bình thường" (không nổi). Thông tin thêm tại đây.

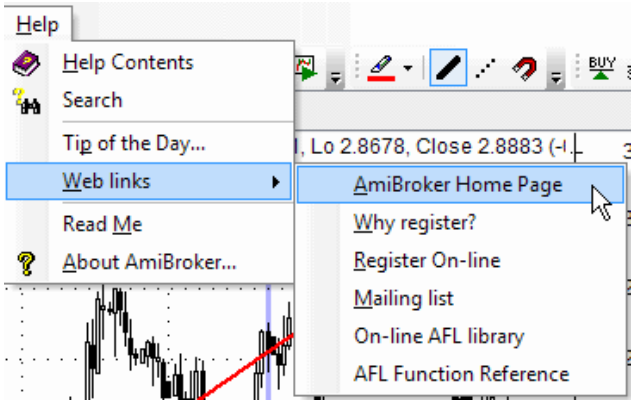
Floating
Chuyển cửa sổ biểu đồ về trạng thái nổi. Thông tin thêm tại đây.

Arrange Icons
Cho phép bạn sắp xếp các cửa sổ đã thu nhỏ. Tùy chọn sắp xếp chỉ hoạt động nếu:

• Bạn đã tạo hơn hai cửa sổ (thông qua Window->New hoặc Window->New Linked)
• Bạn đã thu nhỏ chúng
• Bạn đã di chuyển các hộp thu nhỏ

Tùy chọn sắp xếp biểu tượng sẽ căn chỉnh các cửa sổ một cách đẹp mắt ở dưới cùng của cửa sổ AmiBroker.

Menu Trợ Giúp



Nội dung Trợ Giúp
Hiển thị nội dung của Hướng dẫn Người dùng AmiBroker.

Tìm kiếm
Cho phép bạn tìm kiếm trong Hướng dẫn Người dùng.

Mẹo hàng ngày
Hiển thị hộp thoại Mẹo hàng ngày, nơi nhiều mẹo sử dụng hữu ích được hiển thị.

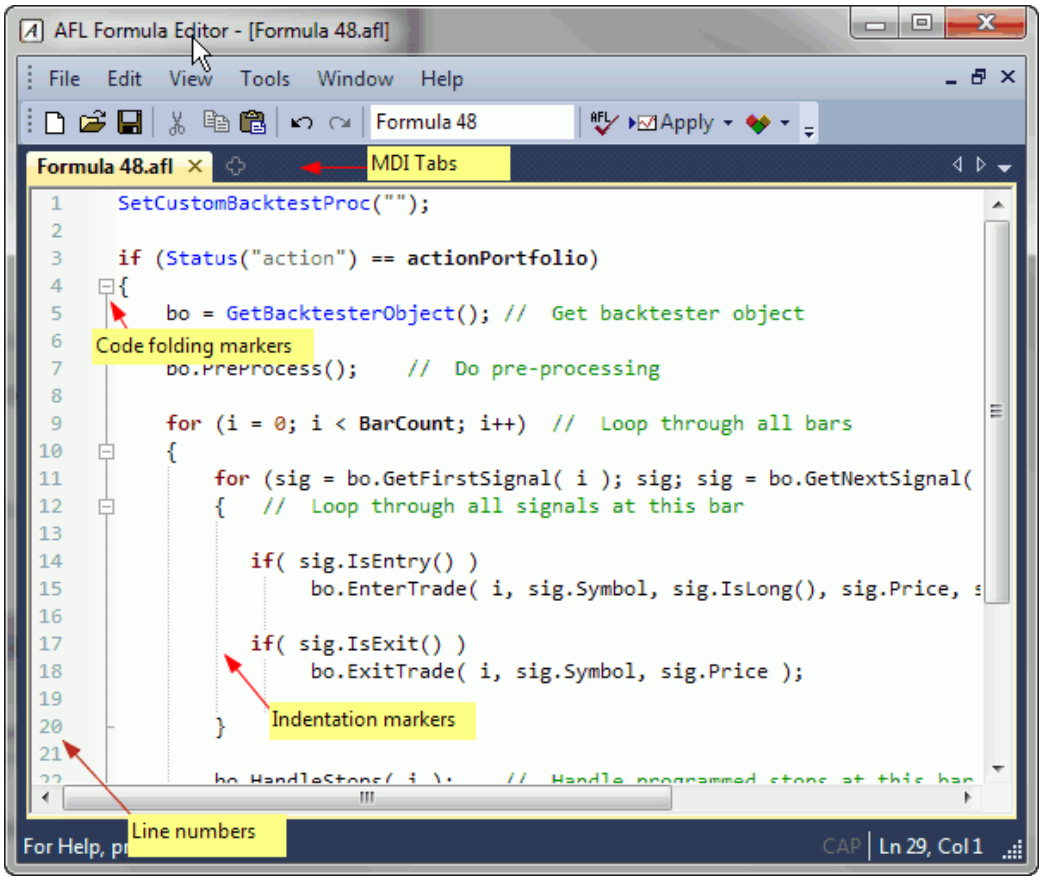
AmiBroker trên web

• Liên kết đến trang chính của AmiBroker
• Danh sách các lợi ích cho người dùng đã đăng ký
• Mẫu đơn đặt hàng trực tuyến an toàn
• Danh sách gửi thư AmiBroker
• Thư viện công thức trực tuyến
• Tài liệu tham khảo chức năng AFL trực tuyến

Readme
Hiển thị nội dung của tệp Readme. Xin lưu ý rằng tất cả các thay đổi gần đây trong các phiên bản beta được báo cáo trong Readme.

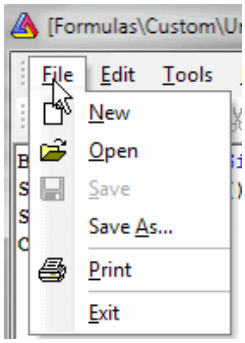
Giới thiệu về AmiBroker
Hiển thị cửa sổ "Giới thiệu", chứa thông tin về phiên bản chương trình và chi tiết người dùng.

Menu Biên tập AFL



Trình biên tập AFL có menu riêng với các lựa chọn sau:

1. File



New - Xóa cửa sổ biên tập công thức.

Open - Mở tệp công thức.

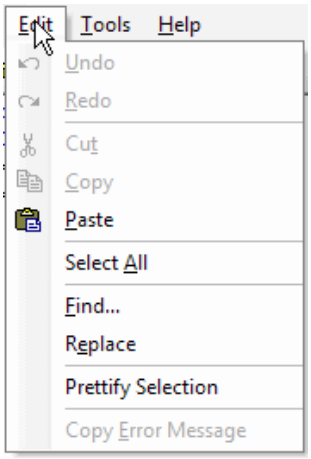
Save - Lưu công thức với tên hiện tại.

Save As.. - Lưu công thức với tên mới.

Print - In công thức.

Exit - Đóng trình biên tập.

2. Edit



Undo - Hoàn tác hành động gần đây (đa cấp độ).

Redo - Làm lại hành động gần đây (đa cấp độ).

Cut - Cắt phần đã chọn và sao chép vào clipboard.

Copy - Sao chép phần đã chọn vào clipboard.

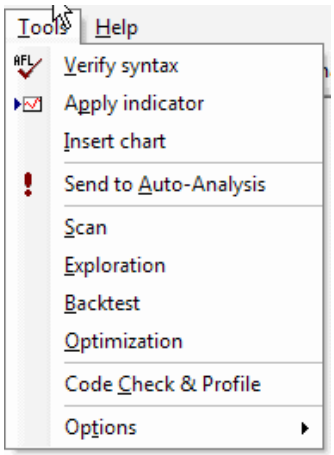
Paste - Dán nội dung hiện tại của clipboard tại vị trí con trỏ hiện tại.

Select All - Chọn toàn bộ văn bản trong trình biên tập.

Find... - Cung cấp truy cập vào công cụ tìm kiếm văn bản.

Copy Error Message - Sao chép thông báo lỗi hiện tại hiển thị ở dưới cùng của cửa sổ biên tập vào clipboard (tùy chọn này chỉ hoạt động khi có lỗi hiển thị sau khi kiểm tra cú pháp).

3. Tools



Verify syntax - Kiểm tra công thức hiện tại để phát hiện lỗi.

Apply indicator - Lưu công thức và áp dụng công thức hiện tại như một biểu đồ/ chỉ báo MỘT LẦN.

Insert chart - Lưu công thức và áp dụng công thức hiện tại như một biểu đồ NHIỀU LẦN (chèn nhiều lần).

Send to Auto-Analysis - Lưu công thức và chọn nó làm công thức hiện tại trong cửa sổ Phân tích Tự động.

Scan - Lưu công thức và thực hiện Quét trong cửa sổ Phân tích Tự động.

Exploration - Lưu công thức và thực hiện Khám phá trong cửa sổ Phân tích Tự động.

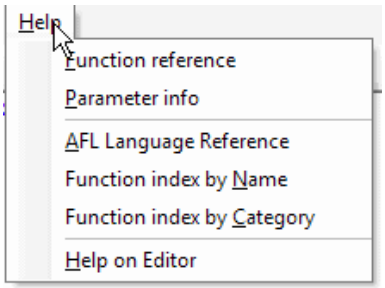
Backtest - Lưu công thức và thực hiện Kiểm tra lại trong cửa sổ Phân tích Tự động.

Optimization - Lưu công thức và thực hiện Tối ưu hóa trong cửa sổ Phân tích Tự động.

Check - Lưu công thức và thực hiện Kiểm tra (nếu công thức đã cho tham chiếu đến tương lai) trong cửa sổ Phân tích Tự động.

Options: Auto-save formula before running analysis - Khi được đánh dấu, bất kỳ nhấp chuột nào vào nút Quét/Kham phá/Kiểm tra lại/Tối ưu hóa trong cửa sổ Phân tích Tự động sẽ kích hoạt tự động lưu công thức hiện tại.

4. Help



Function reference - Hiển thị trang tham khảo cho chức năng AFL hiện đang được đánh dấu.

Parameter info - Hiển thị thông tin tham số cho chức năng AFL hiện đang được đánh dấu.

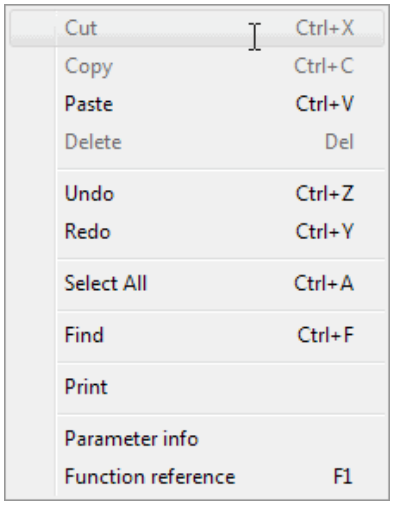
AFL Language reference - Hiển thị trang tham khảo ngôn ngữ.

Function index by Name - Hiển thị danh sách theo thứ tự bảng chữ cái các chức năng AFL.

Function index by Category - Hiển thị danh sách phân loại các chức năng AFL.

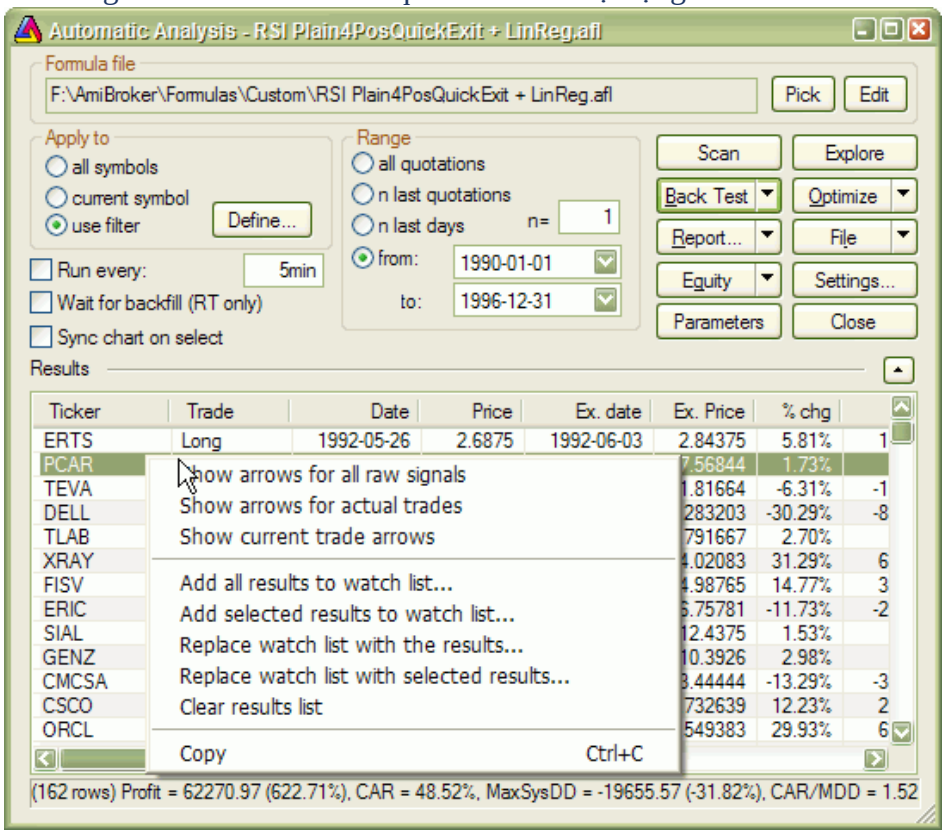
Help on Editor - Hiển thị trang trợ giúp này.

Cũng như menu ngữ cảnh (có sẵn qua nhấp chuột PHẢI vào công thức):



Menu ngữ cảnh này chủ yếu sao chép các lựa chọn có sẵn từ menu thông thường.

Menu ngữ cảnh danh sách kết quả Phân tích Tự động



Menu này xuất hiện khi bạn nhấp chuột PHẢI vào danh sách kết quả Phân tích Tự động.

Các lựa chọn có sẵn:

- **Show arrows for all raw signals** - Hiển thị các mũi tên mua/bán/ngắn/đóng cho tất cả các tín hiệu thô (không lọc). Nếu công thức của bạn là:

  ```
  buy = C > MA( C, 10 );
  ```

bạn sẽ thấy mũi tên mua (xanh đặc) cho tất cả các thanh nơi giá đóng cửa cao hơn trung bình động 10 kỳ.

- **Show arrows for actual trades** - Hiển thị mũi tên chỉ trên các thanh vào/ra giao dịch. Điều này cho thấy mũi tên cho TẤT CẢ CÁC GIAO DỊCH. Nếu công thức của bạn là:
  ```
  buy = C > MA( C, 10 );
  ```

bạn sẽ nhận được một mũi tên mua (xanh đặc) chỉ cho thanh đầu tiên khi giá đóng cửa vượt lên trên trung bình động và giao dịch được khởi tạo, và bạn sẽ không nhận được bất kỳ mũi tên mua tiếp theo nào cho đến khi có một mũi tên bán (đóng giao dịch) tương ứng.

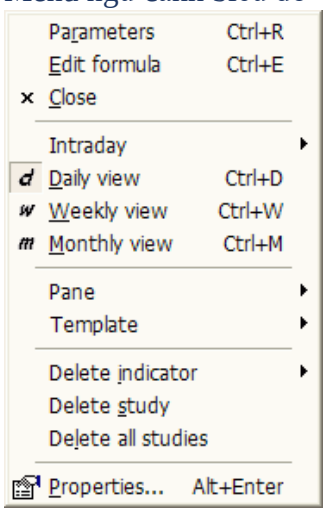
Lưu ý rằng các mũi tên giao dịch đại diện cho tất cả các giao dịch có thể xảy ra. Giao dịch đã cho có thể không được thực hiện bởi trình kiểm tra lại nếu không có đủ tiền để vào.

- **Show current trade arrows** - Hiển thị mũi tên vào/ra cho giao dịch đã chọn. Điều này hiển thị mũi tên cho giao dịch hiện tại được chọn (từ danh sách kết quả). Nó đại diện cho giao dịch thực tế đã thực hiện.
- **Add all results to watch list** - Thêm tất cả các biểu tượng từ danh sách kết quả vào danh sách theo dõi của bạn. Tìm hiểu thêm về điều này tại đây.
- **Add selected results to watch list** - Thêm các biểu tượng từ các dòng đã chọn vào danh sách theo dõi của bạn. Tìm hiểu thêm về điều này tại đây.
- **Replace watch list with all results** - Xóa danh sách theo dõi và sau đó thêm tất cả các biểu tượng từ danh sách kết quả vào danh sách theo dõi của bạn. Tìm hiểu thêm về điều này tại đây.
- **Replace watch list with selected results** - Xóa danh sách theo dõi và sau đó thêm các biểu tượng từ các dòng đã chọn vào danh sách theo dõi của bạn. Tìm hiểu thêm về điều này tại đây.
- **Clear result list** - Xóa tất cả các dòng từ danh sách kết quả.
- **Copy** - Sao chép danh sách kết quả vào clipboard của Windows, để bạn có thể dán vào một ứng dụng khác, như Excel chẳng hạn.

LƯU Ý QUAN TRỌNG:

1. Mũi tên mua là xanh đặc, mũi tên bán là đỏ đặc, mũi tên ngắn là đỏ rỗng, mũi tên đóng là xanh rỗng.
2. Mũi tên chỉ hiển thị trên các biểu đồ có thuộc tính "Show arrows" được bật.

Menu ngữ cảnh biểu đồ



Menu này xuất hiện khi bạn nhấp chuột PHẢI vào bảng biểu đồ.

Các tùy chọn có sẵn:

Parameters... - Mở hộp thoại Tham số cho phép bạn chỉnh sửa các tham số của các chỉ báo, cũng như màu sắc, kiểu dáng, cài đặt tỷ lệ và trục.

Edit Formula... - Mở Trình biên tập Công thức cho phép bạn xem/chỉnh sửa mã AFL của chỉ báo.

Close - Đóng bảng biểu đồ.

Intraday ... - Cho phép bạn chuyển đổi khung thời gian hiển thị sang một trong các khoảng thời gian intraday có sẵn.

Daily view - Chuyển đổi khoảng thời gian hiển thị sang hàng ngày.

Weekly view - Chuyển đổi khoảng thời gian hiển thị sang hàng tuần.

Monthly view - Chuyển đổi khoảng thời gian hiển thị sang hàng tháng.

Pane

- **Close** - Đóng bảng biểu đồ.
- **Arrange all** - Sắp xếp các bảng có chiều cao bằng nhau.
- **Move up** - Di chuyển bảng biểu đồ đã chọn lên (thay đổi thứ tự dọc của bảng).
- **Move down** - Di chuyển bảng biểu đồ đã chọn xuống (thay đổi thứ tự dọc của bảng).
- **Maximize** - Tối đa hóa bảng đã chọn để nó chiếm toàn bộ màn hình.
- **Restore** - Khôi phục bảng đã chọn về kích thước trước đó.

Template

- **Load...** - Tải mẫu biểu đồ cửa sổ đơn từ tệp đã chọn (tìm hiểu thêm về mẫu và bố cục tại đây).
- **Save...** - Lưu mẫu biểu đồ cửa sổ đơn vào tệp đã chọn.
- **Load default** - Tải mẫu cửa sổ đơn mặc định.

- **Save as default** - Lưu thiết lập cửa sổ đơn hiện tại làm mẫu mặc định.

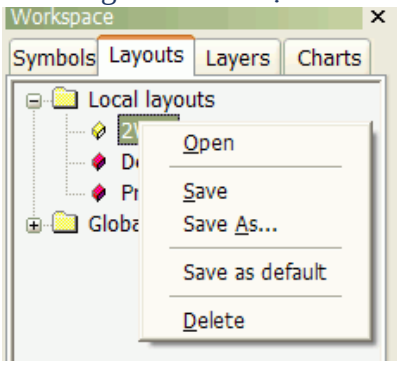
Delete indicator - Xóa một trong các phần chỉ báo kéo và thả được tìm thấy trong mã.

Delete study - Xóa nghiên cứu vẽ tay đã chọn (như đường xu hướng, Fibonacci, Gann...) - tìm hiểu thêm về điều này tại đây.

Delete All studies - Xóa tất cả các nghiên cứu vẽ tay (như đường xu hướng, Fibonacci, Gann...).

Properties - Hiển thị thuộc tính (tọa độ, màu sắc, v.v.) của nghiên cứu vẽ tay (như đường xu hướng, Fibonacci, Gann...) - tìm hiểu thêm tại đây.

Menu ngữ cảnh Bố cục



Menu ngữ cảnh Bố cục xuất hiện khi bạn nhấp chuột PHẢI vào bố cục trong cửa sổ Workspace, tab Bố cục. Các tùy chọn có sẵn:

Open - Tải bố cục đã chọn.

Save - Lưu bố cục cửa sổ hiện tại với tên hiện tại.

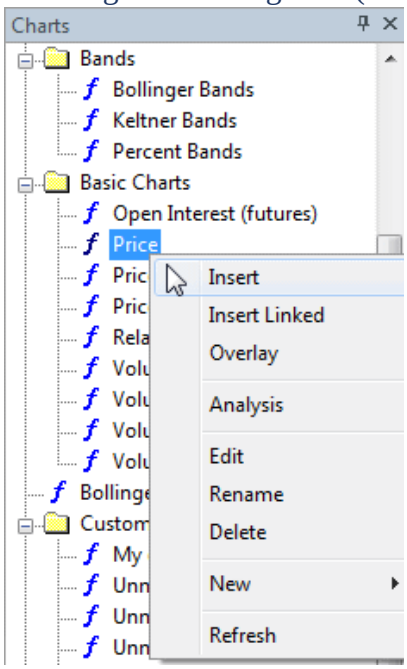
Save As... - Lưu bố cục cửa sổ hiện tại với tên mới.

Save as default - Lưu bố cục cửa sổ hiện tại làm bố cục mặc định (khởi động) cho cơ sở dữ liệu đã cho.

Delete - Xóa bố cục đã chọn.

Để tìm hiểu thêm về Bố cục, vui lòng tham khảo Hướng dẫn: Bảng biểu đồ và Bố cục.

Menu ngữ cảnh Công thức (biểu đồ)

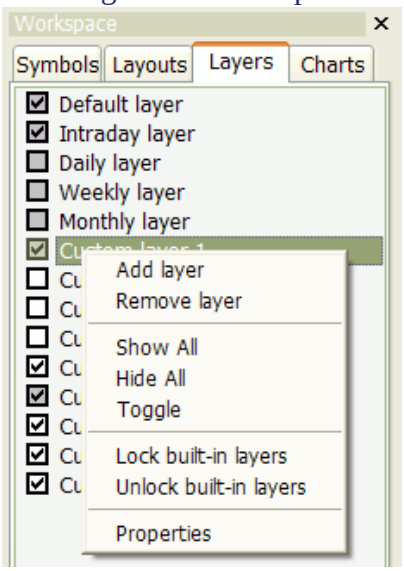


Menu ngữ cảnh Công thức (biểu đồ) xuất hiện khi bạn nhấp chuột PHẢI vào công thức được liệt kê trong tab Biểu đồ của cửa sổ Workspace.

Các tùy chọn có sẵn bao gồm:

- **Insert** - Chèn chỉ báo đã chọn vào bảng biểu đồ mới.
  Lệnh Insert sẽ tạo một bản sao của tệp công thức gốc và đặt bản sao này vào thư mục kéo-thả ẩn, do đó công thức gốc sẽ không bị ảnh hưởng bởi việc chỉnh sửa hoặc chồng thêm chỉ báo khác sau đó. Nhấp đúp vào tên công thức tương đương với việc chọn lệnh Insert từ menu.
- **Insert Linked** - Chèn chỉ báo đã chọn vào bảng biểu đồ mới một cách trực tiếp (liên kết với công thức gốc).
  Lệnh Insert Linked không tạo bản sao của công thức. Thay vào đó, nó tạo bảng biểu đồ mới liên kết trực tiếp với công thức gốc. Bằng cách này, việc chỉnh sửa và/hoặc chồng thêm các chỉ báo khác sẽ làm thay đổi công thức gốc.
- **Overlay** - Chồng chỉ báo đã chọn lên bảng biểu đồ đã chọn.
  Lệnh Overlay sẽ thêm mã bổ sung vào công thức được sử dụng bởi bảng biểu đồ. Nếu bảng biểu đồ đã cho được tạo bằng lệnh Insert Linked, nó sẽ làm thay đổi công thức gốc (đã liên kết).
- **Analysis** - Hiển thị cửa sổ Phân tích Tự động và chọn công thức đã chọn.
- **Edit** - Mở cửa sổ Trình biên tập Công thức để chỉnh sửa công thức đã chọn.
- **Rename** - Đổi tên tệp công thức hiện tại đã chọn.
- **Delete** - Xóa tệp công thức hiện tại đã chọn
- **New**
- **Formula** - Tạo tệp công thức mới trong thư mục hiện tại đã chọn.
- **Folder** - Tạo thư mục con mới trong thư mục hiện tại đã chọn
- **Refresh** - Đọc lại thư mục Công thức và hiển thị lại cây công thức.

Menu ngữ cảnh Các lớp



Menu ngữ cảnh Các lớp xuất hiện khi bạn nhấp chuột PHẢI vào danh sách lớp trong tab Các lớp của cửa sổ Workspace. Các tùy chọn có sẵn bao gồm:

- **Add layer** - Thêm một lớp mới.
- **Remove layer** - Xóa lớp đã chọn.
  Lưu ý rằng bạn không thể xóa 5 lớp đầu tiên (lớp tích hợp).
- **Show All** - Hiển thị tất cả các lớp không bị khóa.
- **Hide All** - Ẩn tất cả các lớp không bị khóa.
- **Toggle** - Chuyển đổi trạng thái hiển thị của các lớp không bị khóa.
- **Lock built-in layers** - Khóa 5 lớp đầu tiên (lớp tích hợp). Khi lớp bị khóa, trạng thái hiển thị của nó sẽ tự động thay đổi khi khoảng thời gian thay đổi và bạn không thể hiển thị/ẩn nó một cách thủ công.
- **Unlock built-in layers** - Mở khóa 5 lớp đầu tiên (lớp tích hợp). Khi lớp được mở khóa, trạng thái hiển thị của nó sẽ không thay đổi tự động khi khoảng thời gian thay đổi và bạn có thể hiển thị/ẩn nó một cách thủ công.
- **Properties** - Mở hộp thoại thuộc tính cho phép bạn đổi tên lớp và quyết định xem lớp đó có nên bị khóa theo khoảng thời gian hiển thị hay không. Nếu bạn đánh dấu ô "Lock visibility to interval", lớp sẽ tự động hiển thị/ẩn tùy thuộc vào khoảng thời gian hiện tại được hiển thị. Bạn có thể định nghĩa trạng thái hiển thị cho mỗi lớp bằng cách sử dụng hộp kết hợp "Interval" và các nút "Show/hide automatically". Lưu ý rằng có một cài đặt trạng thái hiển thị *riêng biệt* cho MỖI khoảng thời gian. Hộp thuộc tính lớp luôn hiển thị khoảng thời gian "monthly" khi bắt đầu, nhưng đây chỉ là điều kiện khởi động; bạn chỉ cần chuyển sang khoảng thời gian cụ thể và chỉnh sửa trạng thái hiển thị. Để thiết lập lớp bị khóa hoàn toàn, bạn phải định nghĩa trạng thái hiển thị cho mỗi lớp được liệt kê trong hộp kết hợp "Interval". Chỉ cần chọn khoảng thời gian và chọn xem lớp nên được hiển thị hay ẩn cho khoảng thời gian này, chọn khoảng thời gian tiếp theo và lại chọn hiển thị hoặc ẩn, chọn tiếp theo và tiếp tục cho đến khi bạn định nghĩa trạng thái hiển thị cho tất cả các khoảng thời gian.

Để biết thêm thông tin về các lớp và cách sử dụng chúng, hãy tham khảo Hướng dẫn: Sử dụng Các lớp.

Menu ngữ cảnh Báo giá Thời gian thực



Time & Sales - Mở cửa sổ Time & Sales, cung cấp thông tin về từng lệnh đặt, lệnh hỏi và giao dịch đang phát trực tuyến từ thị trường.

Easy Alerts - Mở cửa sổ Easy Alerts, cho phép bạn định nghĩa các cảnh báo thời gian thực được thực hiện khi lệnh đặt/lệnh hỏi/lần cuối và các trường khác đạt mức do người dùng xác định.

Add Symbol - Thêm biểu tượng hiện tại vào danh sách Báo giá Thời gian thực.

Add watch list... - Thêm toàn bộ danh sách theo dõi vào cửa sổ báo giá thời gian thực.

Type-in symbols - Cho phép nhập trực tiếp các biểu tượng dưới dạng danh sách ngăn cách bằng dấu phẩy.

Insert empty line - Thêm dòng trống (dòng phân cách) - hữu ích để nhóm các biểu tượng lại với nhau.

Remove Symbol - Xóa dòng (biểu tượng) đã chọn khỏi danh sách Báo giá Thời gian thực.

Remove All - Xóa tất cả các biểu tượng khỏi danh sách báo giá thời gian thực.

Hide - Ẩn danh sách Báo giá Thời gian thực.

Phím tắt

AmiBroker cho phép tùy chỉnh hoàn toàn giao diện người dùng, bao gồm cả phím tắt. Để định nghĩa phím tắt của riêng bạn, hãy sử dụng menu Tools->Customize, tab Keyboard. Đọc thêm về nó trong Hướng dẫn: Tùy chỉnh giao diện người dùng.

Danh sách phím tắt được định nghĩa trước như sau, xin lưu ý rằng nếu bạn sử dụng các tính năng tùy chỉnh bàn phím, danh sách ở đây có thể không hợp lệ vì một số mục nhập có thể đã được thay đổi thành của riêng bạn.

Phím tắt	**Mệnh lệnh**
CTRL+0	VIEW_HOURLY
CTRL+1	VIEW_1MINUTE
CTRL+5	VIEW_5MINUTE
CTRL+6	VIEW_15MINUTE
CTRL+C	EDIT_COPY
CTRL+D	VIEW_DAILY
CTRL+E	CHART_EDITFORMULA
CTRL+H	VIEW_CROSSHAIR
CTRL+I	CHART_MORE_INDICATORS
CTRL+M	VIEW_MONTHLY
CTRL+N	FILE_NEW
CTRL+O	FILE_OPEN
CTRL+P	FILE_PRINT
CTRL+R	CHART_PARAMETERS
CTRL+S	FILE_SAVE
CTRL+V	EDIT_PASTE
CTRL+ADD (CTRL+'+')	VIEW_ZOOM_IN
ALT+BACK	EDIT_UNDO
DELETE	EDIT_CLEAR
ALT+DELETE	EDIT_CLEAR_ALL
SHIFT+DELETE	EDIT_CUT
END	CHART_SCROLL_END
F1	HELP
SHIFT+F1	CONTEXT_HELP
F12	CHART_RANGE_BEGIN

CTRL+F12	CHART_RANGE_HIDE
SHIFT+F12	CHART_RANGE_END
F3	STOCK_FIND
F4	QUICK_FIND
F5	VIEW_REFRESH_CHARTS
F6	NEXT_PANE
SHIFT+F6	PREV_PANE
HOME	CHART_SCROLL_BEGIN
CTRL+INSERT	EDIT_COPY
SHIFT+INSERT	EDIT_PASTE
ALT+LEFT	SYMBOL_PREV
CTRL+ALT+LEFT	VIEW_GO_PREV
SHIFT+ALT+LEFT	SYMBOL_PREV_TREE
PAGE_DOWN	CHART_SCROLL_PAGE_RIGHT
CTRL+PAGE_DOWN	VIEW_SHEET_NEXT
PAGE_UP	CHART_SCROLL_PAGE_LEFT
CTRL+PAGE_UP	VIEW_SHEET_PREV
ALT+RETURN	CHART_STUDY_PROPERTIES
ALT+RIGHT	SYMBOL_NEXT
CTRL+ALT+RIGHT	VIEW_GO_NEXT
SHIFT+ALT+RIGHT	SYMBOL_NEXT_TREE
CTRL+SUBTRACT (CTRL+'-')	VIEW_ZOOM_OUT
CTRL+W	VIEW_WEEKLY
CTRL+X	EDIT_CUT
CTRL+Z	EDIT_UNDO

Nhập ASCII

AmiBroker có tính năng nhập trích dẫn dễ sử dụng và linh hoạt. Tài liệu này mô tả các khái niệm nâng cao về trình nhập ASCII AmiBroker. Người dùng mới nên bắt đầu với Trình hướng dẫn nhập ASCII.

Nó hoạt động như thế nào?

Dữ liệu trích dẫn có thể đến từ nhiều nguồn khác nhau nên định dạng của tệp ASCII (tức là dựa trên văn bản) có thể rất khác nhau giữa các nguồn. Để xử lý tất cả những khác biệt đó, AmiBroker sử dụng các lệnh định nghĩa định dạng để định nghĩa cách thông tin văn bản được trình nhập ASCII diễn giải. Các lệnh định dạng là các từ khóa bắt đầu bằng dấu đô la '$'. Các lệnh này có thể được nhúng trong chính tệp dữ liệu hoặc có thể được lưu trữ trong tệp định nghĩa định dạng riêng biệt để sử dụng nhiều lần. Việc lưu trữ các lệnh định dạng trong tệp riêng biệt giúp tránh phải đưa các lệnh vào mọi tệp dữ liệu. Tên tệp định dạng mặc định là "default.format". Tệp này, tất cả các tệp ".format" khác và tệp "import.types" (sẽ mô tả sau) phải được lưu trữ trong thư mục con \Formats của thư mục làm việc hiện tại của AmiBroker. Các mặc định bị ghi đè bởi bất kỳ lệnh nào được bao gồm (nhúng) trong chính tệp dữ liệu.

Vì vậy, khi bạn sử dụng menu "Nhập từ ASCII", AmiBroker trước tiên sẽ tìm định nghĩa định dạng được lưu trữ trong tệp "default.format" rồi phân tích tệp bạn đã chọn. Nếu không có tệp "default.format" thì nó sẽ sử dụng các mặc định nội bộ (được mô tả bên dưới).

Bạn có thể sửa đổi tệp "default.format" để phù hợp với nhu cầu của mình. Hơn nữa, khi sử dụng giao diện OLE Automation (phiên bản Win32) hoặc ARexx (Amiga), bạn có thể chỉ định tên của tệp định nghĩa định dạng sẽ được sử dụng thay cho tệp "default.format".

Các lệnh định dạng định dạng

Các từ khóa lệnh bắt đầu bằng dấu đô la '$". Mỗi dòng bắt đầu bằng lệnh được diễn giải theo cách đặc biệt. Sau đây là danh sách các lệnh được trình nhập tích hợp của AmiBroker nhận dạng. Các chữ cái in đậm đánh dấu các từ khóa.

Command	**$ADDRESS**	Define company address
Arguments	<string>	address of company
Alias		
Examples	**$ADDRESS "One Microsoft Way"**	

Command	**$ALLOWNEG**	Allow negative numbers in prices

Arguments	<number>	

		0 - do not allow negative values (default),
		1 - allow negative values in prices. This additionally switches off any checking for OHLC relationship so you can import any data into OHLC fields.
		
		when $ALLOWNEG is NOT specified in the ASCII importer definition AmiBroker performs the following range checking and fixup on open, low and high prices
		if(open == 0) open = close;
		if(high < max(open, close)) high = max(open, close);
		if(low == 0) low = min(open, close)

Alias		
Examples	**$ALLOWNEG 1**	

Command	**$ALLOW99SECONDS**	Convert invalid second stamp
Arguments	<onoff>	This flag works ONLY in conjunction with $TICKMODE 1 (see below for details) $ALLOW99SECONDS set to 1 will convert all records with invalid seconds (i.e greater than 59)to 59s. So record stamped 16:29:70 will be treated as 16:29:59
Alias		
Examples	**$ALLOW99SECONDS 1**	

Command	**$APPENDNAME**	append string to the ticker name (useful when you need to join several fields together to make unique stock symbol)
Arguments	<string>	string to append to the ticker symbol
Alias	**$APPENDTICKER**	
Examples		

Command	**$BREAKONERR**	Define on-error behaviour

Arguments	<number>	0 - to continue, 1 - to break import on error (default=0)
Alias		

Examples	**$BREAKONERR 1**

Command	**$CONT**	Define continuous quotations flag
Arguments	<number>	<0 or 1> - continuous quotations flag, this affects $AUTOADD 1 mode - if this is set, newly added stocks are switched to continuous quotation mode (this means enabling candlestick charts for example)
Alias		
Examples	**$CONT 1**	

Command	**$CURRENCY**	Define symbol's currency
Arguments	<string>	Defines currency of symbol
Alias		
Examples	**$CURRENCY EUR** or **$FORMAT NAME, CURRENCY** **$OVERWRITE 1** **$AUTOADD 1**	

Command	**$DATE_DMY**	Define date
Arguments	<number>	The date in Canadian format (DD-MM-YY). If there is no argument given the date is taken from the file name (without an extension)
Alias	**$DATE_CDN**	
Examples	**$DATE_DMY 12-05-99** **$DATE_CDN 12-05-1999**	

Command	**$DATE_MDY**	Define date

Arguments	<number>	The date in US format (MM-DD-YY). If there is no argument given the date is taken from the file name (without an extension)
Alias	**$DATE_USA**	
Examples	**$DATE_MDY 05/12/99** **$DATE_USA 05/12/99**	

Command	**$DATE_YMD**	Define date
Arguments	<number>	The date in International format (YY-MM-DD). If there is no argument given the date is taken from the file name (without an extension)
Alias	**$DATE_INT**	
Examples	**$DATE_INT 99-05-12** **$DATE_CDN 1999.05.12**	

Command	**$DEBUG**	Switch logging (debug) mode
Arguments	<number>	0 - no error logging, 1 - log errors to "import.log" file (default=0)
Alias		
Examples	**$DEBUG 1**	

Command	**$FORMAT**	Define line format (sequence and types of fields)
Arguments	**DATE_MDY**	date in US format: MM-DD-YY (alias: **DATE_USA**)
	DATE_DMY	date in Canadian format: DD-MM-YY (alias: **DATE_CDN**)
	DATE_YMD	date in International format: YY-MM-DD (alias: **DATE_INT**)
	TIME	time in HH:MM:SS or HH:MM or HHMM or HHMMSS format
	NAME	ticker name (alias: **TICKER**)
	ALIAS	symbol alias ($AUTOADD and $OVERWRITE modes only)

	FULLNAME	symbol full name ($AUTOADD and $OVERWRITE modes

	only)
OPEN	open price
HIGH	high price
LOW	low price
CLOSE	close price
ADJCLOSE	split-adjusted close This is provided to read adj. close column from Yahoo. Works **only** in conjunction with CLOSE field. When both CLOSE and ADJCLOSE are present in the ASCII format definition then importer calculates split factor by dividing ADJCLOSE/CLOSE. It then multiples OPEN, HIGH, LOW and CLOSE fields by this factor and divides VOLUME field by this factor. This effectively converts unadjusted prices to split adjusted prices. Split ratio gets locked once ADJCLOSE drops below 0.05.
OPENINT	open interest
VOLUME	volume
VOL1000	volume in thousands shares
VOLMIL	volume in millions shares
VOLFACTOR	volume factor (number of shares in a block) default =1
TURNOVER	turnover
AUX1	AUX1 field (auxilliary data)
AUX2	AUX2 field (auxilliary data)
SKIP	skip (ignore) field

MARKET	specify a field that contains market ID (affects $AUTOADD and

	$OVERWRITE modes only)
GROUP	specify a field that contains group ID (affects $AUTOADD and $OVERWRITE modes only)
WATCHLIST	specify a field that contains watch list number (0-31) (affects $AUTOADD and $OVERWRITE modes only)
INFO	specify a field with additional information (WSE specific: nk, ns, rk, rs, ok, os, zd, bd)
REDUCTION	specify a field with reduction rate in percents (WSE specific)
ICB	(new in 5.60) specify ICB code For example if your file looks as follows: (format is symbol, full name, ICB code) AAN,AARON'S INC,5375 Then to import it usign AmiBroker's import wizard use the following $FORMAT Ticker,FullName,ICB $OVERWRITE 1 $SEPARATOR , $CONT 1 $GROUP 255 $AUTOADD 1 $NOQUOTES 1

GICS	specify GICS code For example if your file looks as follows: (format is symbol, full name, gics sub industry code) AAN,AARON'S

	INC,25504060 Then to import it usign AmiBroker's import wizard use the following $FORMAT Ticker,FullName,GICS $OVERWRITE 1 $SEPARATOR , $CONT 1 $GROUP 255 $AUTOADD 1 $NOQUOTES 1
INDUSTRY	specify a field that contains industry ID (affects $AUTOADD and $OVERWRITE modes only)
INDUSTRYNAME	(new in 5.60) specifies a field that contains Industry Name. AmiBroker will check if given industry name already exists and if not, it will create a new Industry and assign imported stock to the industry specified. Also if SECTORNAME is specified, it will assign newly added industry to specified sector. (affects $AUTOADD and $OVERWRITE modes only)

SECTORNAME	(new in 5.60) specifies a field that contains Sector Name. AmiBroker will check if given sector name already exists and if not, it will create a new Sector. Also if INDUSTRYNAME is specified, it will assign newly added industry to specified sector. (affects $AUTOADD and $OVERWRITE modes

	only)
APPENDTICKER	specify a field that contains string that should be appended to the ticker name (useful when you need to join several fields together to make unique symbol symbol)
MARGIN	future contract margin deposit (positive value = dollars, negative value - percent of full value)
POINTVALUE	future contract point value
ROUNDLOTSIZE	round lot size (trading unit size)
TICKSIZE	tick size
ADVISSUES	number of advancing issues
ADVVOLUME	volume of advancing issues
DECISSUES	number of declining issues
DECVOLUME	volume of declining issues
UNCISSUES	number of unchanged issues
UNCVOLUME	volume of unchanged issues
ADDRESS	street address of company
CURRENCY	specifies currency of symbol
WEBID	specifies web ID

DIV_PAY_DATE **EX_DIV_DATE** **LAST_SPLIT_DATE** **LAST_SPLIT_RATIO** **EPS** **EPS_EST_CUR_YEAR** **EPS_EST_NEXT_YEAR** **EPS_EST_NEXT_QTR** **FORWARD_EPS** **PEG_RATIO** **BOOK_VALUE** (requires	fundamental data fields. For more info read Using Fundamental Data

	SHARES_OUT to be specified as well) **BOOK_VALUE_PER_SHARE** **EBITDA** **PRICE_TO_SALES** (requires **CLOSE** to be specified as well) **PRICE_TO_EARNINGS** (requires **CLOSE** to be specified as well) **PRICE_TO_BV** (requires **CLOSE** to be specified as well) **FORWARD_PE** (requires **CLOSE** to be specified as well) **REVENUE** **SHARES_SHORT** **DIVIDEND** **ONE_YEAR_TARGET** **MARKET_CAP** (requires **CLOSE** to be specified as well - it is used to calculate shares outstanding) **SHARES_FLOAT** **SHARES_OUT** **PROFIT_MARGIN** **OPERATING_MARGIN** **RETURN_ON_ASSETS** **RETURN_ON_EQUITY** **QTRLY_REVENUE_GROWTH** **GROSS_PROFIT** **QTRLY_EARNINGS_GROWTH** **INSIDER_HOLD_PERCENT** **INSTIT_HOLD_PERCENT** **SHARES_SHORT_PREV** **FORWARD_DIV** **OPERATING_CASH_FLOW** **FREE_CASH_FLOW** **BETA**	
Alias		
Examples	**$FORMAT TICKER DATE_MDY OPEN HIGH LOW CLOSE VOLUME** **$FORMAT TICKER, DATE_INT, CLOSE, VOLUME** **$FORMAT SKIP, TICKER, SKIP, SKIP, DATE_INT, OPEN, HIGH, LOW, CLOSE, TURNOVER**	

Command	**$FULLNAME**	Define full symbol name
Arguments	<string>	full symbol name
Alias		
Examples	**$FULLNAME Apple Computer Inc.**	

Command	**$GICS**	Define GICS code (Global Industry Category System)
Arguments	<number>	this affects $AUTOADD 1 and $OVERWRITE 1 modes - if this is specified symbols are assigned to given GICS category
Alias		
Examples	Now you can import GICS symbol-code assignments using ASCII importer. $FORMAT command now supports GICS code and there is $GICS command for single-symbol files. For example if your file looks as follows: (format is symbol, full name, gics sub industry code) AAN,AARON'S INC,25504060 Then to import it usign AmiBroker's import wizard use the following $FORMAT Ticker,FullName,GICS $OVERWRITE 1 $SEPARATOR , $CONT 1 $GROUP 255 $AUTOADD 1 $NOQUOTES 1	

Command	**$GROUP**	Define group ID
Arguments	<number>	this affects $AUTOADD 1 mode - if this is specified, newly added symbols are assigned to group with given number.
Alias		
Examples		

Command	**$HYBRID**	Switch hybrid mode on/off
Arguments	<number>	0 (off) or 1 (on). When this flag is set, you can combine quotations from multiple files - for example one file can contain only open prices and volume and the other file can contain high/low/close data. Useful especially for Warsaw Stock Exchange for combining the data from fixing and later continuous quotations.



Alias	
Examples	

Command	**$INDUSTRY**	Define industry ID
Arguments	<number>	this affects $AUTOADD 1 mode - if this is specified, newly added symbols are assigned to industry with given number.
Alias		
Examples		

Command	**$MARKET**	Define market ID
Arguments	<number>	this affects $AUTOADD 1 mode - if this is specified, newly added symbols are assigned to market with given number.
Alias		
Examples		

Command	**$NAME**	Define ticker name
Arguments	<ticker>	ticker name (symbol) (default = file name without path and extension)
Alias	**$TICKER**	
Examples	**$NAME AAPL** **$TICKER MSFT**	

Command	**$NOQUOTES**	Switch quotation data mode
Arguments	<number>	0 - (default) accept only quotation data (AmiBroker checks for non-zero prices and valid dates) 1 - switch off quotation data checking - this allows importing non-quotation data - for example only ticker and full names
Alias	**$TICKER**	
Examples	**$NAME AAPL** **$TICKER MSFT**	

Command	**$OVERWRITE**	Switch overwrite mode on/off

Arguments	<number>	0 - off, 1 - on. When overwrite mode is on then information provided by GROUP, MARKET, INDUSTRY, FULLNAME fields

		is overwritten for existing symbols (not only for newly added)
Alias	**$OVERWRITE 1**	

Command	**$PRICEFACTOR**	Define price factor
Arguments	<number>	the factor by which price data are multiplied (default = 1)
Alias		
Examples	**$PRICEFACTOR 100**	

Command	**$RAWCLOSE2OI**	Put Raw Close price to OI field
Arguments	<number>	0 - off, 1- on. (off by default) - causes that OpenInterest field gets assigned CLOSE (raw close) field value multiplied by 100
Alias		
Examples	**$RAWCLOSE2OI 1**	

Command	**$RECALCSPLITS**	Recalculate splits

Arguments	<number>	0 - off, 1- on. (off by default) causes that splits are recalculated by AmiBroker by the algorithm that tries to construct correct adjusted price, based on inaccurate information provided by Yahoo. Note that Yahoo provides only 2 decimal digits in adj. close field therefore the more adj. close approaches zero due to adjustements the error grows. The option $RECALCSPLITS 1 is intended to address this problem (at least partially). It works as follows: 1. for each bar ratio ADJCLOSE/CLOSE is calculated 2. if the ratio changes in two consecutive bars by more than 10% it means that split happened that bar. True split ratio is guessed by matching true fraction in the format of X/Y, where X and Y =

		1..9, to the change in ratios. 3. Then true split ratio is used to adjust all past bars until new split is detected. Works only in conjunction with ADJCLOSE
	$RECALCSPLITS 1	

Command	**$RECALCVOL**	Switch automatic index volume recalculation
Arguments	<number>	0 - off, 1 - on (base index only), 2 - on (all indexes). When this is on AmiBroker calculates volumes for indexes based on assignments to markets and base indexes defined in Categories window
Alias		
Examples	**$RECALCVOL 2**	

Command	**$RECALCAD**	Switch automatic advance/decline composite recalculation
Arguments	<number>	0 - off, 1 - on. When this is on AmiBroker calculates numbers and volumes of issues advancing, declining and unchanged based on assignments to markets and base indexes defined in Categories window.
Alias		
Examples	**$RECALCVOL 2**	

Command	**$ROUNDADJ**	Round split adjusted prices to given number of decimaldigits
Arguments	<decimaldigits>	*decimaldigits* - causes split-adjusted prices (see above) to be rounded to 'decimaldigits' precision. By default no rounding is done Works only in conjunction with ADJCLOSE
Alias		

Examples	$ROUNDADJ 2

Command	**$SEPARATOR**	Define field separator character
Arguments	<separator char>	the character used to separate data fields (default = space)
Alias		
Examples	**$SEPARATOR ,** **$SEPARATOR ;**	

Command	**$SKIPLINES**	Define how many lines to skip (ignore)
Arguments	<number>	number of lines to skip (default = 0)
Alias		
Examples	**$SKIPLINES 1**	

Command	**$STRICT**	Switches on/off strict checking if Open, High, Low prices are greater than zero
Arguments	<onoff>	(default = 0)
Alias		
Examples	**$STRICT 1**	

Command	**$TICKMODE**	Switches on/off tick mode $TICKMODE is a special mode of importer that allows to import quotes that haved It makes two assumptions: a) input data should come in the ascending time order (i.e. OLDER records first, b) input data should consist of entire tick history because importer will DELETE ar Once again: Turning on $TICKMODE 1 For example data files like this: Can be imported using the following definition file: $FORMAT Ticker, Skip, Date_YMD, Time, Open, High, Low, Close, Volume

		$SKIPLINES 1 $SEPARATOR , $CONT 1 $GROUP 255 $AUTOADD 1 $DEBUG 1 $TICKMODE 1 Sometimes it happens that input files have invalid timestamps (seconds > 59). For example: MOL,0,20050606,162970,16400.0000,16400.0000,16400.0000,16400.0000,2 Please take a closer look at first line shown in this example it has time:16:29:70 (y So I had to add a special flag to the importer that works around such data errors. It is called $ALLOW99SECONDS 1 and will convert all records with invalid secon So record stamped 16:29:70 will be treated as 16:29:59 Now for tick mode to work with such incorrect records you would need to add two $TICKMODE 1 $ALLOW99SECONDS 1
Arguments	<onoff>	(default = 0)
Alias		
Examples	**$TICKMODE 1**	

Command	**$TIMESHIFT**	Define intraday time shift used during import
Arguments	<number>	number of hours to shift date/time stamps (can be fractional)
Alias		
Examples	**$TIMESHIFT 2** ; will shift 2 hours forward **$TIMESHIFT -11.5** ; will shift 11 and half hour backward	

Command	**$VOLFACTOR**	Define volume factor

Arguments	<number>	

		the factor by which volume data is multiplied (default = 1)
Alias	**$VOLFACTOR 10**	

Command	**$WATCHLIST**	Define watch list number
Arguments	<number>	this affects $AUTOADD 1 and $OVERWRITE 1 modes - if this is specified, newly added symbols are added to the watch list with given number.
Alias		
Examples		

Command	**$CLEANSECTORS**	Clean (wipe) existing sector/industry structure
Arguments	<number>	if this is turned on (1), existing sector/ industry structure will be deleted and initialized with Sector 0, 1, 2, 3...63/ Industry 0...255 This command should only be used in conjunction with **SECTORNAME**, **INDUSTRYNAME** $FORMAT fields to allow setting up fresh industry structure
Alias		
Examples	See example below (importing sector/industry structure)	

Command	**$SORTSECTORS**	Sort sector/industry structure
Arguments	<number>	if this is turned on (1), sector/ industry structure will be sorted alphabetically after importing. This command should only be used in conjunction with **SECTORNAME**, **INDUSTRYNAME** $FORMAT fields to allow setting up fresh industry structure
Alias		
Examples	See example below (importing sector/industry structure)	

Command	**$USEONLYLOCALDB**	Switches "Use only local database" option for the symbol

Arguments	<number>	If data is fed by database plugin, using the ASCII importer to add any symbol causes these newly added symbol to have "Use only local database" flag turned on. A new command: $USEONLYLOCALDB 0 allows to turn this off (so newly added symbols have "use only local database" turned off) This flag does NOT affect existing symbols.
Alias		
Examples		

Command	**$WEBID**	Define web ID
Arguments	<string>	web ID
Alias		
Examples	**$WEBID aapl**	

Lưu ý:

• đối với DATE_xxx, bạn có thể sử dụng - , / hoặc \ làm dấu phân cách ngày/tháng/năm. Bạn thậm chí có thể bỏ dấu phân cách nếu bạn chỉ nhập ngày theo định dạng 6 chữ số (YYMMDD, MMDDYY, DDMMYY) hoặc 8 chữ số (YYYYMMDD, MMDDYYYY, DDMMYYYY).

• AmiBroker nhận dạng cả phân số thập phân và phân số thực trong dữ liệu giá. Phân số thực phải theo sau giá trị toàn bộ sau ít nhất một khoảng trắng. Ví dụ, bạn có thể chỉ định: 5,33 hoặc 5 1/3

AmiBroker không giới hạn ở bất kỳ loại phân số nào, nếu muốn, bạn có thể viết ngay cả: 5 333/999

Bình luận

Bạn có thể đưa bình luận vào cả tệp định nghĩa định dạng và tệp dữ liệu. Mỗi dòng bắt đầu bằng * (dấu sao) hoặc ; (dấu chấm phẩy) hoặc # (dấu thăng) được coi là bình luận và bị trình nhập ASCII bỏ qua.

Ví dụ sử dụng

Những gì có vẻ phức tạp từ danh sách lệnh sẽ trở nên khá rõ ràng sau một số ví dụ. Vì vậy, tôi sẽ cung cấp cho bạn bốn ví dụ về cách viết tệp định dạng. Ví dụ đầu tiên sẽ hiển thị định nghĩa cho các trích dẫn CSV (giá trị được phân tách bằng dấu phẩy) có sẵn trên trang tài chính của Yahoo. Ví dụ thứ hai sẽ hiển thị định nghĩa cho định dạng tệp ASCII của

Metastock. Ví dụ thứ ba hiển thị định nghĩa cho định dạng tệp ASCII của Omega SuperCharts. Và ví dụ thứ tư sẽ hiển thị định nghĩa cho các tệp s được DM BOS (công ty môi giới Ba Lan) sử dụng.

Yahoo CSV

Dữ liệu từ trang web của Yahoo trông như sau:

Ngày, Mở, Cao, Thấp, Đóng, Khối lượng

1-Tháng 2- 0,104,105,100,100.25,2839600

31-Tháng 1- 0,101,103.875,94.50,103.75,6265000

28-Tháng 1- 0,108.1875,110.875,100.625,101.625,3779900

Dòng đầu tiên cho chúng ta một gợi ý về ý nghĩa của các trường được phân tách bằng dấu phẩy. Trường đầu tiên sẽ chứa ngày. Các trường còn lại sẽ chứa giá mở, cao, thấp, đóng và khối lượng. Người nhập khẩu nên bỏ qua dòng đầu tiên và phân tích tất cả các dòng còn lại chỉ chứa dữ liệu được phân tách bằng dấu phẩy. Tệp định dạng định dạng phù hợp sẽ trông như thế này:

$FORMAT Date_DMY,Open,High,Low,Close,Volume

$SKIPLINES 1

$SEPARATOR ,

$DEBUG 1

$AUTOADD 1

$BREAKONERR 1

$DEBUG bật ghi nhật ký lỗi vào tệp "import.log" và $BREAKONERR sẽ khiến trình nhập dừng lại sau khi tìm thấy lỗi đầu tiên. $AUTOADD đảm bảo rằng mã cổ phiếu mới sẽ được thêm vào cơ sở dữ liệu nếu mã cổ phiếu đó bị thiếu. Vâng... bạn có thể hỏi: làm sao nó biết được tên mã cổ phiếu? Câu trả lời rất đơn giản: nếu không có trường nào xác định tên mã cổ phiếu, trình nhập sẽ lấy tên tệp (không có đường dẫn và phần mở rộng) làm mã cổ phiếu. Vì vậy, nếu bạn đang nhập tệp "C:\My data\AAPL.CSV" thì AmiBroker sẽ sử dụng "AAPL" làm tên mã cổ phiếu.

Metastock ASCII

Dữ liệu ở định dạng Metastock ASCII trông như sau:

<ticker>,<per>,<date>,<high>,<low>,<close>,<vol>
AAP,D,1/17/2000,5483.33,5332.01,5362.3,0 AKS,D,1/17/2000,9868.45,9638.03,9687.62,0
FET,D,1/17/2000,3741.3,3540.2,3570.81,0

Trường đầu tiên sẽ chứa tên ticker, trường thứ hai - khoảng thời gian ("D" có nghĩa là dữ liệu hàng ngày), trường thứ ba - ngày báo giá. Phần còn lại sẽ chứa giá cao, giá thấp, giá đóng cửa và khối lượng. Sau đó, người nhập khẩu nên bỏ qua dòng đầu tiên và phân tích tất cả các dòng còn lại chỉ chứa dữ liệu được phân tách bằng dấu phẩy. Tệp định dạng thích hợp sẽ trông như thế này:

$FORMAT Ticker,Skip,Date_MDY,High,Low,Close,Volume

$SKIPLINES 1

$SEPARATOR ,

$DEBUG 1

$AUTOADD 1

$BREAKONERR 1

Skip trong $FORMAT định nghĩa một trường mà trình nhập sẽ bỏ qua.

Omega SuperCharts ASCII

Dữ liệu ở định dạng Omega SC ASCII trông như sau:

ticker,date,open,high,low,close,vol
AAP,20000117,5333.01,5483.33,5332.01,5362.3,3433450

Định dạng này tương tự như các định dạng trước đó, tuy nhiên ngày ở định dạng YYYYMMDD không có dấu phân cách giữa năm, tháng và ngày. Tuy nhiên, AmiBroker có thể xử lý những ngày như vậy một cách dễ dàng. Tệp định dạng thích hợp sẽ trông như thế này:

$FORMAT Name,Date_Int,Open,High,Low,Close,Volume

$SEPARATOR ,

$DEBUG 1

$SKIPLINES 1

$AUTOADD 1

$BREAKONERR 1

Skip in $FORMAT định nghĩa một trường mà trình nhập sẽ bỏ qua.

Tệp S của DMBOS

Dữ liệu ở định dạng này trông như sau:

0,29-02-00,12:05,MIDWIG,1069.1,,,+1.2,336002000,

0,29-02-00,12:05,NIF,48.6,,,+0.8,1763000,

0,29-02-00,12:05,WIG20,2300.3,,,+1.1,336002000,

0,29-02-00,12:05,WIG,21536.8,,,+0.2,336002000,

0,29-02-00,12:05,WIRR, 2732.8,,,+1.6,16373000,

1,29-02-00,12:05,AGORA,144.00,,,+4.7,15802000,

1,29-02-00,12:05,AGROS,40.00,nk,72,+5.0,840000,

1,29-02-00,12:05,AMERBANK,28.00,,,+3.7,22000,

1,29-02-00,12:05,AMICA,41.50,nk,99,+2.2,564000,

Định dạng này phức tạp hơn một chút. Đối với chúng tôi, các trường hữu ích là: 2 - ngày, 4 - mã chứng khoán, 5 - giá đóng cửa, 9 - giá trị doanh thu (đóng * khối lượng). Các trường còn lại chứa thông tin khác không hữu ích đối với chúng tôi. Tệp định dạng định dạng phù hợp sẽ trông như thế này:

$FORMAT Skip,Date_DMY,Skip,Name,Close,Skip,Skip,Skip,Turnover

$SEPARATOR ,

$DEBUG 1

Nhập cấu trúc Ngành/Lĩnh vực

Giả sử chúng ta có một tệp văn bản với Mã chứng khoán, Họ và tên đầy đủ, Tên ngành và tên ngành được liệt kê theo từng dòng như sau:

"DDD", "3D Systems Corporation", "Technology", "Computer Software: Prepackaged Software" "MMM", "3M Company", "Health Care", "Medical/Dental Instruments"

"SVN", "7 Days Group Holdings Limited", "Consumer Services", "Hotels/Resorts" "AHC", "A.H. Belo Corporation", "Consumer Services", "Newspapers/Magazines" "AIR", "AAR Corp.", "Capital Goods", "Aerospace"

"AAN", "Aaron's, Inc.", "Technology", "Diversified Commercial Services" "ABB", "ABB Ltd", "Consumer Durables", "Electrical Products"

Để nhập tệp như vậy, chúng tôi sử dụng định dạng sau:

$FORMAT Ticker, FullName,SectorName,IndustryName

$SEPARATOR,

$AUTOADD 1

$NOQUOTES 1

$OVERWRITE 1

$CLEANSECTORS 1

$SORTSECTORS 1

$NOQUOTES 1 cho trình nhập biết rằng chúng ta sẽ nhập dữ liệu không phải là báo giá. $AUTOADD 1/$OVERWRITE 1 là bắt buộc để tự động thêm ký hiệu mới và ghi đè thông tin ký hiệu hiện có. $CLEANSECTORS 1 xóa cấu trúc ngành/cổ phiếu hiện có trước khi nhập và $SORTSECTORS 1 - sắp xếp các ngành/ngành sau khi nhập để chúng được liệt kê theo thứ tự bảng chữ cái trong cửa sổ Ký hiệu. Lệnh $FORMAT chỉ xác định thứ tự và loại trường cần nhập

AmiBroker sẽ đọc từng tệp ASCII như vậy, sau đó sẽ kiểm tra xem tên ngành/tên ngành đã cho có tồn tại hay không, nếu không - sẽ tạo ngành/ngành mới. Sau đó, sẽ gán ký hiệu đã cho cho ngành/ngành đã chỉ định.

Kết quả sẽ là cơ sở dữ liệu với cấu trúc ngành/ngành mới đang được thiết lập và các ký hiệu được gán cho các ngành và ngành thích hợp.

Chức năng được mô tả được sử dụng để triển khai Công cụ->Cập nhật danh sách ký hiệu và danh mục Hoa Kỳ.

Hành vi mặc định

Khi nhập tệp ASCII, AmiBroker cố gắng mở tệp "default.format" (trong thư mục AmiBroker) để lấy định nghĩa định dạng. Nếu tệp đó bị thiếu, định dạng mặc định sau sẽ được áp dụng:

$FORMAT DATE_USA, OPEN, HIGH, LOW, CLOSE, VOLUME

$SEPARATOR

Điều này có nghĩa là theo mặc định, trình nhập ASCII sẽ sử dụng ký tự khoảng trắng làm dấu phân cách và sẽ phân tích các trường sau: date, open, high, low, close, volume. Tên tệp (không có đường dẫn và phần mở rộng) sẽ được sử dụng làm tên mã chứng khoán. Tất cả các tham số nhập khác ($DEBUG,$AUTOADD, v.v.) được đặt thành 0.

Các loại tệp và định dạng do người dùng xác định

Bây giờ AmiBroker không chỉ có thể sử dụng tệp định nghĩa default.format mà còn có thể sử dụng các tệp khác do người dùng chỉ định. Các loại tệp, bộ lọc và tệp định nghĩa định dạng được chỉ định trong tệp import.types (ví dụ có trong gói cập nhật). Bây giờ, người dùng có thể chuẩn bị/sửa đổi tệp import.types với mô tả về các định dạng ASCII được hỗ trợ và bộ lọc để sử dụng. Định dạng của tệp import.types là:

<Tên mô tả>|<Bộ lọc tệp>|<tên tệp định nghĩa>

Lưu ý các ký tự dòng dọc giữa ba trường này. Ví dụ về tệp import.types trông như sau:

ASCII mặc định (*.*)|*.*|default.format CSV của Yahoo (*.csv)|*.csv|yahoo.format

Metastock ASCII (*.mst)|*.mst|metastock.format Omega SC ASCII (*.txt)|*.txt|omega.format

S-Files (s*.*)|s*.*|sfile.format C-Files (c*.*)|c*.*|cfile.format

Sharenet DAT (*.dat)|*.dat|dat.format

Nếu tệp đó tồn tại, bạn sẽ thấy các loại của mình trong hộp kết hợp "Files of type" và khi bạn chọn một loại - bộ lọc thích hợp sẽ được sử dụng và sau khi chọn một số tệp và nhấp vào OK - trình nhập sẽ sử dụng tệp ".format" đã chỉ định.

Theo cách đó, bạn có thể định nghĩa nhiều định dạng dữ liệu dạng văn bản tùy thích và AmiBroker sẽ có thể "hiểu" tất cả chúng.

Biệt danh mã chứng khoán

Bây giờ, mỗi mã chứng khoán có thể được gán một biệt danh, do đó, các trình nhập tích hợp của AmiBroker có thể nhận dạng bảo mật đó bằng cả ký hiệu mã chứng khoán và tên biệt danh. Điều này hữu ích khi bạn sử dụng hai nguồn dữ liệu sử dụng quy ước đặt tên ký hiệu hơi khác nhau hoặc nếu bạn muốn đặt tên trực quan hơn cho các ký hiệu trong khi vẫn giữ được khả năng sử dụng trình nhập mà không gặp sự cố.

Phân loại GICS

GICS là tiêu chuẩn phân loại ngành toàn cầu, hãy xem http://en.wikipedia.org/wiki/Global_Industry_Classification_Standard để biết thêm chi tiết về hệ thống GICS.

Mã GICS có từ 2 đến 8 chữ số. Chẳng hạn như 10 cho ngành năng lượng hoặc 351010 cho ngành "Thiết bị & vật tư chăm sóc sức khỏe".

Các mã này được cố định ngay cả khi các phân loại mới được thêm vào tại một thời điểm nào đó trong tương lai. Điều quan trọng là phải hiểu rằng các mã này hoạt động theo cách phân cấp.

LƯU Ý: cơ sở dữ liệu hiện tại KHÔNG có mã GICS được gán cho các ký hiệu.

Theo như tôi biết, PremiumData http://www.premiumdata.net/ đang có kế hoạch phát hành cơ sở dữ liệu tương thích với AmiBroker có hỗ trợ GICS.

AmiBroker hiện đọc tệp GICS.txt từ thư mục cài đặt của nó. Tệp này chứa các danh mục GICS được liệt kê từng cái một theo thứ tự mã GICS theo định dạng sau

GICS;Tên;Mô tả<CRLF>

GICS là mã số từ 2 chữ số đến 8 chữ số Tên là tên danh mục GICS

Mô tả là mô tả danh mục GICS Các trường này phải được phân tách bằng dấu chấm phẩy

<CRLF> có nghĩa là ký tự xuống dòng/xuống dòng (có nghĩa là "dòng mới" - chỉ cần nhấn phím ENTER/RETURN nếu bạn đang chỉnh sửa bằng trình soạn thảo văn bản)

Chỉ được có một danh mục trên mỗi dòng trong tệp GICS.txt

Tệp GICS.txt mặc định đã được cung cấp.

Phân loại ICB

ICB là viết tắt của Industry Classification Benchmark (http://en.wikipedia.org/wiki/Industry_Classification_Benchmark).

AmiBroker cũng cho phép hệ thống phân loại 4 cấp độ ICB, nhưng cơ sở dữ liệu demo không có ký hiệu được phân loại theo tiêu chuẩn đó. Bạn có thể tìm thấy mã phân loại ICB trong tệp ICB.txt bên trong thư mục AmiBroker.

LƯU Ý: cơ sở dữ liệu hiện tại KHÔNG có mã ICB được gán cho ký hiệu.

Phân loại ICB cho cổ phiếu NYSE có thể được nhập từ http://www.nyse.com/indexes/nyaindex.csv

AmiBroker hiện đọc tệp ICB.txt từ thư mục cài đặt của nó. Tệp này chứa các danh mục ICB được liệt kê từng cái một theo thứ tự mã ICB theo định dạng sau

ICB;Name<CRLF>

ICB là mã số gồm 4 chữ số.

Tên là tên danh mục ICB

Các trường này phải được phân tách bằng dấu chấm phẩy

<CRLF> có nghĩa là ký tự xuống dòng/xuống dòng (có nghĩa là "dòng mới" - chỉ cần nhấn phím ENTER/RETURN nếu bạn đang chỉnh sửa bằng trình soạn thảo văn bản)

Chỉ được phép có một danh mục trên mỗi dòng trong tệp ICB.txt Tệp ICB.txt mặc định đã được cung cấp.

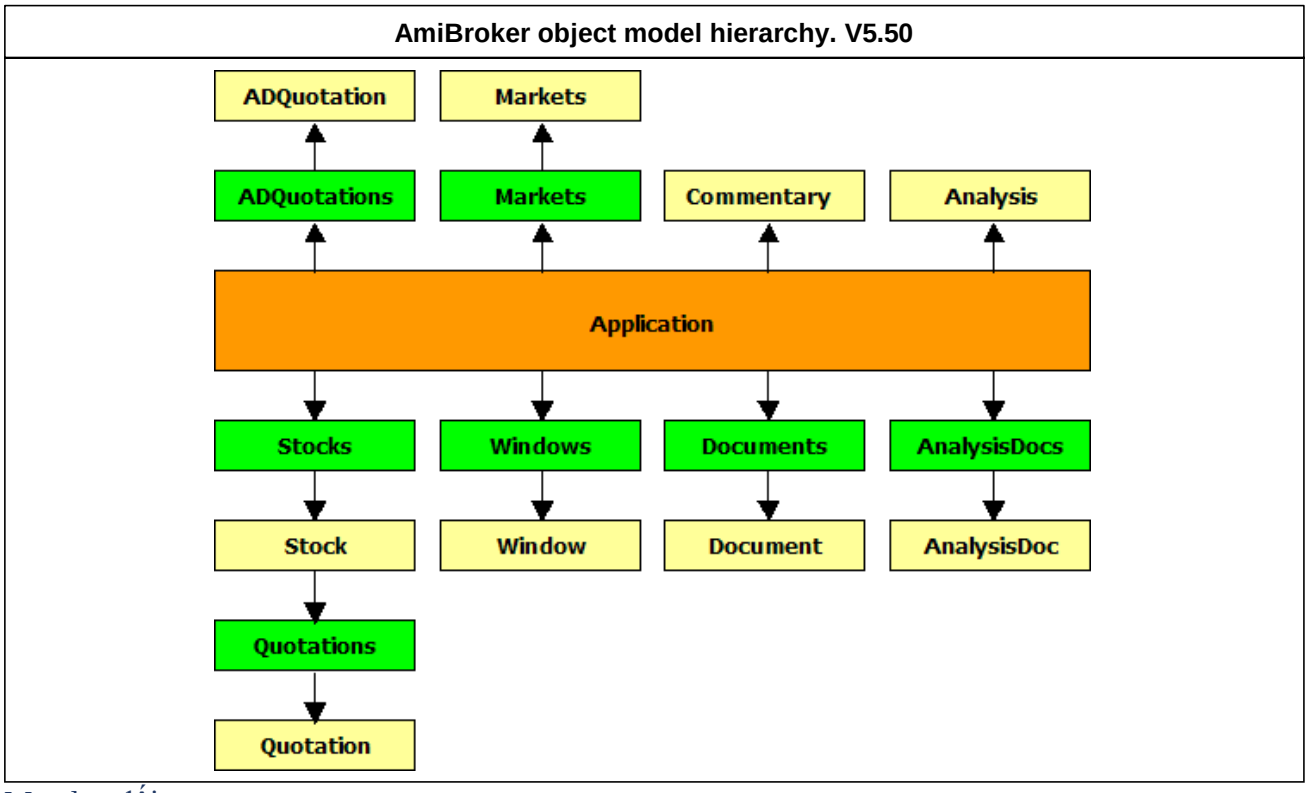
Dấu thời gian có độ phân giải cao (mili giây và micro giây)

Bắt đầu từ phiên bản 6.14, trình nhập ASCII hỗ trợ dấu thời gian có độ phân giải micro giây (HH:MM:SS.mmmuuu) trong đó mmm - mili giây 000..999, uuu - micro giây 000..999

Mô hình đối tượng tự động hóa OLE của AmiBroker

Lưu ý quan trọng về tự động hóa OLE:

Giao diện tự động hóa OLE được cung cấp để điều khiển AmiBroker từ quy trình BÊN NGOÀI (chẳng hạn như máy chủ tập lệnh windows). Mặc dù có thể truy cập Broker.Application và các đối tượng cơ bản từ công thức AFL, nhưng bạn phải rất cẩn thận KHÔNG chạm vào bất kỳ đối tượng giao diện người dùng nào (Documents, Document, Windows, Window, Analysis object) từ công thức AFL vì làm như vậy, bạn có thể sẽ "Cưa đứt nhánh cây mà bạn đang ngồi". Đặc biệt, những việc như chuyển đổi tab biểu đồ từ công thức biểu đồ đang chạy hoàn toàn bị cấm. Việc thay đổi các đối tượng giao diện người dùng thông qua OLE từ AFL hiện đang chạy trong các phần giao diện người dùng đó là công thức dẫn đến thảm họa. Bạn đã được cảnh báo.



Mục lục đối tượng

- ADQuotation
- Windows
- Markets
- ADQuotations
- Commentary
- Quotation
- Analysis(1)

• Document

• Quotations

• AnalysisDoc(2)

• Documents

• Stock

• AnalysisDocs(2)

• Market

• Application

• Window

• Stocks

(1) - Đối tượng Analysis đã lỗi thời kể từ phiên bản 5.50. Nó được để lại ở đây để tương thích ngược và chỉ truy cập vào cửa sổ Phân tích tự động cũ

(2) - Đối tượng AnalysisDoc và bộ sưu tập AnalysisDocs là các đối tượng mới được giới thiệu trong v5.50 và cho phép kiểm soát cửa sổ Phân tích mới

ADQuotation

Thuộc tính:

• Ngày là Ngày

• AdvIssues As Long

• AdvVolume As Single

• DecIssues As Long

• DecVolume As Single

• UncIssues As Long

• UncVolume As Single

Mô tả:

Lớp ADQuotation giữ một thanh thông tin tiến/giảm

ADQuotations

Phương pháp:

• Hàm Add(ByVal Date As Variant) As Object

• Hàm Remove(ByVal Date As Variant) As Boolean

Thuộc tính:

• Mục(ByVal Date As Variant) As Object [r/o] [mặc định]

• Đếm là Dài

Mô tả:

ADQuotations là một tập hợp các đối tượng ADQuotation

Phân tích

Đối tượng này đã lỗi thời. Nó chỉ được cung cấp để duy trì khả năng tương thích với mã cũ. Đối tượng Phân tích luôn truy cập Phân tích tự động CŨ.

Thuộc tính:

• Thuộc tính Filter(ByVal nType As Integer, ByVal pszCategory As String) As Long [r/w]

Phương pháp:

• Sub Backtest([ByVal Type As Variant])

• Sub ClearFilters()

• Sub Edit([ByVal bForceReload As Variant])

• Sub Explore()

• Hàm Export(ByVal pszFileName As String) As Boolean

• Hàm LoadFormula(ByVal FileName As String) As Boolean

• Hàm LoadSettings(ByVal pszFileName As String) As Boolean

• Sub MoveWindow(ByVal Left As Long, ByVal Top As Long, ByVal Width As Long, ByVal Height As Long)

• Sub Optimize([ByVal Type As Variant])

• Hàm Report(ByVal pszFileName As String) As Boolean

• Hàm SaveFormula(ByVal pszFileName As String) As Boolean

• Hàm SaveSettings(ByVal pszFileName As String) As Boolean

• Sub Scan()

• Sub ShowWindow(ByVal nShowCmd As Long)

• Sub SortByColumn(ByVal iColumn As Long, ByVal bAscending As Integer, ByVal

bMultiMode As Integer)

Thuộc tính:

• RangeMode As Long

• RangeN As Long

• RangeFromDate As Date

• RangeToDate As Date

• ApplyTo As Long

Mô tả:

Đối tượng phân tích cung cấp khả năng kiểm soát theo chương trình của cửa sổ phân tích tự động

Ghi chú:

Analysis.Backtest(Type = 2); - chạy backtest

Tham số Type có thể là một trong các giá trị sau:

0 : danh mục backtest/optimize

1 : danh mục backtest/optimize cá nhân

2 : danh mục backtest/optimize cũ

CẦN LƯU Ý RẰNG VÌ LÝ DO TƯƠNG THÍCH NGƯỢC, CHẾ ĐỘ BACKTESTER MẶC ĐỊNH

LÀ BACKTE "CŨ". DO ĐÓ, BẠN PHẢI CHỈ ĐỊNH TYPE = 0 NẾU BẠN MUỐN NHẬN PORTFOLIO BACKTEST.

Analysis.Optimize(Type = 2); - chạy tối ưu hóa Tham số loại có thể là một trong các giá trị sau:

0 : kiểm tra ngược danh mục đầu tư/tối ưu hóa

1 : kiểm tra ngược cá nhân/tối ưu hóa

2 : kiểm tra ngược cũ/tối ưu hóa

3 : kiểm tra tiến tới (AmiBroker phiên bản 5.11.0 trở lên)

Analysis.Report(FileName: String) - lưu báo cáo vào tệp hoặc hiển thị báo cáo nếu FileName = ""

Analysis.ApplyTo - định nghĩa chế độ áp dụng: 0 - tất cả cổ phiếu, 1 - cổ phiếu hiện tại, 2 - sử dụng bộ lọc Analysis.RangeMode - định nghĩa chế độ phạm vi: 0 - tất cả báo giá, 1 - n báo giá cuối cùng, 2 - n ngày cuối cùng, 3

- từ-đến ngày

Analysis.RangeN - định nghĩa N (số thanh/ngày để kiểm tra ngược)
Analysis.RangeFromDate - định nghĩa ngày "Từ" Analysis.RangeToDate - định nghĩa ngày "Đến"

Analysis.Filter(nType: short, Category : String) - đặt/lấy cài đặt bộ lọc Đối số nType định nghĩa loại bộ lọc 0 - bao gồm, 1 - loại trừ

Đối số danh mục định nghĩa danh mục bộ lọc:

"index", "favorite", "market", "group", "sector", "index", "watchlist"

AnalysisDoc

AnalysisDoc là một đối tượng mới được giới thiệu trong phiên bản 5.50. Nó cho phép truy cập các tài liệu dự án New Analysis (phần mở rộng apx) và thực hiện quét/khám phá/kiểm tra ngược/tối ưu hóa đa luồng trong cửa sổ New Analysis theo cách không đồng bộ. Không đồng bộ có nghĩa là phương thức Run() chỉ bắt đầu quy trình và trả về ngay lập tức. Để chờ hoàn tất, bạn phải kiểm tra cờ IsBusy định kỳ (chẳng hạn như mỗi giây) trong mã của riêng bạn.

Thuộc tính:

• Thuộc tính IsBusy As Boolean [r]

Phương pháp:

• Sub Close()

• Hàm Export(ByVal pszFileName As String, [ByVal WhatToExport As Variant]) As Long

• Hàm Run(ByVal Action As Long) As Long

• Sub Abort() (mới trong 6.20)

Mô tả:

Đối tượng AnalysisDoc cung cấp khả năng kiểm soát theo chương trình đối với tài liệu/cửa sổ New Analysis.

Thuộc tính IsBusy cho phép kiểm tra bất cứ khi nào cửa sổ Phân tích bận thực hiện phân tích. Bạn phải kiểm tra cờ này định kỳ nếu bạn muốn chờ hoàn tất. Cẩn thận KHÔNG gọi cờ này quá thường xuyên vì nó sẽ làm giảm hiệu suất. Để có kết quả tốt nhất, hãy kiểm tra cờ này sau mỗi giây. Bạn cũng cần kiểm tra cờ này nếu bạn không chắc chắn bất cứ khi nào

cửa sổ Phân tích bận trước khi thử gọi Export() hoặc Run(), nếu không, các lệnh gọi này sẽ không thành công nếu phân tích đang diễn ra.

Phương thức Close() đóng tài liệu/cửa sổ Phân tích. Nếu có bất kỳ thao tác nào đang diễn ra, thao tác đó sẽ bị chấm dứt. Để ngăn chặn việc chấm dứt sớm, hãy kiểm tra thuộc tính IsBusy.

Phương thức Export(pszFileName, whatToExport) cho phép xuất danh sách kết quả phân tích sang tệp .HTML hoặc .CSV. Trả về 1 nếu thành công (xuất thành công) hoặc 0 nếu không (ví dụ nếu cửa sổ phân tích đang bận). WhatToExport quyết định dữ liệu nào sẽ được xuất: whatToExport = 0 - xuất danh sách kết quả (hành vi mặc định khi tham số này không được cung cấp), whatToExport = 1 - xuất tab walkforward.

Phương thức Run(Action) cho phép chạy quét/khám phá/kiểm tra ngược/tối ưu hóa không đồng bộ. Tham số Action có thể là một trong các giá trị sau:

0 : Quét

1 : Khám phá

2 : Kiểm tra ngược danh mục đầu tư

3 : Kiểm tra ngược cá nhân

4 : Tối ưu hóa danh mục đầu tư

5 : Tối ưu hóa cá nhân (được hỗ trợ bắt đầu từ v5.69)

6 : Kiểm tra tiến trình

Điều quan trọng là phải hiểu rằng phương thức Run chỉ bắt đầu quy trình và trả về ngay lập tức. Nó KHÔNG chờ hoàn tất.

Để chờ hoàn tất, bạn cần truy vấn cờ IsBusy theo định kỳ (chẳng hạn như cứ sau một giây).

Run() trả về 1 khi thành công (quy trình bắt đầu thành công) hoặc 0 khi thất bại (ví dụ nếu cửa sổ phân tích đang bận)

Quy trình chạy kiểm tra ngược tự động bao gồm mở dự án Phân tích đã lưu trước đó (bao gồm tất cả các thiết lập cần thiết để thực hiện bất kỳ hành động nào), gọi Run() và chờ hoàn tất.

Vì hiện tại bạn có thể chạy nhiều dự án phân tích, nên có một bộ sưu tập AnalysisDocs đại diện cho tất cả các tài liệu Analysis đang mở và cho phép bạn mở các tệp đã lưu trước đó (có chứa công thức, cài đặt và mọi thứ cần thiết để chạy).

Đối tượng AnalysisDoc mới không cho phép bạn đọc/ghi cài đặt cho mục đích này - bạn không được phép thao tác UI trong khi cửa sổ Analysis mới đang chạy. Cách sử dụng đúng cửa sổ Phân tích mới là mở tệp dự án hiện có và chạy. Nếu bạn muốn sửa đổi cài đặt, bạn nên viết/sửa đổi tệp dự án hiện có. Tệp dự án phân tích (phần mở rộng .apx) là tệp định dạng XML tự giải thích có thể đọc được bằng con người, có thể được viết/chỉnh sửa/sửa đổi từ bất kỳ ngôn ngữ / trình soạn thảo văn bản nào.

Ví dụ JScript sau

a) mở dự án phân tích từ tệp C:\Analysis1.apx

b) bắt đầu kiểm tra ngược (không đồng bộ)

c) chờ hoàn thành

d) xuất kết quả

e) đóng tài liệu phân tích

```
AB = new ActiveXObject( "Broker.Application" ); // tạo đối tượng AmiBroker

thử
{
NewA = AB.AnalysisDocs.Open( "C:\\analysis1.apx" ); // mở
tệp dự án phân tích đã lưu trước đó
// NewA biểu thị phiên bản của tài liệu/cửa sổ Phân tích mới

if ( NewA )
{
NewA.Run( 2 ); // bắt đầu kiểm tra ngược không đồng bộ

while ( NewA.IsBusy ) WScript.Sleep( 500 ); // kiểm tra IsBusy sau mỗi 0,5 giây

NewA.Export( "test.html" ); // xuất danh sách kết quả sang tệp HTML

WScript.echo( "Hoàn thành" );

NewA.Close(); // đóng Phân tích mới
```

```
}
}
catch ( err )
{

WScript.echo( "Ngoại lệ: " + err.message ); // hiển thị lỗi có thể xảy ra
}
```

Phương thức Abort( ) sẽ hủy bỏ bất kỳ hoạt động quét/khám phá/kiểm tra ngược Phân tích nào đang chạy

AnalysisDocs

AnalysisDocs là một đối tượng mới được giới thiệu trong phiên bản 5.50. Đây là một tập hợp các đối tượng AnalysisDoc. Cho phép Thêm Phân tích mới, Mở dự án phân tích hiện có và lặp qua các đối tượng phân tích.

Phương pháp:

• Hàm Add() Là Đối tượng

• Sub Close()

• Hàm Open(ByVal FileName Là Chuỗi) Là Đối tượng

Thuộc tính:

• Mục(ByVal Index As Long) Là Đối tượng [r/o] [mặc định]

• Đếm As Long

• Ứng dụng Là Đối tượng

• Cha Là Đối tượng

Mô tả:

AnalysisDocs là một tập hợp các đối tượng AnalysisDoc.

Phương pháp Add tạo tài liệu/cửa sổ Analysis mới. Phương pháp trả về đối tượng AnalysisDoc

.

Phương pháp Close đóng tất cả các tài liệu/cửa sổ Analysis đang mở. Nếu bất kỳ dự án phân tích nào đang chạy, dự án đó sẽ bị chấm dứt ngay lập tức

Phương pháp Open cho phép mở tệp dự án Analysis hiện có (.apx). Phương pháp trả về đối tượng AnalysisDoc.

Thuộc tính mục cho phép truy cập vào phần tử Index-th của bộ sưu tập. Thuộc tính trả về đối tượng AnalysisDoc.

Thuộc tính Count cung cấp số lượng tài liệu phân tích đang mở.

Cả thuộc tính Application và Parent đều trỏ đến đối tượng Broker.Application. Ví dụ về cách sử dụng, hãy xem mô tả đối tượng AnalysisDoc.

Ứng dụng

Phương pháp:

• Hàm Import(ByVal Type As Integer, ByVal FileName As String, [ByVal DefFileName As Variant]) As Long

• Hàm LoadDatabase(ByVal Path As String) As Boolean

• Hàm LoadLayout(ByVal pszFileName As String) As Boolean

• Sub LoadWatchlists()

• Hàm Log(ByVal Action As Integer) As Long

• Sub Quit()

• Sub RefreshAll()

• Sub SaveDatabase()

• Hàm SaveLayout(ByVal pszFileName As String) As Boolean

Thuộc tính:

• ActiveDocument As Object

• Stocks As Object

• Version As String

• Documents As Object

• Markets As Object

• DatabasePath As String

• Analysis As Object

• Commentary As Object

• ActiveWindow As Đối tượng

• Hiển thị dưới dạng số nguyên

Mô tả:

Đối tượng ứng dụng là đối tượng tự động hóa OLE chính cho AmiBroker. Bạn phải tạo đối tượng này trước khi truy cập bất kỳ đối tượng nào khác. Để tạo đối tượng Ứng dụng, hãy sử dụng mã sau:

JScript:

AB = new ActiveXObject("Broker.Application"); VB/VBScript:

AB = CreateObject("Broker.Application") AFL:

AB = CreateObject("Broker.Application");

Cửa sổ

Phương pháp:

• Sub Activate()

• Sub Close()

• Hàm ExportImage(ByVal FileName As String, [ByVal Width As Variant], [ByVal Height As Variant], [ByVal Depth As Variant]) As Boolean

• Hàm LoadTemplate(ByVal lpszFileName As String) As Boolean

• Hàm SaveTemplate(ByVal lpszFileName As String) As Boolean

• Hàm ZoomToRange(ByVal From As Variant, ByVal To As Variant) As Boolean

Thuộc tính:

• SelectedTab As Long

• Document As Object

Mô tả:

Đối tượng cửa sổ cung cấp khả năng kiểm soát theo chương trình đối với cửa sổ biểu đồ.

Windows

Phương pháp:

• Hàm Add() As Object

Thuộc tính:

• Item(ByVal Index As Long) As Object [r/o] [mặc định]

• Count As Long

Mô tả:

Windows là một tập hợp các đối tượng Window.

Bình luận

Phương pháp:

• Sub Apply()

• Sub Close()

• Hàm LoadFormula(ByVal pszFileName As String) As Boolean

• Hàm Save(ByVal pszFileName As String) As Boolean

• Hàm SaveFormula(ByVal pszFileName As String) As Boolean

Mô tả:

Đối tượng bình luận cung cấp quyền kiểm soát theo chương trình đối với cửa sổ bình luận của guru.

Tài liệu

Phương pháp:

• Sub Activate()

• Sub Close()

• Sub ShowMessage(ByVal Text As String)

Thuộc tính:

• Application As Object

• Parent As Object

• Name As String

• ActiveWindow As Object

• Windows As Object

• Interval As Integer

Mô tả:

Đối tượng tài liệu biểu thị tài liệu đang hoạt động (thuộc loại 'biểu đồ'). Trong kiến trúc xem tài liệu, mỗi tài liệu có thể có nhiều cửa sổ (chế độ xem) được kết nối. Thuộc tính Name xác định ký hiệu hiện đang được chọn cho tài liệu.

Name là ký hiệu ticker, Interval là khoảng thời gian biểu đồ tính bằng giây.

Tài liệu

Phương pháp:

• Hàm Add() As Object

• Sub Close()

• Hàm Open(ByVal Ticker As String) As Object

Thuộc tính:

• Item(ByVal Index As Long) As Object [r/o] [mặc định]

• Count As Long

• Application As Object

• Parent As Object

Mô tả:

Tài liệu là một tập hợp các đối tượng tài liệu.

Market

Thuộc tính:

• Name As String

• ADQuotations As Object

Mô tả:

Market biểu diễn danh mục thị trường và dữ liệu liên quan (tức là thông tin tăng/giảm theo từng thị trường)

Markets

Thuộc tính:

• Item(ByVal Index As Integer) As Object [r/o] [mặc định]

• Count As Integer

Mô tả:

Markets là tập hợp các đối tượng Market

Quotation

Thuộc tính:

• Date As Date

• Close As Single

• Open As Single

• High As Single

• Low As Single

• Volume As Single

• OpenInt As Single

Mô tả:

Lớp Quotation biểu diễn một thanh dữ liệu giá

Quotations

Phương pháp:

• Hàm Add(ByVal Date As Date) As Object

• Hàm Adjust(ByVal FieldList As String, ByVal Multiplier As Float, ByVal Offset là Float, ByVal DateTime là Date, ByVal Before là Boolean) As Long - mới trong phiên bản 6.40.2

hàm thực hiện điều chỉnh giá trên các báo giá, FieldList xác định các trường nào sẽ được điều chỉnh, chẳng hạn như "OHLC", giá trị của mỗi trường được đề cập trong danh sách trường được nhân với bội số và thêm giá trị bù trừ: new_value = old_value * Multiplier + Offset

báo giá bị ảnh hưởng sẽ là Before (=True) hoặc sau DateTime đã chỉ định

• Hàm Remove(ByVal Item As Variant) As Boolean

• Hàm Retrieve(ByVal Count As Long, ByRef Date As Variant, ByRef Open As Variant, ByRef High As Variant, ByRef Low As Variant, ByRef Close As Variant, ByRef Volume As Variant, ByRef OpenInt As Variant) As Long

Thuộc tính:

• Item(ByVal Item As Variant) As Object [r/o] [mặc định]

• Count As Long

Mô tả:

Quotations là một tập hợp các đối tượng Quotation. Nó đại diện cho tất cả các trích dẫn có sẵn cho ký hiệu đã cho. Bộ sưu tập Quotations có sẵn như một thuộc tính của đối tượng Stock.

Cổ phiếu

Thuộc tính:

• Ticker dạng Chuỗi

• Trích dẫn dạng Đối tượng

• FullName dạng Chuỗi

• Index dạng Boolean
• Favourite dạng Boolean
• Continuous dạng Boolean
• MarketID dạng Dài
• GroupID dạng Dài
• Beta dạng Đơn
• SharesOut dạng Đơn
• BookValuePerShare dạng Đơn
• SharesFloat dạng Đơn
• Address dạng Chuỗi
• WebID dạng Chuỗi
• Alias dạng Chuỗi
• IsDirty dạng Boolean
• IndustryID dạng Dài
• WatchListBits dạng Dài
• DataSource dạng Dài
• DataLocalMode dạng Dài
• PointValue dạng Đơn
• MarginDeposit dạng Đơn
• RoundLotSize dạng Đơn
• TickSize dạng Đơn
• WatchListBits2 dạng Dài
• Currency dạng Chuỗi
• LastSplitFactor dạng Chuỗi
• LastSplitDate dạng Ngày
• DividendPerShare dạng Đơn
• DividendPayDate dạng Ngày
• ExDividendDate dạng Ngày
• PEGRatio dạng Đơn
• ProfitMargin dạng Đơn
• OperatingMargin dạng Đơn
• OneYearTargetPrice dạng Đơn

• ReturnOnAssets dạng Đơn

• ReturnOnEquity dạng Đơn

• QtrlyRevenueGrowth dạng Đơn

• GrossProfitPerShare dạng Đơn

• SalesPerShare dạng Đơn

• EBITDAPerShare dạng Đơn

• QtrlyEarningsGrowth dạng Đơn

• InsiderHoldPercent dạng Đơn

• InstitutionHoldPercent dạng Đơn

• SharesShort dạng Đơn

• SharesShortPrevMonth dạng Đơn

• ForwardDividendPerShare dạng Đơn

• ForwardEPS As Single

• EPS As Single

• EPSEstCurrentYear As Single

• EPSEstNextYear As Single

• EPSEstNextQuarter As Single

• OperatingCashFlow As Single

• LeveredFreeCashFlow As Single

Mô tả:

Lớp Stock biểu diễn dữ liệu ký hiệu đơn. Vì lý do lịch sử, tên của đối tượng là Stock, nhưng nó có thể chứa bất kỳ loại công cụ nào (bao gồm cả hợp đồng tương lai, ngoại hối, v.v.).

Stocks

Phương pháp:

• Hàm Add(ByVal Ticker As String) As Object

• Hàm GetTickerList(ByVal nType As Long) As String

• Hàm Remove(ByVal Item As Variant) As Boolean

Thuộc tính:

• Item(ByVal Item As Variant) As Object [r/o] [mặc định]

• Count As Long

Mô tả:

Stocks là một tập hợp các đối tượng Stock. Nó có sẵn dưới dạng thuộc tính của đối tượng Application.

Lưu ý:

Stock.WatchListBits (dài) - mỗi bit 0..31 biểu diễn phép gán cho một trong 32 danh sách theo dõi để thêm một cổ phiếu vào danh sách theo dõi thứ n (ví dụ JScript):

Stock.WatchListBits |= 1 << nth;

Stock.WatchListBits2 (dài) - mỗi bit 0..31 biểu diễn việc gán cho một trong các danh sách theo dõi được đánh số từ 32..63 để thêm một cổ phiếu vào danh sách theo dõi thứ n (ví dụ JScript): Stock.WatchListBits2 |= 1 << (nth - 32);

Stock.DataSource (0 - mặc định, 1 - chỉ cục bộ)

Stock.DataLocalMode (0 - mặc định, 1 - lưu trữ cục bộ, 2 - không lưu trữ cục bộ)

Ví dụ thực tế:

Ví dụ 1: Chạy backtest đơn giản

```
AB = new ActiveXObject( "Broker.Application" ); // tạo đối tượng AmiBroker

try
{
NewA = AB.AnalysisDocs.Open( "C:\\analysis1.apx" ); // mở tệp
dự án phân tích đã lưu trước đó
// NewA biểu diễn phiên bản của tài liệu/cửa sổ Phân tích mới

if ( NewA )
{
NewA.Run( 2 ); // bắt đầu kiểm tra ngược không đồng bộ

while ( NewA.IsBusy ) WScript.Sleep( 500 ); // kiểm tra IsBusy sau mỗi 0,5 giây
```

```
NewA.Export( "test.html" ); // xuất danh sách kết quả sang tệp HTML WScript.echo( "Hoàn thành" );

NewA.Close(); // đóng Phân tích mới

}

}

catch ( err )

{

WScript.echo( "Ngoại lệ: " + err.message ); // hiển thị lỗi có thể xảy ra

}
```

Ví dụ 2: Thực thi chú thích

```
AB = new ActiveXObject("Broker.Application"); AB.Commentary.LoadFormula("C:\\Program Files\\AmiBroker\\AFL\\MACD_c.afl"); AB.Commentary.Apply();

AB.Commentary.Save("Test.txt"); AB.Commentary.SaveFormula("MACDTest.afl");

//AB.Bình luận.Đóng();
```

Chỉ mục đối tượng Mô hình đối tượng tự động hóa OLE của AmiQuote

Mục lục đối tượng

- **Tài liệu**

AmiQuote là ứng dụng SDI (tài liệu đơn) do đó chỉ có một lớp - Tài liệu - có thể tạo bằng mã sau:

JScript:

AB = new ActiveXObject("AmiQuote.Document"); VB/VBScript:

AB = CreateObject("AmiQuote.Document") AFL:

AB = CreateObject("AmiQuote.Document");

Tài liệu

Phương pháp:

• Hàm AddSymbols(ByVal pszSymbols As String) Là Boolean

• Hàm Download() Là Boolean

• Hàm GetSymbolsFromAmiBroker() Là Boolean

• Hàm Import() Là Boolean

• Sub MoveWindow(ByVal x As Long, ByVal y As Long, ByVal width As Long, ByVal height As Long)

• Hàm Open(ByVal pszFileName As String) Là Boolean

• Hàm RemoveAllSymbols() Là Boolean

• Hàm RemoveSymbols(ByVal pszSymbols As String) Là Boolean

• Hàm Save() Là Boolean

• Hàm SaveAs(ByVal pszFileName As String) Là Boolean

Thuộc tính:

• DownloadInProgress Là Boolean

• ImportInProgress Là Boolean

• Nguồn dài bao nhiêu

• Từ dài bao nhiêu Ngày

• Đến dài bao nhiêu Ngày

• Tự động nhập dưới dạng Boolean

• Tất cả các phiên dưới dạng Boolean

• Khoảng thời gian dài bao nhiêu

• Chạy mọi lúc dài bao nhiêu

• Thư mục đích dưới dạng Chuỗi

Hướng dẫn phân tích kỹ thuật

Giới thiệu

Công cụ cơ bản

Chỉ báo

Giới thiệu

Phân tích kỹ thuật là việc xem xét các biến động giá trong quá khứ để dự báo các biến động giá trong tương lai. Các nhà phân tích kỹ thuật đôi khi được gọi là nhà phân tích biểu đồ vì họ hầu như chỉ dựa vào biểu đồ để phân tích.

Phân tích kỹ thuật áp dụng cho cổ phiếu, chỉ số, hàng hóa, hợp đồng tương lai, tiền tệ hoặc bất kỳ công cụ giao dịch nào mà giá chịu ảnh hưởng của lực cung và cầu. Giá đề cập đến bất kỳ sự kết hợp nào giữa giá mở, giá cao, giá thấp hoặc giá đóng cửa của một chứng khoán nhất định trong một khung thời gian cụ thể. Khung thời gian có thể dựa trên dữ liệu giá trong ngày, hàng ngày, hàng tuần hoặc hàng tháng và kéo dài trong vài giờ hoặc nhiều năm. Ngoài ra, một số nhà phân tích kỹ thuật bao gồm các số liệu về khối lượng hoặc lãi suất mở trong nghiên cứu của họ về hành động giá.

AmiBroker cung cấp một bộ công cụ phân tích kỹ thuật toàn diện sẽ được trình bày trong chương này

Các công cụ cơ bản

AmiBroker có các công cụ phân tích kỹ thuật cơ bản sau:

• Biểu đồ giá

• Đường xu hướng

• Đường trung bình động

• Đường thoái lui Fibonacci

• Múi giờ Fibonacci

• Kênh hồi quy

• Dải Bollinger

Biểu đồ giá

AmiBroker có thể hiển thị giá bằng cách sử dụng:

• biểu đồ đường

chế độ này được sử dụng khi ký hiệu hiện tại sử dụng giá cố định và chỉ có giá đóng cửa khả dụng

• biểu đồ thanh truyền thống

chế độ này được sử dụng khi giao dịch liên tục được bật, nhưng giá mở không khả dụng (hoặc bằng giá đóng cửa)

• Nến Nhật

chế độ này được sử dụng khi giao dịch liên tục được bật với dữ liệu mở/đóng/cao/thấp

Biểu đồ đường là loại biểu đồ đơn giản nhất. Một mức giá (đóng cửa) được vẽ cho mỗi khoảng thời gian. Một đường duy nhất kết nối mỗi điểm giá này. Điểm mạnh chính của loại biểu đồ này là tính đơn giản.

Biểu đồ thanh là một trong những loại biểu đồ phổ biến nhất được sử dụng trong phân tích kỹ thuật. Đối với mỗi ngày giao dịch, một đường thẳng đứng được vẽ. Đỉnh của đường thẳng đứng biểu thị mức giá cao nhất mà một chứng khoán được giao dịch trong ngày và đáy biểu thị mức giá thấp nhất. Giá đóng cửa được hiển thị bằng dấu ở bên phải của thanh và giá mở cửa được hiển thị ở bên trái của thanh.

Được người Nhật phát triển vào những năm 1600, biểu đồ nến chỉ là biểu đồ thanh làm giảm mối quan hệ giữa giá mở cửa, giá cao, giá thấp và giá đóng cửa. Mỗi nến biểu thị một giai đoạn dữ liệu (ngày-tuần) và bao gồm bóng trên, bóng dưới và thân nến. Bóng trên là mức giá cao nhất mà cổ phiếu được giao dịch trong giai đoạn đó trong khi bóng dưới biểu thị mức giá thấp nhất. Thân nến có màu đen khi giá đóng cửa thấp hơn giá mở cửa hoặc màu trắng khi giá đóng cửa cao hơn giá mở cửa. Đỉnh của thân nến là giá mở cửa nếu nến có màu đen và nến được gọi là nến đen dài. Nếu nến có màu trắng, đỉnh của thân nến là giá đóng cửa và nến được gọi là nến trắng dài.

Các bài viết của Steven Nison giải thích về biểu đồ nến xuất hiện trên tạp chí Futures số tháng 12 năm 1989 và tháng 4 năm 1990. Cuốn sách chính thức về chủ đề này là Japanese Candlestick Charting Techniques cũng của Steve Nison.

Có nhiều dạng nến khác nhau. Một số được coi là các hình thành nhỏ trong khi một số khác là hình thành lớn. Biểu đồ nến minh họa rõ nét các khái niệm cung/cầu được xác định bởi các lý thuyết phân tích kỹ thuật cổ điển.

Hình thành biểu đồ nến chính:

Gravestone Doji: Doji (mở và đóng giống nhau) và mức cao cao hơn đáng kể so với giá mở, giá cao và giá đóng. Hình thành này thường xảy ra ở đáy của một xu hướng và báo hiệu sự đảo chiều tăng giá.

Dragon-fly Doji: Doji (mở và đóng giống nhau) và mức thấp thấp hơn đáng kể so với giá mở, giá cao và giá đóng. Hình thành này thường xảy ra ở đỉnh của một xu hướng và báo hiệu sự đảo chiều giảm giá.

Abandoned Baby Doji: Doji, xảy ra ở đáy của một hình thành biểu đồ với các khoảng trống ở cả hai bên của doji.

Harami Cross: Hình thành này báo hiệu đỉnh thị trường. Nó bao gồm một harami, là một nến dài màu đen đi trước và bao trùm một doji không có thân.

Engulfing Pattern: Một mô hình tăng giá gồm hai nến bao gồm một đường màu đen dài nhỏ bị bao phủ bởi nến thứ hai, một đường màu trắng dài.

Evening Star: Một mô hình giảm giá thường xảy ra ở đỉnh. Mô hình bao gồm ba nến. Nến đầu tiên là một đường màu trắng dài theo sau là một ngôi sao và sau đó là một đường màu đen dài. Ngôi sao có thể có màu đen hoặc trắng.

Dark Cloud Cover: Một mô hình gồm hai nến trong đó nến đầu tiên là một đường màu trắng dài và nến thứ hai là một đường màu đen dài có thân nến nằm bên dưới tâm của nến đầu tiên. Đây là mô hình giảm giá.

Đường xu hướng

Phân tích kỹ thuật được xây dựng dựa trên giả định rằng giá cả có xu hướng. Đường xu hướng là một công cụ quan trọng trong phân tích kỹ thuật để xác định và xác nhận xu hướng. Đường xu hướng là đường thẳng nối hai hoặc nhiều điểm giá và sau đó kéo dài đến tương lai để hoạt động như một đường hỗ trợ hoặc kháng cự. Nhiều nguyên tắc áp dụng cho các mức hỗ trợ và kháng cự cũng có thể được áp dụng cho đường xu hướng.

Đường xu hướng tăng

Đường xu hướng tăng có độ dốc dương và được hình thành bằng cách nối hai hoặc nhiều điểm thấp. Điểm thấp thứ hai phải cao hơn điểm thấp thứ nhất để đường có độ dốc dương. Đường xu hướng tăng hoạt động như hỗ trợ và chỉ ra rằng cầu ròng (cầu trừ cung) đang tăng ngay cả khi giá tăng. Giá tăng kết hợp với cầu tăng là rất lạc quan và cho thấy quyết tâm mạnh mẽ từ phía người mua. Miễn là giá vẫn ở trên đường xu hướng, xu hướng tăng được coi là vững chắc và nguyên vẹn. Một sự phá vỡ dưới đường xu hướng tăng cho thấy cầu ròng đã yếu đi và xu hướng có thể sắp thay đổi.

Đường xu hướng giảm

Đường xu hướng giảm có độ dốc âm và được hình thành bằng cách nối hai hoặc nhiều điểm cao. Đỉnh thứ hai phải thấp hơn đỉnh đầu tiên để đường có độ dốc âm. Đường xu hướng giảm đóng vai trò là lực cản và chỉ ra rằng cung ròng (cung trừ đi cầu) đang tăng ngay cả khi giá giảm. Giá giảm kết hợp với cung tăng là rất bi quan và cho thấy quyết tâm mạnh mẽ của người bán. Miễn là giá vẫn nằm dưới đường xu hướng giảm, xu hướng giảm được coi là vững chắc và nguyên vẹn. Phá vỡ trên đường xu hướng giảm cho thấy cung ròng đang giảm và xu hướng có thể sắp thay đổi.

Cài đặt thang đo

Điểm cao và điểm thấp dường như xếp hàng tốt hơn cho đường xu hướng khi giá được hiển thị bằng thang đo bán logarit. Điều này đặc biệt đúng khi đường xu hướng dài hạn được vẽ hoặc có sự thay đổi lớn về giá. AmiBroker cho phép đặt thang đo là số học hoặc logarit (bán logarit). Thang đo số học hiển thị các giá trị gia tăng (5,10,15,20,25,30) đều nhau khi chúng di chuyển lên trục y. Biến động giá 10 đô la sẽ trông giống nhau từ 10 đô la đến 20 đô la hoặc từ 100 đô la đến 110 đô la. Thang đo bán logarit hiển thị các giá trị gia tăng theo phần trăm khi chúng di chuyển lên trục y. Biến động từ 10 đô la đến 20 đô la là mức tăng 100% và có vẻ lớn hơn nhiều so với biến động từ 100 đô la đến 110 đô la, chỉ là mức tăng 10%.

Tuy nhiên, hãy nhớ rằng đường thẳng trong biểu đồ logarit không còn thẳng trong thang đo tuyến tính nữa, do đó, các đường xu hướng được vẽ trong thang đo này có thể trông lạ ở thang đo kia.

Xác thực

Cần hai hoặc nhiều điểm để vẽ đường xu hướng. Càng sử dụng nhiều điểm để vẽ đường xu hướng, thì mức hỗ trợ hoặc kháng cự được biểu thị bằng đường xu hướng càng có giá trị. Đôi khi, có thể khó tìm được nhiều hơn 2 điểm để xây dựng đường xu hướng. Mặc dù đường xu hướng là một khía cạnh quan trọng của phân tích kỹ thuật, nhưng không phải lúc nào cũng có thể vẽ đường xu hướng trên mọi biểu đồ giá. Đôi khi mức thấp hoặc mức cao không khớp nhau và tốt nhất là không nên ép buộc vấn đề. Nguyên tắc chung trong phân tích kỹ thuật là cần hai điểm để vẽ đường xu hướng và điểm thứ ba xác nhận tính hợp lệ.

Đường trung bình động

Đường trung bình động là một trong những công cụ phân tích hữu ích, khách quan và lâu đời nhất. Một số mô hình và chỉ báo có thể hơi chủ quan, trong đó các nhà phân tích có thể không đồng ý về việc mô hình có thực sự hình thành hay có độ lệch có thể là ảo tưởng. Đường trung bình động là một cách tiếp cận rõ ràng hơn để phân tích biểu đồ cổ phiếu và dự đoán hiệu suất, và đây là một trong số ít không yêu cầu trí thông minh thiên tài để giải thích.

Đường trung bình động là một chỉ báo cho thấy giá trị trung bình của giá chứng khoán trong một khoảng thời gian.

Để tìm Đường trung bình động đơn giản 50 ngày, bạn sẽ cộng giá đóng cửa (nhưng không phải lúc nào cũng nhiều hơn sau này) từ 50 ngày qua và chia cho 50. Và vì giá liên tục thay đổi nên đường trung bình động cũng sẽ thay đổi.

Đường trung bình động hàm mũ (EMA) - được tính bằng cách áp dụng tỷ lệ phần trăm giá đóng cửa hôm nay vào giá trị trung bình động của ngày hôm qua. Sử dụng đường trung bình động hàm mũ để đặt nhiều trọng số hơn vào giá gần đây. Như mong đợi, mỗi mức giá mới có tác động lớn hơn đến EMA so với SMA. Và, mỗi mức giá mới chỉ thay đổi đường trung bình động một lần, không phải hai lần.

Các đường trung bình động được sử dụng phổ biến nhất là đường trung bình động 15, 20, 30, 45, 50, 100 và 200 ngày. Mỗi đường trung bình động cung cấp một cách diễn giải khác nhau về diễn biến giá cổ phiếu. Thực sự không chỉ có một khung thời gian "đúng". Mỗi đường trung bình động có khoảng thời gian khác nhau sẽ kể một câu chuyện khác nhau. Khoảng thời gian càng ngắn thì đường trung bình động càng nhạy cảm với những thay đổi về giá. Khoảng thời gian càng dài thì đường trung bình động càng ít nhạy cảm hoặc càng được làm mịn. Đường trung bình động được sử dụng để nhấn mạnh hướng của xu hướng và làm mịn những biến động về giá và khối lượng hoặc "nhiễu" có thể gây nhầm lẫn cho cách diễn giải.

Các nhà đầu tư khác nhau sử dụng đường trung bình động vì những lý do khác nhau. Trong khi một số người sử dụng đường trung bình động như một công cụ phân tích chính, những người khác chỉ sử dụng đường trung bình động như một công cụ xây dựng sự tự tin để hỗ trợ cho các quyết định đầu tư của họ. Sau đây là hai chiến lược khác mà mọi người sử dụng đường trung bình động:

Bộ lọc

Bộ lọc được sử dụng để tăng sự tự tin của bạn về một chỉ báo. Không có quy tắc hoặc điều gì cố định cần chú ý khi lọc, chỉ cần bất cứ điều gì khiến bạn đủ tự tin để đầu tư tiền của mình. Ví dụ, bạn có thể muốn đợi cho đến khi một chứng khoán vượt qua đường trung bình

động của nó và cao hơn ít nhất 10% so với đường trung bình để đảm bảo rằng đó là một sự giao thoa thực sự. Hãy nhớ rằng, việc đặt phần trăm quá cao có thể dẫn đến "bỏ lỡ cơ hội" và mua cổ phiếu khi giá đạt đỉnh.

Một bộ lọc khác là đợi một hoặc hai ngày sau khi chứng khoán vượt qua, điều này có thể được sử dụng để đảm bảo rằng sự gia tăng của chứng khoán không phải là sự may rủi hoặc không bền vững. Một lần nữa, nhược điểm là nếu bạn đợi quá lâu thì bạn có thể bỏ lỡ một số khoản lợi nhuận lớn.

Giao thoa

Sử dụng Giao thoa không dễ như lọc. Có một số loại giao thoa khác nhau, nhưng tất cả đều liên quan đến hai hoặc nhiều đường trung bình động. Trong giao thoa kép, bạn đang tìm kiếm một tình huống mà MA ngắn nhất vượt qua MA dài hơn. Điều này hầu như luôn được coi là tín hiệu mua vì đường trung bình dài hơn là một mức hỗ trợ cho giá cổ phiếu.

Để có thêm sự bảo hiểm, bạn có thể sử dụng giao cắt ba, theo đó đường trung bình động ngắn nhất phải đi qua hai đường cao hơn. Đây được coi là chỉ báo mua mạnh hơn nữa.

Kênh hồi quy

Hồi quy tuyến tính nghe có vẻ đáng sợ, nhưng khái niệm toán học thì rất đơn giản. Tất cả những gì kỹ thuật này làm là tạo một đường thẳng qua một số điểm dữ liệu hữu hạn bằng cách giảm thiểu tổng bình phương khoảng cách theo chiều dọc giữa đường và từng điểm. Trong bối cảnh của chúng tôi, điều này có nghĩa là nếu thời gian được biểu diễn bằng ngày trên trục ngang và giá đóng cửa

vào những ngày đó được biểu thị dưới dạng các chấm trên trục dọc (biểu đồ giá đóng cửa thông thường), thì chúng tôi cố gắng tạo một đường thẳng qua các chấm giá đóng cửa đó sao cho tổng bình phương khoảng cách theo chiều dọc giữa mỗi giá đóng cửa và đường được giảm thiểu. Khi đó, đây sẽ là đường phù hợp nhất của chúng tôi.

Kênh hồi quy Raff Kênh hồi quy Raff cho thấy phạm vi giá có thể lệch khỏi đường xu hướng hồi quy tuyến tính. Được Gilbert Raff phát triển, kênh hồi quy là một đường nghiên cứu các biểu đồ trực tiếp trên biểu đồ giá. Kênh hồi quy cung cấp một cách định lượng chính xác để xác định xu hướng giá và ranh giới của nó. Kênh hồi quy được xây dựng bằng cách vẽ hai đường song song, cách đều nhau ở trên và dưới đường xu hướng hồi quy tuyến tính.

Khoảng cách giữa các đường kênh đến đường hồi quy là khoảng cách lớn nhất mà bất kỳ mức giá cao hoặc thấp nào cách đường hồi quy.

Kênh hồi quy Raff chứa chuyển động giá, với đường kênh dưới cùng cung cấp hỗ trợ và đường kênh trên cùng cung cấp kháng cự. Giá có thể mở rộng ra ngoài kênh trong một thời gian ngắn. Tuy nhiên, nếu giá vẫn nằm ngoài kênh trong một thời gian dài, sự đảo ngược xu hướng có thể sắp xảy ra.

Fibonacci Retracement

Fibonacci Retracement/Extensions được hiển thị bằng cách đầu tiên vẽ một đường xu hướng giữa hai điểm cực trị. Sau khi chọn công cụ Fibonacci Retracement từ thanh công cụ Draw, một loạt tối đa chín đường ngang sẽ được vẽ ở các mức Fibonacci là 0,0%, 23,6%, 38,2%, 50,0%, 61,8%, 100%, 161,8%, 261,8% và 423,6%. Sau một động thái đáng kể (lên hoặc xuống), giá thường sẽ phục hồi và quay lại một phần đáng kể của động thái ban đầu. Khi giá quay lại, các mức hỗ trợ và kháng cự thường sẽ xuất hiện gần các mức Fibonacci Retracement.

Công cụ Fibonacci retracement/extension hoạt động ở 4 chế độ khác nhau tùy thuộc vào hướng của đường xu hướng được vẽ:

• NE - cung cấp đường thoái lui (kiểu cũ) trong xu hướng tăng

• SE - cung cấp đường thoái lui trong xu hướng giảm

• NW - cung cấp đường mở rộng trong xu hướng tăng

• SW - cung cấp đường mở rộng trong xu hướng giảm

Có thể sử dụng đường xu hướng kiểm soát được vẽ theo kiểu chấm bi để xóa nghiên cứu Fibonacci retracement cùng một lúc bằng cách sử dụng menu nút chuột phải.

Múi giờ Fibonacci

Nghiên cứu Múi giờ Fibonacci bao gồm các đường thẳng đứng ở các khoảng Fibonacci là 1, 2, 3, 5, 8, 13, 21, 34, v.v. Việc giải thích Múi giờ Fibonacci liên quan đến việc tìm kiếm những thay đổi đáng kể về giá gần các đường thẳng đứng.

Dải Bollinger

Dải Bollinger là các đường bao quanh các thanh giá trên biểu đồ. Dải Bollinger được vẽ cách đường trung bình động ngắn hạn đơn giản hai độ lệch chuẩn. Đây là điểm khác biệt chính giữa Dải Bollinger và đường bao. Đường bao được vẽ theo tỷ lệ phần trăm cố định trên và dưới đường trung bình động. Vì độ lệch chuẩn là thước đo độ biến động nên Dải Bollinger tự điều chỉnh theo các điều kiện thị trường. Chúng mở rộng trong các giai đoạn thị trường biến động và co lại trong các giai đoạn ít biến động hơn.

Dải Bollinger trở thành dải độ lệch chuẩn động. Dải Bollinger được hiển thị bằng đường thứ ba. Đây là đường trung bình động đơn giản (ngắn hạn). Khoảng thời gian cho đường trung bình động này có thể thay đổi. Mặc định cho đường trung bình động ngắn hạn trong AmiBroker là 15 ngày.

Một điều quan trọng cần lưu ý là Dải Bollinger không tạo ra tín hiệu mua và bán riêng lẻ. Chúng nên được sử dụng với một chỉ báo khác. Ví dụ, RSI là lựa chọn khá tốt để làm bạn đồng hành với dải Bollinger. Khi giá chạm vào một trong các dải, nó có thể chỉ ra một trong hai điều sau. Nó có thể chỉ ra sự tiếp tục của xu hướng; hoặc nó có thể chỉ ra phản ứng theo cách khác. Vì vậy, Dải Bollinger được sử dụng riêng lẻ không cung cấp tất cả những gì các kỹ thuật viên cần biết. Sau đó, RSI, một chỉ báo tuyệt vời liên quan đến các điều kiện quá mua và quá bán, sẽ giúp ích. Nhìn chung, khi giá chạm vào Dải Bollinger trên và RSI dưới 70, chúng ta có dấu hiệu cho thấy xu hướng sẽ tiếp tục. Ngược lại, khi giá chạm vào Dải Bollinger dưới và RSI trên 30, chúng ta có dấu hiệu cho thấy xu hướng sẽ tiếp tục. Nếu chúng ta gặp phải tình huống giá chạm vào Dải Bollinger trên và RSI trên 70 (có thể tiến tới 80), chúng ta có dấu hiệu cho thấy xu hướng có thể đảo ngược và đi xuống. Mặt khác, nếu giá chạm vào Dải Bollinger dưới và RSI dưới 30 (có thể tiến tới 20), chúng ta có dấu hiệu cho thấy xu hướng có thể đảo ngược và đi lên. Tránh bẫy sử dụng nhiều chỉ báo khác nhau, tất cả đều hoạt động trên cùng một dữ liệu đầu vào. Nếu bạn đang sử dụng RSI với Bollinger Bands, đừng sử dụng MACD. Cả hai đều dựa trên cùng một đầu vào. Bạn có thể cân nhắc sử dụng On Balance Volume hoặc Money Flow. RSI, On Balance Volume và Money Flow dựa trên các đầu vào khác nhau.

Chỉ báo

Chỉ báo là gì?

Chỉ báo là phép tính toán học có thể áp dụng cho trường giá và/hoặc khối lượng của chứng khoán. Kết quả là giá trị được sử dụng để dự đoán những thay đổi trong tương lai về giá.

AmiBroker có các chỉ báo sau được tích hợp sẵn:

- ROC
- RSI
- MACD
- CCI
- OBV
- NVI
- MFI
- Tích lũy/Phân phối
- TRIX
- Chaikin
- Sức mạnh tương đối
- Bộ dao động cuối cùng
- Stochastic
- TRIN (Chỉ số Arms)
- Đường AD-Line (Đường tăng/giảm)
- Biểu đồ khối lượng theo giá (Hồ sơ khối lượng)
- Hiệu suất tương đối

Tích lũy/Phân phối

Tích lũy/Phân phối là một chỉ báo động lượng tính đến các thay đổi về giá và khối lượng cùng nhau. Ý tưởng là sự thay đổi về giá kết hợp với sự gia tăng về khối lượng có thể giúp xác nhận động lượng thị trường theo hướng biến động giá.

Lưu ý sự tương đồng của công thức này với công thức của stochastic; về cơ bản đây là một stochastic nhân với khối lượng. Điều này có nghĩa là nếu chứng khoán đóng cửa ở mức cao nhất, hệ số nhân khối lượng sẽ lớn hơn so với khi chứng khoán đóng cửa gần mức thấp nhất.

Nếu chỉ báo Tích lũy/Phân phối đang tăng lên, người mua đang thúc đẩy biến động giá và chứng khoán đang được tích lũy. Giá trị A/D giảm ngụ ý rằng người bán đang thúc đẩy thị trường và chứng khoán đang được phân phối. Nếu có sự phân kỳ xảy ra giữa chỉ báo Tích lũy/Phân phối và giá của chứng khoán, thì có khả năng giá sẽ thay đổi theo hướng.

Công thức Đường tích lũy/Phân phối như sau:

$$\frac{(CLOSE - LOW) - (HIGH - CLOSE)}{HIGH - LOW} \times VOLUME + I$$

Trong đó I là giá trị Tích lũy/Phân phối của ngày hôm qua.

Đường Advance-Decline (AD-Line)

Đường đo lường mức tăng và giảm phản ánh độ rộng của thị trường. Ở dạng đơn giản nhất, ADLine là tổng hợp theo thời gian của chênh lệch ròng hằng ngày giữa số lượng các vấn đề tăng và số lượng các vấn đề giảm. AmiBroker sử dụng công thức được cải tiến đôi chút, công thức này cũng tính đến số lượng các vấn đề không đổi. Công thức AFL chính xác cho ADLine của AmiBroker là:

```
Difference = ( AdvIssues() - DecIssues() )/ ( UncIssues() + 1 );

DiffSqrt = IIF( Difference > 0, sqrt( Difference ), - sqrt( - Difference ) ); ADLine = Cum( DiffSqrt );
```

Đây là một chỉ báo cổ điển có xu hướng đưa ra kết quả tốt về sức mạnh chung của thị trường. Sự phá vỡ đường A/D thường diễn ra sau sự phá vỡ giá. Hãy tìm kiếm sự không xác nhận và phân kỳ.

Xem thêm tham chiếu hàm AFL: Hàm AFL: adline()

ADX / Chỉ số chuyển động theo hướng

Chỉ báo ADX, hay còn gọi là Chỉ số chuyển động theo hướng.

ADX là một hệ thống theo xu hướng. Chỉ số chuyển động theo hướng trung bình, hay ADX, xác định xu hướng thị trường. Khi được sử dụng với các giá trị chỉ báo hướng lên và xuống, +DI và -DI, ADX là một hệ thống giao dịch chính xác. Giải thích chuẩn cho việc sử dụng ADX (đường màu xanh lam) là thiết lập vị thế mua bất cứ khi nào +DI (đường màu xanh lá cây) cắt lên trên -DI (đường màu đỏ). Bạn đảo ngược vị thế đó, thanh lý vị thế mua và thiết lập vị thế bán khi -DI cắt lên trên +DI. Ngoài các quy tắc giao cắt, bạn cũng phải tuân theo quy tắc điểm cực trị. Khi giao cắt xảy ra, hãy sử dụng giá cực trị làm điểm đảo ngược. Đối với vị thế bán, hãy sử dụng mức cao được tạo ra trong khoảng thời gian giao dịch của giao cắt. Ngược lại, hãy đảo ngược vị thế mua bằng cách sử dụng mức thấp được tạo ra trong khoảng thời gian giao dịch của giao cắt. Bạn duy trì điểm đảo ngược, mức cao hoặc mức thấp, làm giá vào hoặc thoát thị trường của mình ngay cả khi +DI và -DI vẫn giao nhau trong một số khoảng thời gian giao dịch. Điều này được cho là để giúp bạn không bị đảo lộn trên thị trường. Đối với một số nhà giao dịch, công dụng quan trọng nhất của ADX là khái niệm điểm ngoặt. Đầu tiên, ADX phải nằm trên cả hai đường DI.

Khi ADX giảm xuống, thị trường thường đảo ngược xu hướng hiện tại. ADX đóng vai trò là cảnh báo cho thị trường sắp thay đổi hướng. Ngoại lệ chính của quy tắc này là thị trường tăng giá mạnh trong giai đoạn thổi bay. ADX giảm xuống chỉ để tăng cao hơn sau vài ngày. Theo nhà phát triển DMI, bạn nên ngừng sử dụng bất kỳ hệ thống theo xu hướng nào khi ADX nằm dưới cả hai đường DI. Thị trường đang trong phạm vi đi ngang không có xu hướng rõ ràng. Nếu bạn cần giải thích thêm, vui lòng tham khảo tác phẩm gốc của tác giả. Cuốn sách có tựa đề New Concepts in Technical Trading Systems của J. Welles Wilder, Jr. giải thích về chỉ báo này và một số chỉ báo khác.

CCI - Chỉ số kênh hàng hóa

Một chỉ báo động lượng giá do Donald R. Lambert phát triển - chỉ báo này đo lường sự biến động giá so với giá trung bình như một biến thể thống kê. Chỉ báo này hoạt động khá tốt với hàng hóa, cổ phiếu và quỹ tương hỗ. Chỉ báo này giữ cho các giao dịch trung lập trong thị trường đi ngang và giúp tham gia thị trường khi có sự đột phá.

Mô tả về công thức CCI như sau:

Đầu tiên, Tính giá trị trung bình của từng giai đoạn. Đây là giá cao nhất, cộng với giá thấp nhất, cộng với giá đóng cửa, chia cho 3. Thứ hai, tính giá trị trung bình động đơn giản của giai đoạn n của các giá trị trung bình này.

Thứ ba, từ giá trung bình của từng giai đoạn, trừ đi giá trị trung bình động đơn giản của giai đoạn n của giá trung bình.

Thứ tư, Tính độ lệch trung bình. Đây là sự khác biệt giữa giá trung bình của từng giai đoạn và giá trung bình động đơn giản của giai đoạn n của các giá trung bình đó.

Thứ năm, Nhân độ lệch trung bình với .015.

Thứ sáu, giá trung bình, mà chúng ta đã tính toán ở bước ba, được chia cho 0,015 lần độ lệch trung bình từ bước 5.

Thông thường, CCI có giá trị dao động từ +100 đến -100. Các quy tắc là mua và mua dài hạn khi CCI vượt qua

+100 và đóng lệnh mua dài hạn khi CCI giảm xuống dưới +100. Ngược lại, bán khống khi CCI vượt qua

-100 và đóng lệnh bán khống khi CCI vượt qua trở lại trên -100.

Chaikin Oscillator

Được Marc Chaikin phát triển vào đầu những năm 1970 khi giá mở cửa bị loại khỏi nhiều danh sách báo khiến việc tính toán OBV của William trở nên khó khăn hơn. Chaikin đã thay thế giá mở cửa của William bằng giá trung bình [(CAO+THẤP)/2] và tạo ra một bộ dao động sử dụng đường trung bình động hàm mũ 10 kỳ và 3 kỳ của Đường tích lũy/phân phối kết quả.

Tiền đề cơ bản của Đường tích lũy/phân phối là mức độ áp lực mua hoặc bán có thể được xác định theo vị trí đóng cửa, so với mức cao và mức thấp trong kỳ tương ứng. Có áp lực mua khi cổ phiếu đóng cửa ở nửa trên của phạm vi một kỳ và có áp lực bán khi cổ phiếu đóng cửa ở nửa dưới của phạm vi giao dịch của kỳ.

Tín hiệu tăng giá

Có hai tín hiệu tăng giá có thể được tạo ra từ Chaikin Oscillator: phân kỳ dương và giao cắt đường trung tâm. Vì Chaikin Oscillator là một chỉ báo của một chỉ báo, nên thận trọng khi tìm kiếm xác nhận về sự phân kỳ dương, ví dụ như bằng một đường trung bình động tăng giá cắt nhau, trước khi coi đây là tín hiệu tăng giá.

Tín hiệu giảm giá

Trái ngược hoàn toàn với các tín hiệu tăng giá, có hai tín hiệu giảm giá có thể được tạo ra từ Chaikin Oscillator: một phân kỳ âm và một đường trung tâm giảm giá cắt nhau. Cho phép một phân kỳ âm được xác nhận bằng một đường trung tâm giảm giá cắt nhau, trước khi tín hiệu tăng giá được đưa ra.

Chaikin Oscillator có tác dụng tốt trong việc bổ sung động lượng cho Đường tích lũy/phân phối, nhưng đôi khi có thể bổ sung quá nhiều động lượng và khó diễn giải. Các đường trung bình động đều tương đối ngắn và do đó sẽ nhạy cảm hơn với những thay đổi trong Đường tích lũy/phân phối. Độ nhạy là quan trọng, nhưng người ta cũng phải có khả năng diễn giải chỉ báo.

MACD - Đường trung bình động hội tụ/phân kỳ

Chỉ báo này sử dụng ba đường trung bình động hàm mũ, một đường trung bình ngắn hoặc nhanh, một đường trung bình dài hoặc chậm và một đường trung bình hàm mũ của sự khác biệt của chúng, đường cuối cùng được sử dụng làm đường tín hiệu hoặc đường kích hoạt. Để hiểu đầy đủ những điều cơ bản về MACD, trước tiên bạn phải hiểu các đường trung bình động đơn giản. Chỉ báo Đường trung bình động hội tụ/phân kỳ đo cường độ tâm lý của công chúng và được Gerald Appel, người phát triển chỉ báo này, coi là một chỉ báo rất tốt báo hiệu các điểm vào thị trường sau một đợt giảm mạnh. Chỉ báo này cho thấy các điều kiện mua quá mức và bán quá mức và tạo ra các tín hiệu dự đoán xu hướng hoặc sự đảo ngược giá. Chỉ báo này cung cấp phép đo nhạy bén về cường độ tâm lý của công chúng và có thể áp dụng cho thị trường chứng khoán, cổ phiếu riêng lẻ hoặc quỹ tương hỗ. Trong một số trường hợp, chỉ báo này có thể đưa ra cảnh báo trước về sự đảo ngược, cho phép bạn mua vào khi giá yếu và bán ra khi giá mạnh.

Chỉ báo Đường trung bình động hội tụ/phân kỳ (MACD) được tính bằng cách trừ giá trị của

đường trung bình động hàm mũ 26 ngày khỏi đường trung bình động hàm mũ 12 ngày. Đường trung bình động hàm mũ 9 ngày ("đường tín hiệu") được tự động hiển thị ở trên cùng của đường chỉ báo MACD.

Quy tắc giao dịch MACD cơ bản là bán khi MACD giảm xuống dưới đường tín hiệu 9 ngày của nó. Tương tự như vậy, tín hiệu mua xảy ra khi MACD tăng lên trên đường tín hiệu của nó.

Chỉ số dòng tiền

Chỉ số dòng tiền (MFI) cố gắng đo lường sức mạnh của dòng tiền chảy vào và chảy ra khỏi một chứng khoán. Chỉ số này có liên quan chặt chẽ đến Chỉ số sức mạnh tương đối (RSI). Sự khác biệt giữa RSI và dòng tiền là RSI chỉ xem xét giá, trong khi Chỉ số dòng tiền cũng tính đến khối lượng.

Tính toán dòng tiền khó hơn một chút so với RSI.

Trước tiên, chúng ta cần giá trung bình trong ngày, sau đó chúng ta cần Dòng tiền:

$$MoneyFlow = Volume \times AveragePrice$$

Bây giờ, để tính tỷ lệ dòng tiền, bạn cần tách dòng tiền trong một khoảng thời gian thành dương và âm. Nếu giá tăng trong một ngày cụ thể, thì đây được coi là "Dòng tiền tích cực". Nếu giá đóng cửa giảm, thì đây được coi là "Dòng tiền tiêu cực".

$$MoneyRatio = \frac{PositiveMoneyFlow}{NegativeMoneyFlow}$$

Tỷ lệ dòng tiền được sử dụng để tính Chỉ số dòng tiền.

$$MFI = 100 - \frac{100}{100 + MoneyRatio}$$

Dòng tiền dao động từ 0 đến 100. Giống như RSI, một cổ phiếu được coi là mua quá mức trong phạm vi 70-80 và bán quá mức trong phạm vi 20-30.

Số ngày bạn sử dụng càng ngắn thì Dòng tiền càng biến động. Mặc định là sử dụng mức trung bình 14 ngày.

Diễn giải về Chỉ số Dòng tiền như sau:

• Tìm kiếm sự phân kỳ/biến động thất bại giữa chỉ báo và hành động giá. Nếu giá có xu hướng tăng (thấp hơn) và MFI có xu hướng giảm (cao hơn), thì sự đảo ngược có thể sắp xảy ra.

• Tìm kiếm đỉnh thị trường xảy ra khi MFI ở trên một mức cụ thể (ví dụ: 80). Tìm kiếm đáy thị trường xảy ra khi MFI ở dưới một mức cụ thể (ví dụ: 20).

Chỉ số khối lượng âm

Chỉ báo này đưa ra một giả định rất quan trọng. Giả định rằng nhà đầu tư thiếu kinh nghiệm sẽ theo dõi xu hướng thị trường, do đó đẩy khối lượng lên khi họ nhảy vào giá chứng khoán tăng. Mặt khác, việc mua và bán có thông tin của những người "biết chuyện" diễn ra trong những giai đoạn yên tĩnh hơn, thể hiện qua những thay đổi về khối lượng âm vào những ngày khối lượng giảm. Đây là một công cụ dự đoán xu hướng thị trường tăng giá tuyệt vời. Chỉ số này chỉ đơn giản đo lường xu hướng giá trong những giai đoạn khối lượng giảm.

Chỉ số giá chỉ được điều chỉnh vào những ngày khối lượng giảm so với ngày trước đó. Nếu khối lượng không thay đổi hoặc là dương, chỉ báo vẫn không đổi. Nếu chỉ số tăng, điều đó có nghĩa là giá chứng khoán đã tăng vào ngày khối lượng giảm. Chỉ số giảm cho biết giá chứng khoán đã giảm trong khi khối lượng giảm. (Sự thay đổi trong chỉ số được tính là phần trăm thay đổi trong giá).

Chỉ số này có thể được so sánh với các mức trung bình trong thời gian dài hơn để phản ánh sự chuyển động của tiền thông minh. Ví dụ, nếu chỉ số hiện tại cao hơn mức trung bình sáu tháng, điều này có thể chỉ ra xu hướng tăng của thị trường hoặc chứng khoán.

OBV - Khối lượng cân bằng

OBV được tạo ra bởi Joe Granville, cha đẻ của phân tích OBV. Đây là tổng khối lượng đang chạy liên quan đến thay đổi giá và khối lượng và cho thấy hành động tích lũy và phân phối.

OBV cổ điển được tính bằng cách cộng tổng khối lượng hôm nay với tổng tích lũy khi giá đóng cửa cao hơn giá đóng cửa hôm qua và trừ tổng khối lượng hôm nay khỏi tích lũy khi giá đóng cửa thấp hơn giá đóng cửa hôm qua. Nếu giá vẫn giữ nguyên, thì OBV không thay đổi. Lượng thay đổi giá thực tế không liên quan và chỉ có hướng thay đổi mới có ý nghĩa đối với các phép tính này.

Chỉ báo này xác định xu hướng bằng cách hiển thị sức mạnh cơ bản của biến động giá theo thời gian. Một xu hướng giá vững chắc được cho là đi kèm với biến động khối lượng mạnh hơn theo cùng một hướng. Phân tích OBV cho rằng xu hướng khối lượng dẫn đến xu hướng giá và các thay đổi của OBV thường đi trước các thay đổi giá. Tìm kiếm sự phân kỳ hoặc không xác nhận giữa biến động giá và khối lượng. Một cổ phiếu có xu hướng tăng và bắt đầu có khối lượng cao hơn vào những ngày giá đóng cửa thấp hơn thường cho thấy xu hướng hiện tại đã kết thúc. Tìm kiếm sự thay đổi hoặc đột phá trong xu hướng OBV. Bán khống khi OBV đột phá xuống và mua dài khi OBV đột phá lên.

Parabolic SAR (Dừng và đảo ngược)

Được phát triển bởi Welles Wilder, người sáng tạo ra RSI và DMI, Parabolic SAR thiết lập các điểm dừng giá theo sau cho các vị thế mua hoặc bán. Còn được gọi là chỉ báo dừng và đảo ngược (SAR là viết tắt của "dừng và đảo ngược"), Parabolic SAR phổ biến hơn để thiết lập các điểm dừng hơn là để thiết lập hướng hoặc xu hướng. Wilder khuyến nghị thiết lập xu hướng trước, sau đó giao dịch với Parabolic SAR theo hướng của xu hướng. Nếu xu hướng tăng, hãy mua khi chỉ báo di chuyển xuống dưới giá. Nếu xu hướng giảm, hãy bán khi chỉ báo di chuyển lên trên giá.

Công thức khá phức tạp, nhưng cách giải thích tương đối đơn giản. Các đường chấm bên dưới giá thiết lập điểm dừng theo sau cho vị thế mua và các đường bên trên thiết lập điểm dừng theo sau cho vị thế bán. Khi bắt đầu di chuyển, Parabolic SAR sẽ cung cấp một đệm lớn hơn giữa giá và điểm dừng theo sau. Khi động thái diễn ra, khoảng cách giữa giá và chỉ báo sẽ thu hẹp lại, do đó tạo ra lệnh dừng lỗ chặt chẽ hơn khi giá di chuyển theo hướng thuận lợi.

Có hai biến: bước và bước tối đa. Bước càng cao, chỉ báo sẽ càng nhạy cảm với những thay đổi về giá. Nếu bước được đặt quá cao, chỉ báo sẽ dao động trên và dưới giá quá thường xuyên, khiến việc giải thích trở nên khó khăn. Bước tối đa kiểm soát việc điều chỉnh SAR khi giá di chuyển. Bước tối đa được đặt càng thấp, lệnh dừng lỗ sẽ càng xa giá. Wilder khuyến nghị đặt bước ở mức .02 và bước tối đa ở mức .20.

RS - Sức mạnh tương đối (so sánh)

So sánh xu hướng hiệu suất của một cổ phiếu hoặc nhóm ngành so với một cổ phiếu, nhóm hoặc chỉ số khác. So sánh này loại bỏ cảm xúc khỏi thị trường. Nhiều lần, sự sụt giảm sức mạnh tương đối có thể chỉ ra sự sụt giảm sắp tới về giá thực tế của chứng khoán. Đừng nhầm lẫn với RSI của Wilders.

Khái niệm này là để xác định cổ phiếu hoặc lĩnh vực thị trường nào đang hoạt động tốt nhất. Giả sử rằng xu hướng sẽ tiếp tục tồn tại trong một thời gian, thì khả năng cao là trước khi giá cổ phiếu giảm mạnh, trước tiên nó sẽ mất sức mạnh tương đối so với các cổ phiếu khác. Điều này sẽ chỉ ra một lệnh bán trước khi giá giảm như vậy. Sự gia tăng sức mạnh tương đối không nhất thiết chỉ ra rằng chỉ số đang tăng, nhưng nó báo hiệu cảnh báo mua.

RSI - Chỉ số sức mạnh tương đối

Một chỉ báo kỹ thuật do Welles Wilder phát triển để giúp các nhà đầu tư đánh giá sức mạnh hiện tại của giá chứng khoán so với hiệu suất trước đây của nó. RSI là một chỉ báo quá mua/quá bán tuyệt vời có thể được sử dụng để dự đoán các điểm đảo ngược xu hướng. Đừng nhầm lẫn chỉ số này với sức mạnh tương đối theo định nghĩa hàng ngày của nó khi được sử dụng để so sánh chuyển động của một chứng khoán, chỉ số hoặc nhóm với chuyển động của một chứng khoán, chỉ số hoặc nhóm khác. Được phát triển bởi J. Welles Wilder, Jr. và được mô tả lần đầu tiên trong cuốn sách "New Concepts in Technical Trading Systems" của ông, đây là một bộ dao động động lượng đo tốc độ chuyển động giá theo hướng.

Nó so sánh mức cao nhất và mức thấp nhất của chứng khoán trong một khoảng thời gian. RSI dựa trên sự khác biệt giữa giá đóng cửa trung bình vào những ngày tăng so với giá đóng cửa trung bình vào những ngày giảm.

RSI=100-[100/(1+U/D)]

U = giá trung bình đóng cửa tăng (EMA của mức tăng)

D = giá trung bình đóng cửa giảm (EMA của mức giảm)

Tỷ lệ giữa giá trung bình đóng cửa tăng và giảm thực chất là thành phần của chỉ số. Khoảng thời gian được chỉ định sẽ quyết định mức độ biến động của RSI. Ví dụ: khoảng thời gian 9 ngày sẽ biến động hơn khoảng thời gian 21 ngày. Tác giả (Wilder) sử dụng giá trị n là 14 ngày nhưng có thể sử dụng các giá trị khác phù hợp hơn với các chứng khoán cụ thể. RSI 9 ngày và 25 ngày cũng đã trở nên phổ biến. Vì bạn có thể thay đổi số khoảng thời gian trong phép tính RSI, tôi khuyên bạn nên thử nghiệm để tìm ra khoảng thời gian phù hợp nhất với mình.

RSI là một bộ dao động theo giá dao động trong khoảng từ 0 đến 100. Một phương pháp phổ biến để phân tích RSI là tìm kiếm sự phân kỳ trong đó chỉ số thị trường đang đạt mức cao mới, nhưng RSI không vượt qua được mức cao trước đó. Sự phân kỳ này sẽ là dấu hiệu của một sự đảo ngược sắp xảy ra. Khi RSI sau đó quay xuống và giảm xuống dưới mức đáy gần đây nhất của nó, thì nó được cho là đã hoàn thành một cú đảo chiều thất bại. Cú đảo chiều thất bại sẽ được coi là sự xác nhận của một sự đảo ngược sắp xảy ra.

Trong cuốn sách của ông Wilder, ông thảo luận về năm cách sử dụng RSI trong việc phân tích biểu đồ hàng hóa (những cách này cũng áp dụng cho các chỉ số):

1. Đỉnh và Đáy: Chỉ số RSI trên 70 cho biết cổ phiếu bị mua quá mức và có khả năng bắt đầu giảm. Chỉ số dưới 30 cho biết cổ phiếu bị bán quá mức và có thể mong đợi một đợt

tăng giá. (AmiBroker tự động vẽ các đường ngang ở các mức này). RSI thường hình thành các đỉnh và đáy này trước biểu đồ giá cơ bản.

2. Hình thành biểu đồ: RSI thường hình thành các mô hình biểu đồ (như đầu và vai hoặc nêm tăng) có thể hoặc không thể nhìn thấy trên biểu đồ giá.

3. Cú đảo chiều thất bại: Đây là nơi RSI vượt qua mức cao trước đó (đỉnh) hoặc giảm xuống dưới mức thấp gần đây (đáy).

4. Hỗ trợ và kháng cự: RSI đôi khi cho thấy rõ hơn biểu đồ giá về mức hỗ trợ và kháng cự.

5. Phân kỳ: Như đã thảo luận ở trên, điều này xảy ra khi giá tạo ra mức cao (hoặc thấp) mới mà không được xác nhận bởi mức cao (hoặc thấp) mới của RSI.

ROC - Tỷ lệ thay đổi giá

Chỉ báo này hiển thị tỷ lệ thay đổi giá của một chứng khoán. Thay đổi được hiển thị dưới dạng phần trăm chứ không phải tỷ lệ.

ROC được tính bằng cách chia mức thay đổi giá trong n kỳ gần nhất cho giá đóng cửa cách đây n kỳ. Điều này cho bạn biết tỷ lệ phần trăm giá đã thay đổi trong n kỳ gần nhất.

Khi đường ROC 10 ngày nằm trên đường trung tâm, giá hôm nay cao hơn so với 10 kỳ trước. Khi đường ROC nằm dưới đường trung tâm, giá hôm nay thấp hơn so với 10 ngày trước. Nếu đường ROC nằm trên đường trung tâm, giá cao hơn so với 10 ngày trước. Nếu đường ROC nằm dưới đường trung tâm nhưng đang tăng, giá hiện tại vẫn thấp hơn so với 10 ngày trước, nhưng phạm vi đang thu hẹp lại.

ROC 12 ngày được sử dụng tốt nhất như một chỉ báo quá mua/quá bán ngắn hạn đến trung hạn. ROC càng cao, chứng khoán càng bị mua quá mức; ROC càng thấp, khả năng tăng giá càng cao. Tuy nhiên, giống như tất cả các chỉ báo quá mua/quá bán, tốt nhất là đợi thị trường bắt đầu điều chỉnh (tức là tăng hoặc giảm) trước khi thực hiện giao dịch của bạn. Một thị trường có vẻ quá mua có thể vẫn quá mua trong một thời gian. Trên thực tế, các chỉ số quá mua/quá bán cực độ thường ngụ ý sự tiếp tục của xu hướng hiện tại.

ROC 12 ngày có xu hướng rất theo chu kỳ, dao động qua lại theo một mô hình khá đều đặn. Thông thường, có thể dự đoán được những thay đổi về giá bằng cách nghiên cứu các chu kỳ trước của ROC và liên hệ các chu kỳ trước đó với thị trường hiện tại.

Mức mua quá mức/bán quá mức tối ưu (ví dụ: +/-5) sẽ thay đổi tùy thuộc vào chứng khoán đang được phân tích và điều kiện chung của thị trường. Trong thị trường tăng giá mạnh, thường có lợi khi sử dụng các mức cao hơn, có thể là +10 và -5.

Stochastic Slow

Stochastic là một bộ dao động đo vị thế của một cổ phiếu hoặc chứng khoán so với phạm vi giao dịch gần đây của nó, cho biết tình trạng quá mua hoặc quá bán.

Nó hiển thị giá hiện tại theo tỷ lệ phần trăm so với phạm vi giao dịch của chứng khoán (cao/thấp) trong khoảng thời gian được chỉ định.

$$FastStoc = \%K = \frac{(\text{today's close}) - (\text{low price in period n})}{(\text{high price in period n}) - (\text{low price in period n})}$$

Trong Stochastic Slow, mức cao và mức thấp được tính trung bình trong một khoảng thời gian chậm lại. Mặc định thường là 3 cho chậm và 1 (không chậm lại) cho nhanh. Sau đó, đường này có thể được làm mịn bằng cách sử dụng trung bình động hàm mũ, có trọng số hoặc trung bình động đơn giản %D. Có thể đọc các tín hiệu Mua/bán xác nhận tại các giao điểm của %D với %K.

Bộ dao động Stochastic luôn dao động trong khoảng từ 0% đến 100%. Giá trị đọc là 0% cho thấy giá đóng cửa của chứng khoán là mức giá thấp nhất mà chứng khoán đã giao dịch trong x khoảng thời gian trước đó. Chỉ số 100% cho thấy giá đóng cửa của chứng khoán là mức giá cao nhất mà chứng khoán đã giao dịch trong x giai đoạn thời gian trước đó. Khi giá đóng cửa gần đỉnh của phạm vi giao dịch gần đây (trên 80%), chứng khoán đang trong tình trạng mua quá mức và có thể báo hiệu khả năng điều chỉnh. Tình trạng bán quá mức tồn tại ở điểm dưới %20. Giá đóng cửa gần đỉnh của phạm vi trong xu hướng tăng và gần đáy của phạm vi trong xu hướng giảm

TRIN - Chỉ số Arms

Chỉ số giao dịch, một thước đo kỹ thuật về sự tiến triển và suy giảm trong thị trường. TRIN tính đến số lượng và khối lượng các vấn đề tăng giá, và số lượng và khối lượng các vấn đề giảm giá. Chỉ số này đo lường sức mạnh tương đối của khối lượng liên quan đến các cổ phiếu tăng giá so với sức mạnh của khối lượng liên quan đến các cổ phiếu giảm giá.

Công thức AFL chính xác cho TRIN là:

```
ArmsIndex = ( AdvIssues() / DecIssues() ) / ( AdvVolume() / DecVolume ) );
```

Giá trị TRIN bằng 1 cho biết tỷ lệ khối lượng tăng so với khối lượng giảm bằng với tỷ lệ các vấn đề tăng so với các vấn đề giảm giá và thị trường ở trạng thái trung lập. Trạng thái trung lập chỉ có nghĩa là khối lượng tăng được phân bổ đều cho các vấn đề tăng giá và khối lượng giảm được phân bổ đều cho các vấn đề giảm giá trong ngày.

Chỉ báo này, mặc dù có công thức đơn giản, nhưng cần phải nghiên cứu nhiều khi áp dụng. Có nhiều biến thể được áp dụng cho TRIN. Nhiều nhà phân tích sử dụng đường trung bình động 10 ngày của TRIN làm chỉ báo.

AmiBroker vẽ hai đường trung bình khác nhau cho TRIN với các khoảng thời gian trung bình mặc định là 15 và 45. Giá trị đọc nhỏ hơn 1.0 thường biểu thị nhu cầu tăng giá trong khi giá trị đọc lớn hơn 1 có thể biểu thị tình trạng thị trường giảm giá. Cần lưu ý rằng hành vi của chỉ báo và cách đọc cũng như cách diễn giải của nó phụ thuộc vào việc thị trường đang trong giai đoạn tăng giá hay giảm giá. Khoảng thời gian thực tế của giai đoạn thị trường này cũng phải được xem xét. Không cố gắng đưa ra quyết định mua hoặc bán dựa trên các biến động của chỉ báo này.

Xem thêm tham chiếu về Hàm AFL: Hàm AFL: trin()

TRIX - TRIple eXponential

TRI-ple eXponential. TRIX hiển thị tỷ lệ phần trăm thay đổi của đường trung bình động được làm mịn theo hàm mũ ba lần của giá đóng cửa của một chứng khoán.

TRIX được tính là tỷ lệ thay đổi một kỳ của đường trung bình động hàm mũ thứ ba của giá đóng cửa.

TRIX được thiết kế để lọc ra các chu kỳ không đáng kể - những chu kỳ nhỏ hơn số lượng đường trung bình động được chỉ định. Chỉ báo TRIX dao động quanh một đường bằng không. Các giao dịch nên được thực hiện khi chỉ báo thay đổi hướng.

Ultimate Oscillator

Larry Williams, nhà thiết kế của Ultimate Oscillator, muốn giải quyết các vấn đề gặp phải với hầu hết các bộ dao động khi sử dụng trong các khoảng thời gian khác nhau.

Các tín hiệu của Ultimate Oscillator như sau: phân kỳ và đột phá trong xu hướng của Bộ dao động, cũng như các mức quá mua và quá bán.

Giá trị của các bộ dao động khác có thể thay đổi rất nhiều tùy thuộc vào số chu kỳ thời gian được sử dụng trong quá trình tính toán. Vì vậy, Ultimate Oscillator sử dụng tổng trọng số của ba bộ dao động biểu thị các chu kỳ thị trường ngắn hạn, trung hạn và dài hạn (7, 14 và 28 kỳ) và được biểu thị dưới dạng một đường thẳng trên thang đo dọc từ 0 đến 100.

Ba thành phần này dựa trên định nghĩa của Williams về "áp lực" mua và bán.

Nên bắt đầu giao dịch sau khi phân kỳ và đột phá trong xu hướng của Ultimate Oscillator.

Tín hiệu:

Tín hiệu mua được tạo ra khi:

Xảy ra sự phân kỳ dương hoặc tăng giá giữa bộ dao động Ultimate và giá.

Ultimate giảm xuống dưới 30 rồi tăng lên trên mức cao trước đó được thiết lập trong quá trình phân kỳ (tín hiệu mua thực tế).

Tín hiệu bán được đưa ra khi:

Xảy ra sự phân kỳ âm hoặc giảm giá giữa Ultimate và giá.

Ultimate tăng lên trên 70 rồi giảm xuống dưới mức thấp trước đó được thiết lập trong quá trình phân kỳ (tín hiệu bán thực tế).

Đóng các vị thế hiện có:

Đóng các vị thế mua khi Ultimate vượt quá 70. Đóng các vị thế bán khi Ultimate xuống dưới 30.

Cũng như hầu hết các chỉ báo, tốt nhất là các tín hiệu này được xác nhận bởi các chỉ báo khác trước khi được tác động

VAP - Biểu đồ khối lượng theo giá

Biểu đồ khối lượng theo giá còn được gọi là biểu đồ "Hồ sơ khối lượng".

Để bật, chỉ cần vào Công cụ->Tùy chọn và thay đổi Loại VAP từ "NONE" thành "Biểu đồ vùng rắn bên trái, phía sau" chẳng hạn

VAP hiển thị tổng khối lượng giao dịch diễn ra ở mức giá nhất định. VAP được tính toán từ các thanh dữ liệu hiện đang hiển thị.

Thuật toán thực tế không liên quan đến MỘT mức giá mà là PHẠM VI giá Cao-Thấp.

AmiBroker PHÂN PHỐI đều khối lượng của thanh trên phạm vi Cao-Thấp để tạo biểu đồ VAP. Ví dụ, nếu thể tích của thanh là 10000 và phạm vi H-L trải dài 3 "dòng" của biểu đồ VAP thì mỗi dòng trong số 3 dòng

liên quan sẽ được thêm 10000/3 để tạo số liệu thống kê. Điều này mang lại kết quả chính xác hơn nhiều so với việc sử dụng giá đơn lẻ

như một số triển khai khác.

Để bật/tắt VAP, hãy sử dụng: Tools->Preferences->Main chart

Bạn cũng có thể thêm VAP vào biểu đồ tùy chỉnh của riêng mình bằng hàm PlotVAPOverlay AFL.

Biểu đồ hiệu suất tương đối

Biểu đồ hiệu suất tương đối so sánh tỷ lệ thay đổi giá của hai hoặc nhiều công cụ có thể giao dịch. Biểu đồ bắt đầu với 0% tại thanh đầu tiên hiển thị và hiển thị phần trăm thay đổi của giá đóng cửa kể từ thời điểm đó cho mọi ký hiệu trong danh sách. Biểu đồ hiệu suất tương đối rất tuyệt vời để so sánh các vấn đề có giá không giống nhau (ví dụ: cổ phiếu và chỉ số) vì nó hiển thị phần trăm thay đổi chứ không phải giá trị tuyệt đối. Bạn có thể dễ dàng xem công cụ nào hoạt động tốt hơn những công cụ khác và chọn công cụ hoạt động tốt nhất cho giao dịch của mình.

Bạn có thể điều chỉnh danh sách các ký hiệu được vẽ trong biểu đồ hiệu suất tương đối bằng cách nhấp chuột PHẢI vào biểu đồ và chọn mục "Tham số" từ menu ngữ cảnh. Trong hộp thoại Tham số, bạn có thể nhập danh sách các ký hiệu được phân tách bằng dấu phẩy mà bạn muốn lấy biểu đồ. Không có giới hạn về số lượng ký hiệu bạn có thể nhập, nhưng hãy nhớ phân tách các ký hiệu bằng dấu phẩy và không sử dụng khoảng trắng trừ khi chính ký hiệu đó có chúng. Biểu đồ hiệu suất tương đối

Ngôn ngữ công thức Amibroker (Afl)

AmiBroker được trang bị một ngôn ngữ công thức mạnh mẽ, cho phép bạn viết các quy tắc hệ thống giao dịch, xác định các chỉ số của riêng bạn và các bình luận tùy chỉnh. Chương này giải thích ngôn ngữ, cung cấp cho bạn tài liệu tham khảo chi tiết của các hàm phân tích tích hợp và hướng dẫn cách sử dụng các công cụ AFL như trình phân tích tự động và trình soạn thảo công thức.

- Tham khảo ngôn ngữ
  ♦ Các yếu tố cơ bản (các phần tử từ vựng, các biến được định trước)
  ♦ Từ khóa
- Tham khảo hàm
  ♦ Danh sách theo thứ tự bảng chữ cái của tất cả các hàm AFL
  ♦ Danh sách được phân loại theo hàm AFL
  ♦ Hàm AddToComposite - tạo số liệu thống kê bảo mật nhiều lần
  ♦ Hàm Equity - phân tích hiệu suất hệ thống giao dịch của bạn
  ♦ Các hàm với chu kỳ biến đổi
- Các hàm do người dùng xác định và phạm vi biến
- Công cụ AFL
- Máy chủ kịch bản AFL
- Hỗ trợ mô hình đối tượng thành phần trong AFL
- Lỗi lập trình phổ biến
- Giao diện backtester danh mục đầu tư nâng cao
- Thêm số liệu backtester tùy chỉnh
- Sử dụng các hàm đồ họa cấp thấp
  Cũng xem: Hướng dẫn: Hiểu cách AFL hoạt động

Giới Thiệu Sổ Tay Tham Chiếu AFL

AFL là một ngôn ngữ lập trình đặc biệt được sử dụng để định nghĩa và tạo các chỉ số tùy chỉnh, quét, sự khám phá, kiểm tra lại và bình luận của chuyên gia.

Những Yếu Tố Cơ Bản Của Ngôn Ngữ

Yếu tố từ vựng

Chương này mô tả các loại đơn vị giống từ (tokens) khác nhau được công nhận bởi trình thông dịch ngôn ngữ AFL.

Khoảng Trắng

Khoảng trắng là tên gọi chung dành cho dấu cách (khoảng trắng), dấu tab, ký tự dòng mới và nhận xét. Khoảng trắng có thể chức năng để chỉ định nơi bắt đầu và kết thúc tokens, nhưng vượt ra khỏi chức năng này, bất kì khoảng trắng dư thừa nào cũng sẽ bị loại bỏ.

Bình Luận

Bình luận là những đoạn văn bản được sử dụng để chú thích cho chương trình. Chúng được theo sau từ mã nguồn trước khi phân tích cú pháp. Có hai cách để phân định bình luận: Bình luận kiểu C và bình luận kiểu C++. Bình luận kiểu C là bất kỳ chuỗi ký tự nào được đặt sau cặp ký hiệu /*. Bình luận kết thúc tại lần xuất hiện đầu tiên của cặp */ theo sau ký hiệu /* ban đầu. Toàn bộ chuỗi, bao gồm bốn ký hiệu phân cách bình luận, được thay thế bằng một khoảng trắng. Bình luận kiểu C++ là bình luận một dòng

bắt đầu bằng cách sử dụng hai dấu gạch chéo liền kề (//) ở bất kỳ vị trí nào trong dòng và kéo dài cho đến dòng mới tiếp theo.

AFL không cho phép nhúng các bình luận.

Token

AFL nhận dạng năm loại tokens:

- identifiers (định danh)
- constants (hằng số)
- string-literals (chuỗi)
- operators (toán tử)
- punctuators (còn được gọi là separators- phân tách)

Định danh là tên tùy ý với mọi độ dài được đặt cho các hàm và biến. Các định danh có thể chứa các chữ cái (a-z, A-Z), ký tự gạch dưới ("_") và các số (0-9). Ký tự đầu tiên phải là một chữ cái. Định danh AFL KHÔNG phân biệt chữ hoa chữ thường.

Hằng số là tokens đại diện cho giá trị số hoặc ký tự cố định.

Chuỗi hằng, còn được biết đến như *chuỗi hằng số*, hình thành một hạng mục đặc biệt của hằng số được sử dụng để xử lý cố định

Chuỗi ký tự được viết dưới dạng một chuỗi bất kỳ số lượng ký tự nào và được bao quanh bởi dấu ngoặc kép:
"This is literally a string"

Chuỗi rỗng (null) được thể hiện như "". Các ký tự bên trong dấu ngoặc kép có thể bao gồm các chuỗi thoát (\n - chuỗi thoát dòng mới).

Một biểu thức hằng số là biểu thức luôn luôn đánh giá thành hằng số. Chúng được đánh giá như các biểu thức thông thường.

Dấu phân cách (còn gọi là dấu phân tách) trong AFL là một trong những ký tự sau:

( ) , ; = .

Dấu ngoặc (mở (và đóng)) nhóm các biểu thức, cô lập các biểu thức điều kiện và chỉ ra gọi hàm và tham số hàm:

```
d = c * ( a + b )        /* ghi đè mức độ ưu tiên bình thường */

a= (b AND c) OR (d AND e)        /* biểu thức điều kiện */

func()          /* gọi hàm không có đối số */
```

Dấu phẩy (,) phân cách các phần tử của danh sách đối số hàm

Dấu chấm phẩy (;) là dấu kết thúc câu lệnh. Bất kỳ biểu thức hợp lệ AFL nào theo sau bởi dấu chấm phẩy được hiểu là một lệnh, được gọi là lệnh biểu thức. Biểu thức được đánh giá và giá trị của nó được loại bỏ (trừ khi trong Guru Commentaries khi giá trị chuỗi được viết vào cửa sổ ngoài)

Dấu chấm (.) là toán tử truy cập thành viên. Nó được sử dụng để gọi phương thức đối tượng COM. Nếu biến myobj chứa đối tượng, sử dụng toán tử chấm, chúng ta có thể gọi các phương thức (hàm) của đối tượng myobj:

```
myobj.Method();
```

Dấu bằng (=) phân cách khai báo biến khỏi danh sách khởi tạo:

```
x = 5;
```

Nó cũng chỉ ra giá trị mặc định cho một tham số (xem mô tả hàm tích hợp):

```
macd( fast = 12; slow = 26 ) /* giá trị mặc định cho các đối số fast và slow */
```

Cấu trúc ngôn ngữ

Mỗi công thức trong AFL bao gồm một hoặc nhiều lệnh biểu thức. Mỗi lệnh PHẢI được kết thúc bằng dấu chấm phẩy (;). Như vậy, bạn có thể chia biểu thức dài thành nhiều dòng (để tăng tính rõ ràng) và AmiBroker vẫn sẽ xem nó như một câu lệnh duy nhất cho đến dấu chấm phẩy kết thúc. Ví dụ:

```
x = ( y + 3 );  /* x được gán giá trị của y + 3 */
x = y = 0;      /* Cả x và y được khởi tạo là 0 */

proc( arg1, arg2 );   /* Gọi hàm, giá trị trả về bị loại bỏ */
y = z = ( f( x ) + 3 );  /* Một biểu thức gọi hàm */
```

Close - MA(Close,9),

MA(Close, 9) - Close);
/ một câu lệnh trên nhiều dòng */*

Identifiers (Định danh)

Identifiers trong AFL được sử dụng để nhận biết các biến và hàm. Có một số Identifier được định nghĩa trước đề cập đến các mảng và hàm được xây dựng sẵn. Quan trọng nhất là Identifier mảng giá. Chúng nhận dạng các trường giá cụ thể mà công thức nên hoạt động trên. Identifier mảng giá hợp lệ là Open, High, Low, Close, Volume, Openint, Average. Identifier mảng giá có thể được viết tắt như được thể hiện trong bảng sau. Lưu ý rằng những biến này không phân biệt viết Hoa-thường.

Long name	Abbreviation	Comment
Open	O	
High	H	
Low	L	
Close	C	
Volume	V	
OpenInt	OI	
Avg	<none available>	(High+Low+Close)/3 - so called "typical price"

Examples of the use of price array identifiers in formulas are shown below.

Các ví dụ về việc sử dụng mã định danh mảng giá trong công thức được hiển thị bên dưới.

MA(Close, 10); IIf(H > Ref(H,-1), MA(H,20), MA(C,20));

Toán tử (Operator)

Toán tử so sánh

Toán tử so sánh được chia thành hai loại:
- quan hệ (<, >, <=, >=)
- bằng nhau (==, !=)

Symbol	Meaning
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to
==	Equal to
!=	Not equal to

Những toán tử này cho kết quả là giá trị đúng (1) hoặc sai (0) sau khi so sánh.

Toán tử gán

Symbol	Meaning
=	Store the value of the second operand in the object specified by the first operand (“simple assignment”).

Biểu tượng Nghĩa

=Lưu giá trị của toán hạng thứ hai vào đối tượng được chỉ định bởi toán hạng thứ nhất ("gán đơn giản").

Toán tử gán gán một giá trị cho một biến: result = expression; nơi result là định danh biến và expression là bất kỳ biểu thức số học, mảng hoặc văn bản. Vì toán tử = hoạt động giống như các toán tử khác, các biểu thức sử dụng nó có một giá trị ngoài việc gán giá trị đó vào biến. Điều này có nghĩa là bạn có thể ghép nối các toán tử gán như sau: j = k = l = 0; j, k, và l bằng không sau khi câu lệnh ví dụ được thực hiện. Chú ý: xin VUI LÒNG KHÔNG nhầm lẫn toán tử gán (=) với kiểm tra bằng (= =) Đây là hai toán tử khác nhau và bạn không được sử dụng gán (=) để kiểm tra sự bằng nhau. if(Name() = "MSFT") // WRONG !!! - toán tử gán biến được sử dụng thay vì kiểm tra bằng nhau { } if(Name() == "MSFT") // CORRECT - toán tử bằng được sử dụng đúng cách { } Đây là một trong những lỗi lập trình thường gặp được liệt kê tại đây.

Toán tử số học Các công thức có thể chứa các toán tử toán học sau đây:

Symbol	Meaning
+	Addition
-	Subtraction (or negative value)
*	Multiplication
/	Division
%	Modulus (or remainder) (AFL 1.7+)
^	Exponentiation (raising to a power)
|	Bit-wise "Or" (AFL 2.1+)
&	Bit-wise "And" (AFL 2.1+)

Biểu tượng Nghĩa

Các công thức sau đây minh họa việc sử dụng các toán tử trong một công thức:
var1 = (H + L) / 2; var2 = MA(C,10)-MA(C,20) / (H + L + C);
var3 = Close + ((1.02 * High)-High);
Toán tử logic

Ký hiệu Ý nghĩa

NOT Logic "Not" - trả về "True" khi toán hạng bằng false

Symbol	Meaning
NOT	Logical "Not" - gives "True" when operand is equal to false
AND	Logical "And" - gives "True" result if BOTH operands are true at the same time
OR	Logical "Or" - gives "True" result if ANY of operands is true

AND Logic "And" - trả về "True" nếu CẢ HAI toán hạng đều đúng cùng một thời điểm

OR Logic "Or" - trả về "True" nếu BẤT KỲ toán hạng nào là đúng

Nếu một công thức đòi hỏi nhiều điều kiện, bạn có thể kết hợp các điều kiện với các toán tử AND và OR. Ví dụ, có thể bạn muốn vẽ một dấu cộng +1 khi MACD lớn hơn không và RSI lớn hơn 70:
Condition = MACD() > 0 AND RSI(14) > 70; Bạn có thể thêm bất kỳ điều kiện nào trong một công thức như bạn thích.

Toán tử gán kết hợp Được giới thiệu trong phiên bản 5.00, các toán tử kết hợp được xác định dưới dạng: destinvar op= expr; nơi destinvar là biến, expr là biểu thức, và op là một trong các toán tử số học sau: +, -, *, /, %, &, | Định dạng destinvar op= expr hoạt động như: destinvar = destinvar op expr; Đây là hình thức tắt cho các câu lệnh gán phổ biến như k = k

+ 2; vì vậy, bạn có thể viết ngắn gọn hơn như: k += 2; và nó sẽ hoạt động giống như vậy nhưng nhanh hơn một chút. Danh sách đầy đủ các toán tử gán có sẵn ở đây:

Không Ký Hiệu Ý Nghĩa

1 = Lưu giá trị của toán hạng thứ hai vào đối tượng được chỉ định bởi toán hạng thứ nhất ("gán đơn giản").

2	*=	Nhân giá trị của toán hạng thứ nhất với giá trị của toán hạng thứ hai; lưu kết quả vào đối tượng được chỉ định bởi toán hạng đầu tiên.
3	/=	Chia giá trị của toán hạng thứ nhất cho giá trị của toán hạng thứ hai; lưu kết quả vào đối tượng được xác định bởi toán hạng đầu tiên.

No	Symbol	Meaning
1	=	Store the value of the second operand in the object specified by the first operand (“simple assignment”).
2	*=	Multiply the value of the first operand by the value of the second operand; store the result in the object specified by the first operand.
3	/=	Divide the value of the first operand by the value of the second operand; store the result in the object specified by the first operand.

No	Symbol	Meaning
1	++	Post-increment/pre-increment (i++ works like i = i + 1)
2	--	Post-decrement/pre-decrement (i-- works like i = i - 1)
3	[ ]	Array element (subscript) operator
4	^	Exponentiation
5	-	Negation - Unary minus
6	*	Multiplication
7	/	Division
8	%	Reminder (Modulo operator)
9	+	Addition
10	-	Subtraction
11	<	Less than
12	>	Greater than
13	<=	Less than or equal to
14	>=	Greater than or equal to
15	==	Equal to
16	!=	Not equal to
17	&	Bit-wise "And" (AFL 2.1+)
18	|	Bit-wise "Or" (AFL 2.1+)
19	NOT	Logical "Not"
20	AND	Logical "And"
21	OR	Logical "Or"
22	=	Variable assignment operator
23	*= /= %= += -= &= |=	Compound assignment

Chương trình sẽ tiếp tục thực thi, nhưng nó có thể thay đổi giữa các lần thực thi khi có thêm bars, yếu tố phóng đại thay đổi hoặc biểu tượng thay đổi. Để nhận được bar đầu tiên, bạn có thể sử dụng array[0], để nhận được bar cuối cùng của mảng, bạn có thể sử dụng array[BarCount - 1]. Ví dụ như sau:

- Close[5]; Đại diện cho phần tử thứ sáu (bar) của mảng đóng.
- Close[0]; Đại diện cho bar đầu tiên có sẵn của mảng đóng.

- High[BarCount - 1]; Đại diện cho bar cuối cùng của mảng High.

Ma trận và Các toán tử Ma trận

Một ma trận là mảng hai chiều của các số.

Để tạo một ma trận, hãy sử dụng:

my_var_name = Matrix(rows, cols, initvalue);

Để truy cập vào các phần tử của ma trận, hãy sử dụng:

my_var_name[row][col];

ở đây row là chỉ số hàng (0... số lượng hàng-1)

và col là chỉ số cột (0... số lượng cột-1).

Ma trận và các phần tử của ma trận hỗ trợ tất cả các phép toán số học và logic phần tử. Tất cả các toán tử chuẩn này đều được thực hiện theo từng phần tử của ma trận.

Vì lý do đó, ví dụ để cộng hai ma trận, chúng phải có cùng kích thước (số hàng và số cột phải giống nhau). Nếu chúng không giống nhau, quyết định về việc thực hiện phép toán trên mỗi phần tử thông qua vòng lặp là do bạn. Vì vậy, bạn có thể thêm, trừ, nhân, chia hai ma trận nếu chúng có cùng kích thước bằng một hàm gọi.

x = Matrix(5, 6, 9); // ma trận 5 hàng 6 cột, giá trị ban đầu là 9
y = Matrix(5, 6, 10); // ma trận 5 hàng 6 cột, giá trị ban đầu là 10
z = y - x; // sẽ cho bạn ma trận 5 hàng và 6 cột được điền với các phần tử có giá trị 1 (sự khác biệt giữa 10 và 9).

Tất cả các phép toán này đều được thực hiện theo từng PHẦN TỬ. Bạn cũng có thể áp dụng bất kỳ phép toán số học và logic nào lên ma trận VÀ giá trị vô hướng. Điều này sẽ thực hiện theo từng phần tử của ma trận.

Thao tác trên từng phần tử của ma trận nguồn và giá trị scalar cho trước.

```
m = Matrix( 10, 10, 0 ); // m sẽ là ma trận 10x10 được điền bằng số không
z = m; // bây giờ z cũng là một ma trận
for( i = 0; i < 10; i++ )
{
   z[ i ][ 4 ] = i; // điền z với một số giá trị khác, lưu ý rằng m sẽ không bị ảnh hưởng.
}
for( i = 0; i < 10; i++ )
 _TRACEF( "%g = %g, %g, %g\n", i, m[i][1], m[ i][4], z[ i][4]); // phép cộng scalar (từng phần tử)
z += 3;
m += 5;
for( i = 0; i < 10; i++ )
 _TRACEF( "%g = %g, %g, %g\n", i, m[i][1], m[ i][4], z[ i][4]);
```

Ký hiệu của phép toán tích ma trận là @ (dấu 'at'). Tích ma trận là cách của đại số tuyến tính để nhân các ma trận.

Nếu bạn viết C = A @ B, nó nhân ma trận A(n,k) với ma trận B(k,m) để tạo ra ma trận C(n,m) nên số cột trong ma trận A phải bằng số dòng trong ma trận B.

Để biết thêm thông tin, xem tại: Matrix Multiplication Wikipedia

Ưu tiên của phép toán tích ma trận @ giống như * (vì vậy nó có độ ưu tiên cao hơn phép cộng và trừ).

```
A = Matrix( 1, 3 );
B = Matrix( 3, 2 );
// ma trận A = [ 1, 4, 6 ]
// ma trận B =
// [ 2, 3 ]
// [ 5, 8 ]
// [ 7, 9 ]
A[ 0 ][ 0 ] = 1;  A[ 0 ][ 1 ] = 4;  A[ 0 ][ 2 ] = 6;
B[ 0 ][ 0 ] = 2; B[ 0 ][ 1 ] = 3; B[ 1 ][ 0 ] = 5; B[ 1 ][ 1 ] = 8; B[ 2 ][ 0 ] = 7; B[ 2 ][ 1 ] = 9;
X = A @ B;
_TRACEF("%g %g", X[ 0 ][ 0 ], X[ 0 ][ 1 ] );
```

Compound statements (Blocks)

Lệnh ghép bao gồm không hoặc nhiều lệnh được đóng trong dấu ngoặc nhọn ({ }). Lệnh ghép có thể được sử dụng bất kỳ nơi nào một lệnh được yêu cầu. Lệnh ghép thường được gọi là khối lệnh.

"Khối lệnh."
{
Lệnh1; .... LệnhN;
}
(Phần này được 'mượn' từ ngôn ngữ C, người dùng của các ngôn ngữ lập trình khác thường dùng BEGIN thay cho { và END thay cho })

Nếu số Amount lớn hơn 100
if(Amount > 100)
{
_TRACE("Amount above 100");
Balance = Balance + Amount; } else Balance = Balance - Amount;

Các Hàm Dựng Sẵn

Không chỉ bao gồm các toán tử toán học, AmiBroker còn chứa hơn 70 hàm được xây dựng sẵn, thực hiện các phép toán toán học.

Công thức sau đây bao gồm một hàm đơn lẻ, cung cấp căn bậc hai của giá đóng cửa:
sqrt(**Close**);

Công thức sau đây bao gồm một hàm đơn lẻ, cung cấp chỉ số RSI trong 14 khoảng:
Graph0 = RSI(14);

Công thức sau đây gồm hai hàm. Kết quả là sự khác biệt giữa chỉ số MACD và một trung bình di chuyển hàm mũ trong 9 khoảng của MACD:
Graph0 = MACD() - EMA(MACD(),9);

Tất cả các lệnh gọi hàm phải gồm tên định danh function (tên) theo sau bởi một cặp dấu ngoặc.

Như đã ám chỉ trong các ví dụ trước, một hàm có thể được "gắn vào" trong một hàm. Hàm gắn vào có thể phục vụ như tham số mảng dữ liệu của hàm chính. Các ví dụ sau đây cho thấy các hàm được gắn vào trong các hàm:
MA(RSI(15), 10);
MA(EMA(RSI(15), 20), 10);
Ví dụ đầu tiên tính toán một trung bình đơn giản với 10 khoảng của Chỉ số Sức mạnh Tương đối (RSI) trong 15 khoảng.

Ví dụ thứ hai tính toán một giá trị trung bình di động mũ 20 chu kỳ của chỉ số RSI 15 chu kỳ, sau đó tính toán một giá trị trung bình di động đơn giản 10 chu kỳ của giá trị trung bình di động này.

Hàm có điều kiện IIF()

Hàm iif() được sử dụng để tạo phép gán có điều kiện. Nó chứa ba tham số như được trình bày trong ví dụ sau đây.

dynamicrsi = IIf(Close > MA(C,10), RSI(9), RSI(14));

Câu lệnh "iif" trên đọc (bằng tiếng Anh) như sau: Nếu giá đóng cửa hôm nay lớn hơn giá trị trung bình di động đơn giản 10 ngày của giá đóng cửa, thì gán một giá trị RSI 9 ngày cho biến dynamicrsi, ngược lại, gán một giá trị RSI 14 ngày. Công thức tiếp theo gán khối lượng dương cho biến volresult nếu giá đóng cửa lớn hơn giá trung, nếu không, "khối lượng âm" sẽ được gán.

volresult = IIf(Close > (High+Low)/2, Volume, -Volume);

Nếu bạn chỉ muốn một biểu thức được đánh giá là đúng hoặc sai, nó có thể được thực hiện mà không cần sử dụng hàm iif(). Công thức sau sẽ cho kết quả là 1 (đúng) hoặc 0 (sai):
result = RSI(14) > 70;

Cũng có thể làm tương tự với iif() nhưng công thức sẽ dài hơn.

result = IIf(RSI(14) > 70, 1, 0);

Lưu ý rằng IIF là một hàm - vì vậy kết quả của việc đánh giá được trả lại bởi hàm đó và nên được gán cho một biến nào đó. Ilf luôn luôn đánh giá cả TRUE_PART và FALSE_PART, mặc dù nó chỉ trả lại một trong hai. Vì vậy, bạn nên chú ý đến những tác động phụ không mong muốn. Hàm IIF KHÔNG phải là một câu lệnh kiểm soát luồng. Nếu bạn cần kiểm soát luồng (thực hiện có điều kiện một số phần mã), bạn nên tìm hiểu về câu lệnh điều kiện if-else được mô tả sau trong tài liệu này. Ví dụ sau đây cho thấy một lỗi phổ biến thường gặp khi sử dụng hàm IIF:
IIf(condition, result = 7, result = 9); // ĐÂY LÀ SAI

Cách sử dụng chính xác là:
result = IIf(condition, 7, 9);
/ 7 hoặc 9 sẽ được trả về và gán cho biến result tùy thuộc vào điều kiện */*

Biến

Để rút ngắn, đơn giản hóa, cải tiến và dễ bảo dưỡng các công thức phức tạp, bạn có thể muốn sử dụng biến. Trên thực tế, sử dụng biến có thể cải thiện đáng kể tốc độ tính toán công thức. Vì vậy, khuyến khích sử dụng biến và không có giới hạn về số lượng biến bạn có thể xác định.

Một biến là một định danh được gán cho một biểu thức hoặc một hằng số. Số lượng biến được sử dụng trong một công thức không bị giới hạn. Biến phải được gán trước khi được sử dụng trong công thức. Không thể gán biến trong một cuộc gọi hàm. Tên biến do người dùng định nghĩa (định danh) không được trùng lặp với các tên đã được sử dụng bởi các hàm (ví dụ: ma, rsi, cci, iif, v.v.) hoặc các định danh mảng được định nghĩa trước (ví dụ: open, high, low, close, simple, o, c, l, h, s, a).

Biến dự trữ

AmiBroker sử dụng một số tên biến dự trữ trong các công thức của nó, ví dụ: trong cửa sổ Auto-Analysis, bạn phải gán giá trị cho 2 biến có tên 'buy' hoặc 'sell' để chỉ định các điều kiện mà "buy" và "sell" xảy ra. Ví dụ (hệ thống mua khi MACD tăng lên trên 0, và bán khi MACD giảm xuống dưới 0)

Buy = Cross(MACD(), 0);
Sell = Cross(0, MACD());

AmiBroker sử dụng các tên biến dự trữ sau. Xin lưu ý rằng các biến được đánh dấu là lỗi thời KHÔNG nên được sử dụng trong việc lập trình mới. Chúng chỉ được giữ lại vì tính tương thích ngược và các công thức mới nên sử dụng các hàm hiện đại như Plot() để vẽ các chỉ số và AddColumn() để định nghĩa các cột khám phá.

Biến	Sử dụng	Áp dụng cho
buy	định nghĩa quy tắc Mua (Vào vị thế Long)	Phân tích tự động, Bình luận
sell	định nghĩa quy tắc Bán (đóng vị thế Long)	Phân tích tự động, Bình luận
short	định nghĩa quy tắc Bán khống short (Vào vị thế Short)	Phân tích tự động
cover	định nghĩa quy tắc cover (đóng vị thế Short)	Phân tích tự động
buyprice	định nghĩa mảng giá mua (mảng này được điền bằng các giá trị mặc định theo cài đặt của Automatic Analyser)	Phân tích tự động
sellprice	định nghĩa mảng giá bán (mảng này được điền bằng các giá trị mặc định theo cài đặt của Automatic Analyser)	Phân tích tự động
shortprice	định nghĩa mảng giá bán short (mảng này được điền bằng các giá trị mặc định theo cài đặt của Automatic Analyser)	Phân tích tự động
coverprice		
title		
tooltip		
graphxspace		
graphzorder	(last to first) được sử dụng khi GraphZOrder là 1 (đúng)	

	- thứ tự đảo ngược sẽ được áp dụng.	

exclude	Nếu được định nghĩa, giá trị đúng (hoặc 1) của biến này loại trừ biểu tượng hiện tại khỏi quét/khám phá/kiểm tra lại. Chúng cũng không được xem xét trong các phép tính mua và giữ. Hữu ích khi bạn muốn thu hẹp phân tích của mình với một số tập biểu tượng nhất định.	Phân tích tự động
roundlotsize	định nghĩa các kích cỡ mẫu tròn được sử dụng bởi backtester (xem giảng giải phía dưới)	Phân tích tự động (mới trong 4.10)
ticksize	định nghĩa kích thước tick được dùng để căn giá do built-in stops gây ra (xem giảng giải phía dưới) (lưu ý: nó không ảnh hưởng đến giá buyprice/sellprice/shortprice/coverprice)	Phân tích tự động (mới trong 4.10)
pointvalue	cho phép đọc và sửa đổi giá trị điểm hợp đồng tương lai (xem backtesting futures). CHÚ Ý: biến AFL này mặc định được đặt là 1 (một) bất kể nội dung của cửa sổ Thông tin TRỪ KHI bạn bật chế độ futures (SetOption("FuturesMode", True))	Phân tích tự động (mới trong 4.10)
margindeposit	cho phép đọc và sửa đổi lề hợp đồng tương lai (xem backtesting futures)	Phân tích tự động (mới trong 4.10)
positionsize	Cho phép kiểm soát số tiền hoặc phần trăm của danh mục đầu tư được đầu tư vào giao dịch (thông tin thêm có sẵn trong "Hướng dẫn: Kiểm tra lại ý tưởng giao dịch của bạn")	Phân tích tự động (mới trong 3.9)
positionscore	Định rõ điểm của vị trí. Thông tin thêm: "Hướng dẫn: Testing danh mục đầu tư"	Phân tích tự động
numcolumns	Chỉ khám phá: định nghĩa số lượng cột của riêng bạn (loại trừ các cột định sẵn như ticker và date) và gán giá trị cột cho biến	Phân tích tự động
filter	Chỉ khám phá: kiểm soát các kí tự/báo giá nào được chấp nhận. Nếu "true" (hoặc 1) được gán cho biến này cho biểu tượng/báo giá cho, biểu tượng/báo giá đó sẽ được hiển thị trong báo cáo.	Phân tích tự động
columnN (lỗi thời)		
columnNformat (lỗi thời)	Sẽ hiển thị giá đóng cửa với 4 số thập phân. (Lưu ý cho người dùng nâng cao: phần nguyên của số này có thể được sử dụng để thêm dấu cách trước số được	

	định dạng - 6.0 sẽ không có số thập phân nhưng một số được thêm dấu cách cho đến 6 kí tự.)	

columnNname (lỗi thời)	Chỉ dành cho Exploration: cho phép bạn định nghĩa tên tiêu đề. Gán column0name = "Close"; Sẽ thay đổi tên của cột tùy chỉnh đầu tiên từ tên mặc định "Column 0" thành "Close" phù hợp hơn.	
maxgraph	chỉ ra số lượng tối đa các đồ thị được vẽ trong chỉ số tùy chỉnh	Chỉ báo
graphN (lỗi thời)	xác định công thức cho đồ thị số N (ví dụ N là một số	Chỉ báo
graphNname (lỗi thời)		Chỉ báo
graphNcolor (lỗi thời)	xác định chỉ số màu của đường đồ thị thứ N (chỉ số màu liên quan đến bảng màu hiện tại - xem Preferences/Color) 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	Chỉ báo

	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	

graphNbarcolor (lỗi thời)	(định nghĩa mảng chứa chỉ số màu cho mỗi thanh được vẽ)	Chỉ báo
graphNstyle (lỗi thời)	(định nghĩa phong cách của biểu đồ Nth. Phong cách được định nghĩa dưới dạng tổ hợp (tổng) của một hoặc nhiều cờ sau: styleLine = 1 - biểu đồ line (mặc định) styleHistogram = 2 - biểu đồ histogram styleThick =4 - line dày styleDots = 8 - bao gồm các dots styleNoLine = 16 - không có line styleLog = 32 - tỷ lệ log phân nửa styleCandle = 64 - biểu đồ candlestick styleBar = 128 - biểu đồ bar truyền thống styleNoDraw = 256 - không vẽ (chỉ scale trục) styleStaircase= 512 - biểu đồ vuông (square chart) styleSwingDots = 1024 - dấu chấm ở giữa cho biểu đồ square styleNoRescale = 2048 - không scale lại styleNoLabel = 4096 - không có nhãn giá trị stylePointAndFigure = 8192 - dạng biểu đồ điểm và hình (mới trong 4.20) styleArea = 16384 - biểu đồ khu vực (histogram rộng đặc biệt) styleOwnScale = 32768 - biểu đồ sử dụng quy mô độc	Chỉ báo

	lập styleLeftAxisScale = 65536 - biểu đồ sử dụng quy mô trục trái (độc lập với trục phải) Không phải tất cả các tổ hợp cờ đều hợp lí, ví dụ (64+1) (candlestick + line) sẽ tạo thành biểu đồ candlestick (style=64) Ghi chú về biểu đồ candlestick/bar: những phong cách này sử dụng các mảng O, H, L ngoài graphN.	

graphNbarcolor (lỗi thời)	Định nghĩa mảng của chỉ số màu cho thanh và nến trong Nth graph ine (chỉ số màu liên quan đến bảng màu hiện tại - xem Preferences/Color)	Chỉ báo

XEM THÊM:

- Từ khóa
- THỦ TỤC DO NGƯỜI DÙNG ĐỊNH RÕ, PHẠM VI CỤC BỘ/GLOBAL

Từ khóa

Dưới đây là các từ khóa trong ngôn ngữ AmiBroker Formula Language:

Vòng lặp Loops:

- do (một phần của câu lệnh do-while)
- while
- for

Điều kiện thực thi / Điều khiển dòng chảy:

- if (một phần của câu lệnh if-else)
- else (một phần của câu lệnh if-else)
- switch
- break (một phần của câu lệnh switch hoặc câu lệnh for/while)
- case (một phần của câu lệnh switch)
- continue (một phần của câu lệnh for/while)
- default (một phần của câu lệnh switch)

Hàm số:

- function
- procedure
- return
- local (phạm vi biến)
- global (phạm vi biến)

Loại biến:

- static
- typeof

Hãy đảm bảo rằng các thuật ngữ tiếng Anh kỹ thuật vẫn giữ nguyên không thay đổi. Đồng thời, duy trì cấu trúc và thông tin gốc trong khi cải thiện khả năng đọc và định dạng.

Từ khóa Break

Từ khóa break là một phần của câu lệnh switch và là một phần tùy chọn của các câu lệnh lặp for, do-while và while. Từ khóa break kết thúc câu lệnh do, for, switch, hoặc while nhỏ nhất bao quanh nơi nó xuất hiện.

Break;

Câu lệnh break được sử dụng để thoát khỏi một vòng lặp hoặc câu lệnh switch. Nó chuyển quyền kiểm soát đến câu lệnh ngay sau vòng lặp con hoặc câu lệnh switch. Câu lệnh break chỉ kết thúc vòng lặp hoặc câu lệnh switch bao quanh nó một cách sát sao nhất. Trong vòng lặp, break được sử dụng để kết thúc trước khi tiêu chí kết thúc được đánh giá là 0. Trong câu lệnh switch, break được sử dụng để kết thúc các phần code - thông thường trước một nhãn trường hợp. Ví dụ sau minh họa việc sử dụng câu lệnh break trong một vòng lặp for:

```
i = 0;
while (i < 10)
{
   i++;
   // break at step 5
   if( i == 5 )
   {
      break;
    }
   printf("Step " + i );
}
```

Để xem ví dụ về việc sử dụng câu lệnh break trong phần thân của một câu lệnh switch, hãy xem phần Câu Lệnh Switch.

Từ khóa Case

Từ khóa case là một phần không thể thiếu của câu lệnh switch-case.

Xin lưu ý: Đã giữ nguyên những thuật ngữ tiếng Anh kỹ thuật để không làm thay đổi nghĩa của nội dung. Đồng thời, cải thiện định dạng và khả năng đọc để nâng cao trải nghiệm người đọc.

Từ khóa Continue

Từ khóa continue là một phần tùy chọn của câu lệnh for, do-while và while. Nó dừng vòng lặp hiện tại và bắt đầu một vòng lặp mới.

Continue;

Bạn chỉ có thể sử dụng mệnh đề continue bên trong một vòng lặp while, do...while, hoặc for. Thực thi mệnh đề continue dừng vòng lặp hiện tại và tiếp tục chương trình tại điểm bắt đầu của vòng lặp. Điều này có những tác động sau đối với các loại vòng lặp khác nhau: vòng lặp while và do...while kiểm tra điều kiện của chúng, và nếu true, thực thi vòng lặp một lần nữa. Vòng lặp for thực thi biểu thức tăng của nó, và nếu biểu thức kiểm tra là true, thực thi vòng lặp một lần nữa. Ví dụ sau minh họa cách sử dụng mệnh đề continue:

```
i = 0;
while ( i < 10 )
{
   i++;
   // Bỏ qua 5
   if( i == 5 )
   {
      continue;
   }
   printf("Step " + i );
}
```

Từ khoá Default

Từ khoá default là một phần không thể thiếu của câu lệnh switch-case.

Vui lòng cải tiến và cải thiện nội dung Markdown này. Đảm bảo rằng các thuật ngữ tiếng Anh kỹ thuật vẫn giữ nguyên. Hãy duy trì cấu trúc và thông tin gốc trong khi nâng cao khả năng đọc và định dạng. Làm cho nó trở nên gọn gàng, dễ nhìn hơn và dịch phần còn lại sang tiếng Việt. Kết quả xuất ra ở dạng Markdown tiếng Việt, giữ nguyên các thuật ngữ tiếng Anh chuyên môn và cải thiện khả năng đọc và định dạng.

Từ khóa Do

Từ khóa 'do' là một phần của câu lệnh do-while.

Câu lệnh Do-While

Câu lệnh do-while cho phép bạn lặp lại một câu lệnh hoặc một câu lệnh phức hợp cho đến khi một biểu thức được chỉ định trở thành sai.

Cú pháp

do statement while (expression) ;

Biểu thức trong câu lệnh do-while được đánh giá sau khi thân vòng lặp được thực hiện. Do đó, thân vòng lặp luôn được thực hiện ít nhất một lần.

Biểu thức phải có kiểu số học hoặc boolean. Quá trình thực thi diễn ra như sau:

1. Thân câu lệnh được thực hiện.
2. Sau đó, biểu thức được đánh giá. Nếu biểu thức sai, câu lệnh do-while kết thúc và quyền kiểm soát chuyển sang câu lệnh tiếp theo trong chương trình. Nếu biểu thức đúng (không phải là không), quá trình lặp lại, bắt đầu với bước 1.

Dưới đây là một ví dụ về câu lệnh do-while:

```
x=100;
do
{
        y = sin( x );
        x--;
}
while ( x > 0 );
```

Trong câu lệnh do-while này, hai câu lệnh y = sin(x); và x--; được thực hiện, bất kể giá trị ban đầu của x. Sau đó, x > 0 được đánh giá. Nếu x lớn hơn 0, thì thân câu lệnh được thực hiện lại và x > 0 được đánh giá lại. Thân câu lệnh được thực hiện liên tục miễn là x vẫn lớn hơn 0. Thực thi của câu lệnh do-while kết thúc khi x trở thành 0 hoặc số âm. Thân vòng lặp được thực hiện ít nhất một lần.

Từ khoá Else

Từ khoá else là một phần tùy chọn của câu lệnh if-else.

Câu lệnh If, Else

if(biểu_thức)

câu_lệnh1

[else câu_lệnh2]

Từ khoá if sẽ thực thi câu_lệnh1 nếu biểu_thức là đúng (khác không); nếu else được sử dụng và giá trị của biểu_thức là sai (bằng không), nó sẽ thực thi câu_lệnh2. Sau khi thực thi câu_lệnh1 hoặc câu_lệnh2, control sẽ chuyển sang câu lệnh tiếp theo.

Ví dụ 1

```
if ( i > 0 )
   y = x / i;
else
{
   x = i;
   y = abs( x );
}
```

Trong ví dụ này, câu lệnh y = x/i; được thực thi nếu i lớn hơn 0. Nếu i nhỏ hơn hoặc bằng 0, giá trị của i sẽ được gán cho x và giá trị tuyệt đối của x (abs(x)) sẽ được gán cho y. Chú ý rằng câu lệnh tạo thành mệnh đề if kết thúc bằng dấu chấm phẩy. Khi lồng các câu lệnh if và các mệnh đề else, hãy sử dụng dấu ngoặc nhọn để nhóm các câu lệnh và các mệnh đề thành các câu lệnh phức để rõ ràng ý định của bạn. Nếu không có dấu ngoặc nhọn, trình biên dịch sẽ giải quyết những điểm không rõ bằng cách liên kết mỗi else với câu lệnh if gần nhất mà không có else.

Ví dụ 2

```
if ( i > 0 )          /* Không có dấu ngoặc nhọn */
if ( j > i )
       x = j;
else
       x = i;
```

Mệnh đề else trong ví dụ này được liên kết với câu lệnh if bên trong. Nếu i nhỏ hơn hoặc bằng 0, không có giá trị nào được gán cho x.

Ví dụ 3

```
if ( i > 0 )       /* Có dấu ngoặc nhọn */
{
         if ( j > i )
          x = j;
}
else x = i;
```

Hy vọng nội dung này giúp bạn dễ hiểu hơn về việc sử dụng từ khoá else trong lập trình.
Dấu ngoặc nhọn xung quanh câu lệnh if bên trong trong ví dụ này làm cho phần else trở thành một phần của câu lệnh if bên ngoài. Nếu giá trị của i nhỏ hơn hoặc bằng 0, i sẽ được gán cho x.

Những Hiểu Lầm Phổ Biến

Câu hỏi về "Vấn đề mới về if-else":
Tại sao tôi gặp lỗi cú pháp khi viết: if(H > Ref(H,-1))?

Câu trả lời:
Câu lệnh if-else thay đổi dòng chảy của quá trình thực thi (đối lập với hàm IIF, hàm này đánh giá tất cả các đối số và hoạt động trên mảng) và bạn không thể viết nếu "if (H >Ref(H,-1))" vì nó không có ý nghĩa. Nó sẽ được dịch là "Nếu mảng cao hơn mảng cao được dịch một thanh" (xem hướng dẫn dưới đây). Câu lệnh điều khiển dòng chảy (như if-else) phải nhận GIÁ TRỊ boolean ĐƠN để quyết định đường dẫn thực thi nào nên được chọn. Nếu bạn ghi H (hoặc High) nghĩa là ARRAY (toàn bộ mảng). Nếu bạn viết H[i] - nghĩa là phần tử thứ i của mảng. Toán tử dấu ngoặc vuông [ ] cho phép bạn truy cập vào các phần tử mảng riêng lẻ.

Thay vào đó, bạn nên viết:

```
for( i = 1; i < BarCount; i++ )
{
   if ( High[ i ] > High[ i - 1 ] )
   {
      x[ i ] = High[ i ];
   }
   else
   {
      x[ i ] = Low[ i ];
   }
}
```

Điều nó sẽ dịch là "cho MỖI THANH 'i' gán phần tử thứ i của mảng cao cho phần tử thứ i của mảng x nếu phần tử thứ i của mảng cao cao hơn phần tử trước đó, nếu không thì gán khẩu độ thấp của mảng thứ i vào phần tử mảng thứ i của x". Quy tắc là: các câu lệnh mới như if-else và while cần một giá trị boolean đơn (không phải mảng) để quyết định đường dẫn thực thi nào nên được chọn. Nếu bạn muốn sử dụng chúng với các mảng, bạn phải lặp qua các bar bằng vòng lặp for (như đã hiển thị ở trên).
Mặt khác, điều này có thể đươc thực hiện trong một dòng dùng các thao tác trên mảng cũ và hàm IIF:

```
x = IIf( High > Ref( High, -1 ), High, Low );
```

Điều này hoạt động vì IIF hoạt động trên các MẢNG như mô tả trong hướng dẫn. Như bạn có thể thấy trong nhiều trường hợp, AFL cũ cung cấp một dạng hình thức ngắn gọn hơn nhiều. Tôi luôn cố gắng giải thích điều này rằng có rất nhiều lợi thế khi sử dụng AFL nhưng chỉ một số người nhận ra điều đó. Các lệnh điều khiển mới nên được sử dụng nơi mà chúng có thể được sử dụng một cách tốt nhất. Như tôi đã cố gắng giải thích trong những năm qua, trong 80% trường hợp, AFL kiểu 'cũ' cung cấp công thức ngắn nhất. Chỉ có 20% số trường hợp còn lại cần đến script. Những trường hợp 'chỉ sử dụng script' giờ đây có thể được mã hóa bằng AFL native nhờ vào các lệnh for/while/if-else mới. Và đây là cách sử dụng chính xác của chúng - thay thế phần script.

for Statement

for (init-expression ; cond-expression ; loop-expression) lệnh

Từ khoá Function

Từ khoá function bắt đầu định nghĩa một "user function". Các hàm do người dùng định nghĩa cho phép đóng gói mã của người dùng thành các mô-đun dễ sử dụng, có thể được sử dụng ở nhiều nơi mà không cần sao chép cùng một đoạn code đi đi lại. Hàm function phải có một định nghĩa. Định nghĩa hàm bao gồm phần thân của hàm - đoạn code mà sẽ được thực thi khi hàm được gọi. Định nghĩa hàm xác định tên, và các thuộc tính (hoặc tham số) của hàm. Định nghĩa hàm phải xuất hiện trước lời gọi hàm. Định nghĩa bắt đầu bằng từ khoá function, tiếp theo là tên hàm, dấu ngoặc mở, danh sách tham số tùy chọn và dấu ngoặc đóng. Sau đó là phần thân của hàm được bao trong dấu ngoặc nhọn. Khi gọi hàm, quyền kiểm soát chương trình sẽ được chuyển từ hàm gọi tới hàm được gọi. Các tham số, nếu có, sẽ được truyền theo giá trị tới hàm được gọi. Việc thực thi lệnh trả về trong hàm được gọi sẽ chuyển quyền kiểm soát và có thể là một giá trị tới hàm gọi. Nếu hàm không bao gồm bất kỳ lệnh trả về nào (không trả về bất kỳ gì) thì chúng ta gọi đó là một thủ tục. Dưới đây là một ví dụ về việc định nghĩa hàm:

```
function IIR2( input, f0, f1, f2 )
{
   result[ 0 ] = input[ 0 ];
   result[ 1 ] = input[ 1 ];

for( i = 2; i < BarCount; i++ )
   {
      result[ i ] = f0 * input[ i ] +
                    f1 * result[ i - 1 ] +
                    f2 * result[ i - 2 ];
   }

return result;
}

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

```
Plot( IIR2( Close, 0.2, 1.4, -0.6 ), "function example", colorRed );
```

Trong đoạn code này, IIR2 là một hàm được định nghĩa bởi người dùng. input, f0, f1, f2 là các tham số form của hàm. Khi gọi hàm, giá trị của các đối số được truyền vào các biến này. Các tham số form hoạt động như các biến cục bộ. Sau đó, chúng ta có result và i, đây là các biến cục bộ. Các biến cục bộ chỉ hiển thị trong hàm. Nếu một hàm khác sử dụng cùng tên biến, chúng sẽ không gây ra sự xung đột lẫn nhau.

Từ khoá Global

Từ khóa global khai báo biến toàn cục bên trong hàm do người dùng định nghĩa. Biến toàn cục là biến có thể nhìn thấy/truy cập được bên trong hàm VÀ bên ngoài hàm (ở cấp công thức toàn cục).

Do thực tế là AFL theo mặc định không yêu cầu khai báo biến nên quyết định bất cứ khi nào biến đã cho được coi là cục bộ hay toàn cục đều phụ thuộc vào nơi nó được SỬ DỤNG ĐẦU TIÊN.

Tuy nhiên, bạn có thể buộc công cụ AFL yêu cầu tất cả các biến phải được khai báo bằng từ khóa cục bộ hoặc toàn cục trên cơ sở từng công thức bằng cách đặt SetOption("RequireDeclarations", True); ở đầu công thức.

Nếu định danh đã cho xuất hiện đầu tiên BÊN TRONG định nghĩa hàm - thì nó được coi là biến CỤC BỘ.

Nếu định danh đã cho xuất hiện đầu tiên BÊN NGOÀI định nghĩa hàm - thì nó được coi là biến TOÀN CỤC.

Tuy nhiên, hành vi mặc định này có thể được ghi đè bằng từ khóa toàn cục và cục bộ (được giới thiệu trong 4.36) – xem ví dụ 2.

Ví dụ (bình luận):

```
k = 4; // đây là biến TOÀN CỤC
function f( x )
{
  z = 3; // Đây là biến CỤC BỘ
  return z * x * k; // 'k' ở đây tham chiếu đến biến toàn cục k (được sử dụng lần đầu bên ngoài hàm)
}
z = 5; // đây là biến TOÀN CỤC có cùng tên với biến cục bộ trong hàm f

"The value of z before function call :" + WriteVal( z );

// Now even if we call function
// the value of our global variable z
// is not affected by function call because
// global variable z and local variable z are separate and
// arguments are passed by value (not by reference)
```

```
"The result of f( z ) = " + WriteVal( f( z ) );


"The value of z after function call is unchanged : " + WriteVal( z );
```

Ví dụ 2: Sử dụng biến cục bộ và toàn cục để ghi đè các quy tắc hiển thị mặc định

```
VariableA = 5; // biến toàn cục ngầm định

function Test()
{
   local VariableA;     // biến cục bộ rõ ràng có cùng định danh với biến toàn cục
   global VariableB;  // biến toàn cục rõ ràng chưa được định nghĩa trước đó
                      // có thể được sử dụng để trả về nhiều hơn một giá trị từ hàm

   VariableA = 99;
   VariableB = 333;
}

VariableB = 1;         // Biến toàn cục

"Trước khi gọi hàm";
"VariableA = " + VariableA;
"VariableB = " + VariableB;

Test();

"Sau khi gọi hàm";
"VariableA = " + VariableA + " (Không bị ảnh hưởng bởi việc gọi hàm )";
"VariableB = " + VariableB + " (Bị ảnh hưởng bởi việc gọi hàm)"
```

Từ khóa If

Từ khóa if là một phần không thể thiếu của câu lệnh if-else.

Câu lệnh If, Else

if(expression) statement1 [**else** statement2]. Từ khóa if sẽ thực hiện statement1 nếu expression là đúng (khác không); nếu có từ khóa else và expression là sai (bằng không), nó sẽ thực hiện statement2. Sau khi thực hiện statement1 hoặc statement2, quyền kiểm soát sẽ chuyển sang câu lệnh tiếp theo.

Ví dụ 1

```
if (i > 0)
y = x / i;
else { x = i; y = abs(x); }.
```

Trong ví dụ này, câu lệnh y = x/i; sẽ được thực hiện nếu i lớn hơn 0. Nếu i nhỏ hơn hoặc bằng 0, i sẽ được gán cho x và abs(x) sẽ được gán cho y. Lưu ý rằng câu lệnh tạo thành điều kiện if kết thúc bằng dấu chấm phẩy. Khi lồng những câu lệnh if và else, hãy sử dụng dấu ngoặc nhọn để nhóm các câu lệnh và các điều kiện thành các câu lệnh hợp chất, làm rõ ý định của bạn. Nếu không có dấu ngoặc nhọn, trình biên dịch sẽ giải quyết sự không rõ ràng bằng cách kết hợp mỗi else với if gần nhất mà không có else.

Ví dụ 2

```
if (i > 0) /* Không có dấu ngoặc nhọn */
if (j > i)
x = j; else x = i;
```

Else trong ví dụ này được kết hợp với câu lệnh if ở trong cùng. Nếu i nhỏ hơn hoặc bằng 0, không có giá trị nào được gán cho x.

Ví dụ 3

```
if (i > 0) { /* Có dấu ngoặc nhọn */
if (j > i)
x = j; }
else x = i;
```

Dấu ngoặc nhọn xung quanh câu lệnh if bên trong trong ví dụ này khiến cho câu lệnh else trở thành một phần của câu lệnh if bên ngoài. Nếu i nhỏ hơn hoặc bằng 0, i sẽ được gán cho x.

Những Hiểu Lầm Phổ Biến

Câu hỏi về "Vấn đề mới với câu lệnh if-else": Tại sao tôi lại gặp lỗi cú pháp khi tôi viết: if(H > Ref(H,-1))? Câu trả lời: câu lệnh if-else thay đổi dòng chảy thực thi (đối lập với hàm IIF, hàm này đánh giá tất cả các đối số và hoạt động trên các mảng) và thực sự bạn không thể viết if (H >Ref(H,-1))
bởi nó không có ý nghĩa. Nó sẽ được dịch thành "Nếu mảng cao hơn mảng cao được dịch chuyển một thanh" (xem hướng dẫn dưới đây). Câu lệnh điều khiển dòng chảy (như if-else) phải nhận ĐƠN lẻ giá trị boolean để quyết định đường thực thi nào nên được chọn. Nếu bạn viết H (hoặc High) nó có nghĩa là MẢNG (toàn bộ mảng). nếu bạn viết H[i] - nó có nghĩa là phần tử i-th của mảng. Toán tử chỉ mục [ ] giúp bạn truy cập vào các phần tử mảng riêng lẻ. Thay vào đó, bạn nên viết:

```
for( i = 1; i < **BarCount**; i++ )
{
  if ( **High**[ i ] > **High**[ i - 1 ] )
  {
    x[ i ] = **High**[ i ];
  }
  else
  {
    x[ i ] = Low[ i ];
  }
}
```

Câu này sẽ được dịch đúng thành "với MỌI THANH 'i' gán phần tử i-th của mảng cao cho phần tử i-th của mảng x nếu phần tử i-th của mảng cao cao hơn phần tử trước đó, nếu không thì gán phần tử i-th của mảng thấp cho phần tử i-th của mảng x". Quy tắc là: câu lệnh if-else mới và câu lệnh while cần một giá trị boolean đơn lẻ (không phải mảng) để quyết định đường thực thi nào nên được chọn. Nếu bạn muốn sử dụng chúng với mảng, bạn phải lặp qua các thanh bằng vòng lặp for (như đã được chứng minh ở trên). Ngược lại, điều này có thể được thực hiện trong một dòng duy nhất bằng cách sử dụng các thao tác mảng kiểu cũ và hàm IIF:

```
x = IIf( **High** > Ref( **High**, -1 ), **High**, **Low** );
```

Lý do là hàm IIF hoạt động trên CÁC MẢNG như được mô tả trong hướng dẫn. Như bạn có thể thấy trong nhiều trường hợp, AFL kiểu cũ mang lại hình dạng gọn gàng hơn rất nhiều. Tôi luôn cố gắng giải thích điều này với các ưu điểm khi sử dụng AFL, nhưng chỉ có một số ít nhận ra điều đó. Các câu lệnh kiểm soát mới nên được sử dụng nơi mà chúng thực sự hữu ích. Như tôi đã cố gắng giải thích trong những năm qua, trong 80% các trường hợp, 'old-style' AFL cung cấp công thức ngắn nhất. Chỉ còn lại 20% các trường hợp cần đến script. Những trường hợp 'script-only' giờ đây có thể được lập trình bằng AFL gốc nhờ vào các lệnh mới như for/while/if-else. Đây chính là cách sử dụng đúng nhất của chúng - để thay thế các phần script.

Từ khóa Local

Từ khóa local dùng để khai báo biến cục bộ trong hàm do người dùng định nghĩa. Biến cục bộ chỉ có thể nhìn thấy/truy cập được bên trong hàm. Do AFL mặc định không yêu cầu phải khai báo biến, quyết định xem một biến cụ thể sẽ được xử lý như biến toàn cục hay biến cục bộ phụ thuộc vào nơi nó được SỬ DỤNG ĐẦU TIÊN. Tuy nhiên, bạn có thể yêu cầu AFL engine phải khai báo tất cả các biến bằng cách sử dụng từ khóa local hoặc global trên từng công thức bằng cách đặt SetOption("RequireDeclarations", True); ở đầu công thức. Nếu một định danh xuất hiện ĐẦU TIÊN BÊN TRONG định nghĩa hàm - thì nó được xem là biến CỤC BỘ. Nếu một định danh xuất hiện ĐẦU TIÊN BÊN NGOÀI định nghĩa hàm - thì nó được xem là biến TOÀN CỤC. Tuy nhiên, hành vi mặc định này có thể được ghi đè bằng cách sử dụng các từ khóa global và local (được giới thiệu trong 4.36) - xem ví dụ 2.
Ví dụ (lời bình):

```
k = 4; // đây là biến TOÀN CỤC
function f( x )
{
z = 3; // đây là biến CỤC BỘ
return z * x * k; // 'k' ở đây tham chiếu đến biến toàn cục k (được sử dụng lần đầu bên ngoài hàm)
}
z = 5; // đây là biến TOÀN CỤC có cùng tên với biến cục bộ trong hàm f
"Gía trị của z trước khi gọi hàm:" + WriteVal( z );
// Bây giờ ngay cả khi chúng ta gọi hàm
// giá trị của biến toàn cục z
// không bị ảnh hưởng bởi việc gọi hàm bởi vì
// biến toàn cục z và biến cục bộ z là riêng biệt và
// các đối số được truyền qua giá trị (không qua tham chiếu)
"Kết quả của f( z ) = " + WriteVal( f( z ) );
"Gía trị của z sau khi gọi hàm không thay đổi: " + WriteVal( z );
```

Ví dụ 2: Sử dụng từ khóa local và global để ghi đè quy tắc hiển thị mặc định:

```
VariableA = 5; // biến toàn cục ngầm định
function Test()
{
 local VariableA; // biến cục bộ rõ ràng có cùng định danh với biến toàn cục
```

```
global VariableB; // biến toàn cục rõ ràng chưa được định nghĩa trước đó
// có thể được sử dụng để trả về nhiều hơn một giá trị từ hàm
VariableA = 99;
VariableB = 333;
}
VariableB = 1; // biến toàn cục
"Trước khi gọi hàm";
"VariableA = " + VariableA;
"VariableB = " + VariableB;
Test();
"Sau khi gọi hàm";
"VariableA = " + VariableA + " (không bị ảnh hưởng bởi lệnh gọi hàm)";
"VariableB = " + VariableB + " (bị ảnh hưởng bởi lệnh gọi hàm)"
```

Từ khóa Procedure

Từ khóa procedure bắt đầu định nghĩa của user-procedure. Procedure là một hàm không trả về bất kỳ giá trị nào (không có câu lệnh trả về). Xin tham khảo từ khóa function để biết thêm chi tiết.

Từ khoá Return

Từ khoá return cho phép trả giá trị từ hàm.

```
function RiseToAPowerOf2( x )
{
 return x ^ 2;
}
```

Ở cuối hàm, ta có thể thấy câu lệnh 'return' được sử dụng để trả kết quả cho người gọi. Lưu ý rằng hiện nay câu lệnh return phải được đặt ở cuối cùng của hàm. Hãy tham khảo thêm từ khóa 'function' để biết thêm chi tiết.

Từ khóa static cho phép khai báo cú pháp:

static identifier;
static identifier1, identifier2, identifier3, ...;

Từ khóa static khai báo biến là static. Các biến được khai báo static có thể và nên được sử dụng như các biến 'bình thường', không cần gọi chức năng (Static*).

QUAN TRỌNG:
Từ khóa **static** chỉ có sẵn khi bạn sử dụng
#pragma **enable_static_decl**(prefix)

ở đầu công thức của bạn. Tiền tố sẽ được thêm tự động vào tên biến static thực tế để tránh xung đột tên. Tiền tố nên là chuỗi kí tự hằng số (như trong Ví dụ 1).

Bạn cũng có thể có biến static đã khai báo với tiền tố duy nhất bao gồm chartID, khoảng thời gian hoặc biểu tượng hiện tại. Để làm vậy, hãy sử dụng các token {chartid}, {interval} hoặc {symbol} trong chuỗi tiền tố. Chúng sẽ được thay thế trong thời gian runtime bởi các giá trị tương ứng (xem Ví dụ 2).

Ví dụ 1:

```
// yêu cầu pragma bên dưới để hỗ trợ khai báo static
// nó cũng định nghĩa tiền tố được thêm vào biến được khai báo với từ khóa static
// có một tiền tố cho mỗi công thức.
#pragma enable_static_decl(prefix)
// rất là ý tưởng tốt khi sử dụng quy ước đặt tên
// phân biệt biến static đã khai báo từ những biến khác
// chúng tôi khuyên dùng DẤU GẠCH DƯỚI " trước tất cả biến static đã khai báo
// để chúng có thể dễ dàng nhận biết trong mã lệnh
// Lựa chọn khác có thể là tiền tố s
static _myvar;
// biến này cũng có thể truy cập bằng "prefix_myvar"
printf( "Biến này thực sự là static %g", _myvar++ );
// hãy nhớ rằng vì tốc độ biến static chỉ đọc một lần tại thời điểm khai báo
// và chỉ lưu một lần - khi thực hiện công thức hoàn tất
// Cách này giúp biến static đã khai báo cung cấp tốc độ giống như các biến thông thường (không hình phạt tốc độ)
```

Ví dụ 2:

```
// Ví dụ 2 - Nếu bạn muốn khai báo các biến tĩnh theo chartID
#pragma enable_static_decl "myprefx{chartid}" static x;
```

// Biến tĩnh này sẽ có tên là myprefixIDx, trong đó ID là chartID hiện tại

Thể hiện rõ hơn về khóa từ static được bao gồm trong diễn đàn:
https://forum.amibroker.com/t/amibroker-6-27-1-beta-released/2925/

switch Statement

Các câu lệnh switch và case giúp kiểm soát các hoạt động có điều kiện và phân nhánh phức tạp. Câu lệnh switch chuyển quyền kiểm soát cho một câu lệnh trong thân của nó.
Syntax:

```
switch ( expression )
{
case constant-expression1 : statement;
case constant-expression2 : statement;
...
case constant-expressionN : statement;
default : statement;
}
```

Quyền điều khiển được chuyển đến câu lệnh có hằng số trường hợp-biểu thức khớp với giá trị của switch (biểu thức).
Câu lệnh switch có thể bao gồm bất kỳ số lượng trường hợp nào, nhưng không có hai hằng số trường hợp nào trong cùng một câu lệnh switch có thể có cùng giá trị. Việc thực thi phần thân câu lệnh bắt đầu tại câu lệnh đã chọn và tiếp tục cho đến khi kết thúc phần thân hoặc cho đến khi câu lệnh break chuyển quyền điều khiển ra khỏi phần thân.

Bạn có thể sử dụng câu lệnh break để kết thúc quá trình xử lý một trường hợp cụ thể trong câu lệnh switch và nhánh đến cuối câu lệnh switch. Nếu không có break, chương trình sẽ tiếp tục đến trường hợp tiếp theo, thực thi các câu lệnh cho đến khi đạt đến break hoặc kết thúc câu lệnh. Trong một số trường hợp, việc tiếp tục này có thể là mong muốn.
Câu lệnh default được thực thi nếu không có case constant-expression nào bằng giá trị của switch (expression). Nếu câu lệnh default bị bỏ qua và không tìm thấy case match nào, thì không có câu lệnh nào trong phần thân switch được thực thi. Chỉ có thể có nhiều nhất một câu lệnh default. Câu lệnh default, nếu tồn tại, PHẢI nằm ở cuối. Nếu không, nó có thể được thực thi trước khi đạt đến các điều kiện được xác định bên dưới nó. Nhãn case hoặc default chỉ được phép xuất hiện bên trong câu lệnh switch.

Kiểu của switch expression và case constant-expression có thể là bất kỳ. Giá trị của mỗi case constant-expression phải là duy nhất trong phần thân câu lệnh. Nếu không, first-match sẽ được sử dụng.

Ví dụ:

```
for( n = 0; n < 10; n++ )
```

```
{
printf("Current n = %f\n", n );

switch(n) {
case 0:
printf("The number is zero.\n");
break;
case 3:
case 5:
case 7:
printf("n is a prime number\n");
break;
case 2: printf("n is a prime number\n");
case 4:
case 6:
case 8:
printf("n is an even number\n");
break;
case 1:
case 9:
printf("n is a perfect square\n");
break;
default:
printf("Only single-digit numbers are allowed\n");
break;
  }
```

Có thể tìm thêm thông tin qua: http://en.wikipedia.org/wiki/Switch_statement

Từ khóa typeof

Từ khóa typeof là một toán tử trả về chuỗi mô tả loại toán hạng.

Cú pháp

Toán tử typeof được sử dụng theo cách sau:
typeof (operand)

Toán tử typeof trả về một chuỗi biểu thị loại toán hạng *chưa được đánh giá*. Toán hạng là chuỗi, biến, định danh hàm hoặc đối tượng mà loại sẽ được trả về. Khi cung cấp định danh, nó nên được cung cấp riêng, không có toán tử số học, không có đối số bổ sung và không có dấu ngoặc nhọn. Nếu bạn muốn kiểm tra loại giá trị được hàm trả về, trước tiên bạn phải gán giá trị trả về cho một biến và sau đó sử dụng:

typeof(variable)

Các giá trị trả về có thể là:

• "undefined" - identifier is not defined
• "number" - operand represents a number (scalar)
• "array" - operand represents an array
• "string" - operand represents a string
• "matrix" - operand represents a matrix
• "function" - operand is a built-in function identifier
• "user function" - operand is a user-defined function
• "object" - operand represents COM object
• "member" - operand represents member function or property of COM object
• "handle" - operand represents Windows handle
"unknown" - type of operand is unknown (should not happen) typeof operator allows among other things to detect undefined variables in the following way

Từ khóa While

Từ khóa while là phần của câu lệnh while (sẽ mô tả dưới đây) và câu lệnh do-while.

Câu lệnh While

Câu lệnh while cho phép bạn lặp lại một câu lệnh cho đến khi một biểu thức được chỉ định trở thành sai.

Cú pháp

Cú pháp để sử dụng câu lệnh while như sau: while (biểu thức) câu lệnh
Biểu thức phải có kiểu số học (số/hỗn hợp). Thực thi sẽ tiến hành như sau:
1. Biểu thức được đánh giá.

2. Nếu biểu thức ban đầu là sai, thân của câu lệnh while sẽ không bao giờ được thực thi, và quyền điều khiển sẽ được chuyển từ câu lệnh while sang câu lệnh tiếp theo trong chương trình.
Nếu biểu thức là đúng (không phải là số không), thân của câu lệnh sẽ được thực thi và quá trình lặp lại bắt đầu từ bước 1.

Đây là một ví dụ về câu lệnh while:

```
i = 10; while( i < 20 ) {
Plot( MA( Close, i ), "MA" + WriteVal( i, 0 ), colorBlack + i );
 i = i + 1; }
```

Ví dụ vẽ ra 10, 11, 12, 13, 14, 15, 16, 17, 18, 19 - bar di chuyển trung bình.

Tham chiếu hàm AFL - Danh sách chức năng theo thứ tự chữ cái

1. #include (Các hàm khác) - lệnh bao gồm bộ đào tạo xử lý (AFL 2.2)
2. #include_once (Các hàm khác) - bao gồm lệnh một lần (AFL 2.70)
3. #pragma (Các hàm khác) - đặt tùy chọn bộ xử lý trước AFL (AFL 2.4)
4. abs (Hàm toán học) - giá trị tuyệt đối
5. AccDist (Chỉ số) - tổng hợp/phân phối
6. acos (Hàm toán học) - hàm arccosine
7. AddColumn (Khám phá / Chỉ số) - thêm cột khám phá số học (AFL 1.8)
8. AddMultiTextColumn (Khám phá / Chỉ số) - thêm cột văn bản khám phá dựa trên mảng (AFL 4.20)
9. AddRankColumn (Khám phá / Chỉ số) - thêm cột(x) xếp hạng theo thứ tự hiện tại được thiết lập bởi SetSortColumns (AFL 5.70)
10. AddRow (Khám phá / Chỉ số) - thêm hàng văn bản thô vào khám phá (AFL 4.0)
11. AddSummaryRows (Khám phá / Chỉ số) - thêm hàng tóm tắt vào kết quả khám phá (AFL 3.2)
12. AddTextColumn (Khám phá / Chỉ số) - thêm cột khám phá văn bản (AFL 1.8)
13. AddToComposite (Tổng hợp) - thêm giá trị vào ticker tổng hợp (AFL 2.0)
14. ADLine (Tổng hợp) - dòng advance/decline (AFL 1.2)
15. AdvIssues (Tổng hợp) - vấn đề đang thăng tiến (AFL 1.2)
16. AdvVolume (Tổng hợp) - khối lượng vấn đề đang thăng tiến (AFL 1.2)
17. ADX (Chỉ số) - chỉ số di chuyển hướng trung bình (AFL 1.3)
18. AlertIf (Bộ công cụ hệ thống giao dịch) - kích hoạt cảnh báo (AFL 2.1)
19. AlmostEqual (Các hàm toán học) - so sánh không nhạy cảm với lỗi làm tròn (AFL 2.80)
20. AMA (Trung bình động, tổng) - trung bình động thích nghi (AFL 1.5)
21. AMA2 (Trung bình động, tổng) - trung bình động thích nghi (AFL 1.5)
22. ApplyStop (Bộ công cụ hệ thống giao dịch) - áp dụng stop được xây dựng sẵn (AFL 1.7)
23. Asc (Manipulation chuỗi) - nhận mã ASCII của ký tự (AFL 2.80)
24. Asin (Các hàm toán học) - hàm arcsine
25. Atan (Các hàm toán học) - arc tan
26. Atan2 (Các hàm toán học) - tính arctangent của y/x (AFL 2.90)
27. ATR (Chỉ số) - phạm vi đúng trung bình (AFL 1.3)
28. BarIndex (Ngày/Giờ) - nhận số bar dựa trên zero (AFL 2.3)
29. BarsSince (Bộ công cụ hệ thống giao dịch) - BarsSince
30. BarsSinceCompare (Bộ công cụ hệ thống giao dịch) - các bar từ khi so sánh giữa các giá trị mảng trong quá khứ và hiện tại là đúng (AFL 4.40)
31. BbandBot (Chỉ số) - dải bollinger dưới
32. BBandTop (Chỉ số) - dải bollinger trên
33. BeginValue (Ngày / Giờ) - Giá trị của mảng tại bắt đầu phạm vi (AFL 2.3)
34. CategoryAddSymbol (Thông tin / Danh mục) - thêm một biểu tượng vào một danh mục (AFL 2.5)
35. CategoryCreate (Thông tin / Danh mục) - thêm danh mục mới (như danh sách theo dõi) (AFL 3.70)
36. CategoryFind (Thông tin / Danh mục) - tìm kiếm danh mục theo tên (AFL 3.0)
37. CategoryGetName (Thông tin / Danh mục) - nhận tên của một danh mục (AFL 2.5)
38. CategoryGetSymbols (Thông tin / Danh mục) - lấy danh sách các biểu tượng được phân tách bằng dấu phẩy thuộc danh mục đã cho (AFL 2.5)

39. CategoryRemoveSymbol (Thông tin / Danh mục) - loại bỏ một biểu tượng khỏi một danh mục (AFL 2.5)
40. CategorySetName (Thông tin / Danh mục) - đặt tên danh mục (nhóm, thị trường, dánh sách theo dõi, ngành) (AFL 3.20)
41. CCI (Indicators) - chỉ số kênh hàng hóa
42. ceil (Math functions) - giá trị ceil
43. Chaikin (Indicators) - dao động chaikin
44. Chr (String manipulation) - lấy chuỗi có mã ASCII đã cho (AFL 4.40)
45. ClipboardGet (Miscellaneous functions) - truy xuất nội dung hiện tại của clipboard Windows (AFL 2.60)
46. ClipboardSet (Miscellaneous functions) - sao chép văn bản vào clipboard Windows (AFL 2.6)
47. ColorBlend (Indicators) - pha trộn (mixes) hai màu (AFL 3.30)
48. ColorHSB (Miscellaneous functions) - chỉ định màu bằng Hue-Saturation-Brightness (AFL 2.80)
49. ColorRGB (Miscellaneous functions) - chỉ định màu bằng các thành phần Red-Green-Blue (AFL 2.80)
50. Correlation (Statistical chức năng) - tương quan (AFL 1.4)
51. cos (Các hàm toán học) - cosine
52. cosh (Các hàm toán học) - hàm cosine hyperbolic (AFL 2.80)
53. CreateObject (Các hàm khác nhau) - tạo đối tượng COM (AFL 1.8)
54. CreateStaticObject (Các hàm khác nhau) - tạo đối tượng COM tĩnh (AFL 1.8)
55. Cross (Hộp công cụ hệ thống giao dịch) - kiểm tra chéo
56. Cum (Các đường trung bình động, tổng) - tổng tích lũy
57. CumProd (Các đường trung bình động, tổng) - tích tích lũy của tất cả các phần tử mảng (AFL 4.20)
58. Date (Date/Time) - ngày (AFL 1.1)
59. DateNum (Date/Time) - số ngày (AFL 1.4)
60. DateTime (Date/Time) - truy xuất ngày giờ đã mã hóa (AFL 2.3)
61. DateTimeAdd (Date/Time) - thêm số lượng giây/phút/giờ/ngày vào datetime (AFL 3.40)
62. DateTimeConvert (Date/Time) - chuyển đổi định dạng ngày/giờ (AFL 2.90)
63. DateTimeDiff (Date/Time) - lấy khoảng thời gian trong giây giữa hai giá trị datetime (AFL 3.30)
64. DateTimeFormat (Date/Time) - chuyển đổi datetime thành chuỗi (AFL 4.20)
65. DateTimeToStr (Date/Time) - chuyển đổi datetime thành chuỗi (AFL 2.8)
66. Day (Date/Time) - ngày trong tháng (AFL 1.4)
67. DayOfWeek (Date/Time) - ngày trong tuần (AFL 1.4)
68. DayOfYear (Date/Time) - lấy số thứ tự của ngày trong năm (AFL 2.4)
69. DaysSince1900 (Date/Time) - lấy số ngày kể từ ngày 1 tháng 1 năm 1900 (AFL 3.20)
70. DecIssues (Composites) - số vấn đề giảm (AFL 1.2)
71. DecVolume (Composites) - khối lượng vấn đề giảm (AFL 1.2)
72. DEMA (Moving averages, summation) - double exponential moving average (AFL 2.0)
73. EMA (Moving averages, summation) - exponential moving average
74. EnableRotationalTrading (Trading system toolbox) - Bật chế độ giao dịch xoay vòng của backtester (AFL 2.5)
75. EnableScript (Miscellaneous functions) - kích hoạt công cụ scripting (AFL 1.5)
76. EnableTextOutput (Miscellaneous functions) - cho phép bật hoặc tắt đầu ra văn bản (AFL 2.20)

77. EncodeColor (Exploration / Indicators) - mã hóa màu cho tiêu đề chỉ số (AFL 2.2)
78. EndValue (Ngày/Giờ) - giá trị của mảng ở cuối phạm vi đã chọn (AFL 2.3)
79. Equity (Hộp công cụ hệ thống giao dịch) - tính toán đường vốn chủ sở hữu ký hiệu đơn (AFL 2.0)
80. erf (Hàm toán học) - tính toán hàm lỗi Gauss (erf) (AFL 4.40)
81. Error (Hàm khác) - hiển thị thông báo lỗi do người dùng xác định và dừng thực thi (AFL 3.7)
82. EXP (Hàm toán học) - hàm mũ
83. ExRem (Hộp công cụ hệ thống giao dịch) - xóa tín hiệu quá mức (AFL 1.5)
84. ExRemSpan (Hộp công cụ hệ thống giao dịch) - xóa tín hiệu quá mức kéo dài số thanh nhất định (AFL 2.0)
85. fclose (Hàm nhập/xuất tệp) - đóng tệp (AFL 2.5)
86. fdelete (Hàm nhập/xuất tệp) - xóa tệp (AFL 2.70)
87. fdir (Các hàm File Input/Output) - liệt kê nội dung thư mục (AFL 3.70)
88. feof (Các hàm File Input/Output) - kiểm tra kết thúc tệp (AFL 2.5)
89. FFT (Phát hiện mẫu giá cơ bản) - thực hiện Biến đổi Fourier nhanh (AFL 2.90)
90. fgetcwd (Các hàm File Input/Output) - lấy thư mục làm việc hiện tại (AFL 4.10)
91. fgets (Các hàm File Input/Output) - lấy một chuỗi từ tệp (AFL 2.5)
92. fgetstatus (Các hàm File Input/Output) - truy xuất trạng thái/thuộc tính của tệp (AFL 2.90)
93. FindIndex (Các hàm khác) - tìm chỉ mục của mục mảng khớp với giá trị đã chỉ định (AFL 4.40)
94. FIR (Các giá trị trung bình động, tổng) - Bộ lọc phản hồi xung hữu hạn (AFL 3.40)
95. FirstVisibleValue (Các chỉ báo) - lấy giá trị đầu tiên có thể nhìn thấy của mảng (AFL 3.40)
96. Flip (Hộp công cụ hệ thống giao dịch) - (AFL 1.5)
97. floor (Các hàm toán học) - giá trị sàn
98. fmkdir (Các hàm File Input/Output) - tạo (làm) một thư mục (AFL 2.70)
99. fopen (Các hàm File Input/Output) - mở một tệp (AFL 2.5)
100. Foreign (Tham chiếu dữ liệu ký hiệu khác) - truy cập dữ liệu bảo mật nước ngoài (AFL 1.5)
101. fputs (Các hàm File Input/Output) - ghi một chuỗi vào tệp (AFL 2.5)
102. frac (Các hàm toán học) - phần phân số
103. frmdir (Các hàm File Input/Output) - xóa một thư mục (AFL 2.70)
104. FullName (Thông tin / Thể loại) - tên đầy đủ của ký hiệu (AFL 1.1)
105. GapDown (Phát hiện mô hình giá cơ bản) - gap down
106. GapUp (Nhận dạng mô hình giá cơ bản) - gap up
107. GetAsyncKeyState (Các chức năng khác) - truy vấn trạng thái hiện tại của các phím bàn phím (AFL 3.60)
108. GetBacktesterObject (Hộp công cụ hệ thống giao dịch) - nhận quyền truy cập vào đối tượng backtester (AFL 2.60)
109. GetBaseIndex (Tham chiếu dữ liệu kí hiệu khác) - truy xuất kí hiệu của chỉ số cơ sở sức mạnh tương đối (AFL 2.1)
110. GetCategorySymbols (Thông tin / Danh mục) - truy xuất danh sách các biểu tượng được phân tách bằng dấu phẩy thuộc danh mục đã cho (AFL 2.4)

111. GetChartBkColor (Chỉ số) - lấy giá trị màu RGB của nền biểu đồ (AFL 3.20)

112. GetChartID (Khám phá / Chỉ số) - lấy ID biểu đồ hiện tại (AFL 2.3)

113. GetCursorMouseButtons (Chỉ số) - lấy trạng thái hiện tại của các nút chuột (AFL 2.80)

114. GetCursorXPosition (Chỉ số) - lấy vị trí X hiện tại của con trỏ chuột (AFL 2.80)

115. GetCursorYPosition (Chỉ số) - lấy vị trí Y hiện tại của con trỏ chuột (AFL 2.80)

116. GetDatabaseName (Thông tin / Danh mục) - truy xuất tên thư mục của cơ sở dữ liệu hiện tại (AFL 2.3)

117. GetExtraData (Các chức năng khác) - lấy dữ liệu phụ từ nguồn dữ liệu bên ngoài (AFL 1.9)

118. GetExtraDataForeign (Các chức năng khác) - lấy dữ liệu phụ từ nguồn dữ liệu bên ngoài cho ký hiệu đã chỉ định (AFL 4.20)
119. GetFnData (Thông tin/Danh mục) - lấy dữ liệu cơ bản (AFL 2.90)
120. GetFnDataForeign (Thông tin/Danh mục) - lấy dữ liệu cơ bản cho ký hiệu đã chỉ định (AFL 4.20)
121. GetFormulaPath (Các hàm khác) - lấy đường dẫn tệp của công thức hiện tại (AFL 3.90)
122. GetLastOSError (Các hàm khác) - lấy văn bản lỗi hệ điều hành gần đây nhất (Windows) (AFL 4.30)
123. GetObject (Các hàm khác) - lấy xử lý đối tượng OLE đang chạy (AFL 4.40)
124. GetOption (Hộp công cụ hệ thống giao dịch) - lấy giá trị của tùy chọn trong cài đặt phân tích tự động (AFL 2.60)
125. GetPerformanceCounter (Các hàm khác) - truy xuất giá trị hiện tại của bộ đếm hiệu suất độ phân giải cao (AFL 2.90)
126. GetPlaybackDateTime (Ngày/Giờ) - lấy ngày/giờ vị trí phát lại thanh (AFL 3.0)
127. GetPriceStyle (Khám phá/Chỉ báo) - lấy kiểu biểu đồ giá hiện tại (AFL 2.70)
128. GetRTData (Các hàm khác nhau) - truy xuất các trường dữ liệu thời gian thực (AFL 2.60)
129. GetRTDataForeign (Các hàm khác nhau) - truy xuất các trường dữ liệu thời gian thực (cho ký hiệu đã chỉ định) (AFL 2.80)
130. GetScriptObject (Các hàm khác nhau) - truy cập vào đối tượng COM của tập lệnh (AFL 1.8)
131. GetTradingInterface (Hộp công cụ hệ thống giao dịch) - truy xuất đối tượng tự động hóa OLE vào giao diện giao dịch tự động (AFL 2.70)
132. GfxArc (Đồ họa cấp thấp) - vẽ một cung tròn (AFL 3.0)
133. GfxChord (Đồ họa cấp thấp) - vẽ một chord (AFL 3.0)
134. GfxCircle (Đồ họa cấp thấp) - vẽ một vòng tròn (AFL 3.0)
135. GfxDrawImage (Đồ họa cấp thấp) - vẽ ảnh bitmap (AFL 4.20)
136. GfxDrawText (Đồ họa cấp thấp) - vẽ một văn bản (được cắt thành hình chữ nhật) (AFL 3.0)
137. GfxEllipse (Đồ họa cấp thấp) - vẽ một hình elip (AFL 3.0)
138. GfxFillSolidRect (Đồ họa cấp thấp) - tô hình chữ nhật bằng màu đặc (AFL 4.10)
139. GfxGetTextWidth (Đồ họa cấp thấp) - lấy chiều rộng pixel của văn bản (AFL 2.80)

140. GfxGradientRect (Đồ họa cấp thấp) - vẽ một hình chữ nhật với tô màu chuyển sắc (AFL 3.0)
141. GfxLineTo (Đồ họa cấp thấp) - vẽ một đường thẳng đến điểm đã chỉ định (AFL 3.0)
142. GfxMoveTo (Đồ họa cấp thấp) - di chuyển con trỏ đồ họa đến vị trí mới (AFL 3.0)
143. GfxPie (Đồ họa cấp thấp) - vẽ một hình tròn (AFL 3.0)
144. GfxPolygon (Đồ họa cấp thấp) - vẽ một đa giác (AFL 3.0)
145. GfxPolyline (Đồ họa cấp thấp) - vẽ một đường đa giác (AFL 3.0)
146. GfxRectangle (Đồ họa cấp thấp) - vẽ một hình chữ nhật (AFL 3.0)
147. GfxRoundRect (Đồ họa cấp thấp) - vẽ một hình chữ nhật có các góc bo tròn (AFL 3.0)
148. GfxSelectFont (Đồ họa cấp thấp) - tạo / chọn phông chữ đồ họa (AFL 3.0)
149. GfxSelectHatchBrush (Đồ họa cấp thấp) - chọn cọ kiểu gạch chéo (AFL 4.0)
150. GfxSelectPen (Low-level graphics) - tạo / chọn bút đồ họa (AFL 3.0)
151. GfxSelectSolidBrush (Low-level graphics) - tạo / chọn bàn chải đồ họa (AFL 3.0)
152. GfxSelectStockObject (Low-level graphics) - chọn đối tượng đồ họa có sẵn (AFL 4.00)
153. GfxSetBkColor (Low-level graphics) - đặt màu nền đồ họa (AFL 3.0)
154. GfxSetBkMode (Low-level graphics) - đặt chế độ nền đồ họa (AFL 3.0)
155. GfxSetCoordsMode (Low-level graphics) - đặt chế độ toạ độ đồ họa cấp thấp (AFL 2.80)
156. GfxSetOverlayMode (Low-level graphics) - đặt chế độ superimpose đồ họa cấp thấp (AFL 3.0)
157. GfxSetPixel (Low-level graphics) - đặt pixel ở vị trí chỉ định với màu sắc chỉ định (AFL 3.0)
158. GfxSetTextAlign (Low-level graphics) - đặt căn chỉnh văn bản (AFL 3.0)
159. GfxSetTextColor (Low-level graphics) - đặt màu văn bản đồ họa (AFL 3.0)
160. GfxSetZOrder (Low-level graphics) - đặt lớp Z-order đồ họa cấp thấp hiện tại (AFL 2.80)
161. GfxTextOut (Low-level graphics) - viết văn bản tại vị trí chỉ định (AFL 3.0)
162. GicsID (Thông tin / Danh mục) - lấy thông tin danh mục GICS (AFL 3.40)
163. GroupID (Thông tin / Danh mục) - lấy ID/nhóm tên (AFL 1.8)
164. GuiButton (Các hàm GUI) - tạo nút điều khiển trên biểu đồ (AFL 4.30)
165. GuiCheckBox (Các hàm GUI) - tạo hộp kiểm trên biểu đồ (AFL 4.30)
166. GuiDateTime (Các hàm GUI) - tạo trình chọn ngày-giờ trên biểu đồ (AFL 4.30)
167. GuiEdit (Các hàm GUI) - tạo điều khiển chỉnh sửa trên biểu đồ (AFL 4.30)
168. GuiEnable (Các hàm GUI) - cho phép hoặc ẩn điều khiển trên biểu đồ (AFL 4.30)
169. GuiGetCheck (Các hàm GUI) - lấy trạng thái kiểm tra của điều khiển (AFL 4.30)
170. GuiGetEvent (Các hàm GUI) - lấy sự kiện GUI (AFL 4.30)
171. GuiGetText (Các hàm GUI) - lấy văn bản từ điều khiển trên biểu đồ (AFL 4.30)
172. GuiGetValue (Các hàm GUI) - lấy giá trị số học từ điều khiển trên biểu đồ (AFL 4.30)
173. GuiRadio (Các hàm GUI) - tạo nút radio trên biểu đồ (AFL 4.30)
174. GuiSendKeyEvents (Các hàm GUI) - yêu cầu thông báo GUI cho
175. 175. GuiSetCheck (các hàm GUI) - đặt trạng thái đã kiểm tra của điều khiển trên biểu đồ (AFL 4.30)
176. GuiSetColors (các hàm GUI) - đặt màu cho các điều khiển GUI (AFL 2.30)
177. GuiSetFont (các hàm GUI) - đặt phông chữ cho điều khiển trên biểu đồ (AFL 4.30)
178. GuiSetRange (các hàm GUI) - đặt phạm vi điều khiển thanh trượt (AFL 4.30)

179. GuiSetText (các hàm GUI) - đặt giá trị văn bản của điều khiển trên biểu đồ (AFL 4.30)
180. GuiSetValue (các hàm GUI) - đặt giá trị số của điều khiển trên biểu đồ (AFL 4.30)
181. GuiSetVisible (các hàm GUI) - hiển thị hoặc ẩn điều khiển trên biểu đồ (AFL 4.30)
182. GuiSlider (các hàm GUI) - tạo điều khiển thanh trượt trên biểu đồ (AFL 4.30)
183. HHV (Thấp nhất/Cao nhất) - giá trị cao nhất
184. HHVBars (Thấp nhất/Cao nhất) - thanh kể từ giá cao nhất
185. Highest (Thấp nhất/Cao nhất) - giá trị cao nhất
186. HighestBars (Thấp nhất/Cao nhất) - thanh kể từ giá trị cao nhất
187. HighestSince (Thấp nhất/Cao nhất) - giá trị cao nhất kể từ khi điều kiện được đáp ứng (AFL 1.4)
188. HighestSinceBars (Thấp nhất/Cao nhất) - thanh kể từ khi giá trị cao nhất kể từ khi điều kiện được đáp ứng (AFL 1.4)
189. HighestVisibleValue (Chỉ báo) - lấy giá trị cao nhất trong vùng biểu đồ có thể nhìn thấy (AFL 3.30)
190. HMA (Đường trung bình động, tổng hợp) - Đường trung bình động Hull (AFL 3.40)
191. Hold (Hộp công cụ hệ thống giao dịch) - giữ tín hiệu cảnh báo
192. Hour (Ngày/Giờ) - lấy giờ của thanh hiện tại (AFL 2.0)
193. IcbID (Thông tin/Danh mục) - lấy thông tin danh mục ICB (AFL 3.60)
194. IIf (Hộp công cụ hệ thống giao dịch) - hàm IF tức thời
195. IIR (Đường trung bình động, tổng) - bộ lọc phản hồi xung vô hạn (AFL 4.0)
196. IndustryID (Thông tin/Danh mục) - lấy ID/tên ngành (AFL 1.8)
197. InGICS (Thông tin/Danh mục) - kiểm tra tư cách thành viên GICS (AFL 3.40)
198. InICB (Thông tin/Danh mục) - kiểm tra tư cách thành viên ICB (AFL 3.60)
199. Inside (Phát hiện mô hình giá cơ bản) - bên trong ngày
200. Int (Hàm toán học) - phần nguyên
201. InternetClose (Hàm đầu vào/đầu ra tệp) - đóng trình xử lý tệp Internet (AFL 4.20)
202. InternetGetStatusCode (Các hàm File Input/Output) - trả về mã trạng thái HTTP của cuộc gọi Internet cuối cùng (AFL 4.40).
203. InternetOpenURL (Các hàm File Input/Output) - mở tài nguyên web Internet (URL) (AFL 4.20)
204. InternetPostRequest (Các hàm File Input/Output) - gửi yêu cầu HTTP Post đến tài nguyên web Internet (URL) (AFL 4.30)
205. InternetReadString (Các hàm File Input/Output) - đọc một chuỗi từ tài nguyên Internet (AFL 4.20)
206. InternetSetAgent (Các hàm File Input/Output) - đặt chuỗi agent cho chức năng Internet (AFL 4.30)
207. InternetSetHeaders (Các hàm File Input/Output) - đặt tiêu đề HTTP tùy chỉnh cho các yêu cầu web tiếp theo (AFL 4.40)
208. InternetSetOption (Các hàm File Input/Output) - đặt tùy chọn HTTP cho phiên Internet (AFL 4.40)
209. Interval (Date/Time) - lấy khoảng thời gian (theo giây) (AFL 2.1)
210. inverf (Hàm toán học) - hàm Gauss erf nghịch đảo (AFL 4.40)
211. InWatchList (Thông tin / Danh mục) - thử nghiệm thành viên danh sách theo dõi (theo số thứ tự)
212. InWatchListName (Thông tin / Danh mục) - thử nghiệm thành viên danh sách theo dõi (theo tên) (AFL 3.0)

213. IsContinuous (Thông tin/Danh mục) - kiểm tra trạng thái cờ 'trích dẫn liên tục' (AFL 2.60)
214. IsEmpty (Các hàm khác nhau) - kiểm tra giá trị rỗng (AFL 1.5)
215. IsFavorite (Thông tin/Danh mục) - kiểm tra xem ký hiệu hiện tại có thuộc mục yêu thích không (AFL 2.5)
216. IsFinite (Các hàm khác nhau) - kiểm tra xem giá trị không phải là vô hạn (AFL 2.3)
217. IsIndex (Thông tin/Danh mục) - kiểm tra xem ký hiệu hiện tại có phải là chỉ mục không (AFL 2.5)
218. IsNan (Các hàm khác nhau) - kiểm tra NaN (không phải là số) (AFL 2.3)
219. IsNull (Các hàm khác nhau) - kiểm tra giá trị Null (rỗng) (AFL 2.3)
220. IsTrue (Các hàm khác nhau) - giá trị true (không rỗng và khác không) kiểm tra (AFL 1.5)
221. Kurtosis (Hàm toán học) - tính toán độ nhọn (AFL 4.20)
222. LastValue (Hộp công cụ hệ thống giao dịch) - giá trị cuối cùng của mảng
223. LastVisibleValue (Chỉ báo) - lấy giá trị hiển thị cuối cùng của mảng (AFL 3.40)
224. LineArray (Khám phá / Chỉ báo) - tạo mảng đường xu hướng (AFL 2.5)
225. LinearReg (Hàm thống kê) - điểm cuối hồi quy tuyến tính (AFL 2.2)
226. LinRegIntercept (Hàm thống kê) - (AFL 2.2)
227. LinRegSlope (Hàm thống kê) - độ dốc hồi quy tuyến tính (AFL 1.4)
228. LLV (Thấp nhất/Cao nhất) - giá trị thấp nhất
229. LLVBars (Thấp nhất/Cao nhất) - các thanh kể từ mức thấp nhất
230. log (Math chức năng) - logarit tự nhiên
231. log10 (Các hàm toán học) - logarit thập phân
232. Lookup (Ngày/Giờ) - tìm kiếm mảng thanh có ngày/giờ được chỉ định (AFL 3.40)
233. Lowest (Thấp nhất/Cao nhất) - giá trị thấp nhất
234. LowestBars (Thấp nhất/Cao nhất) - thanh kể từ thấp nhất
235. LowestSince (Thấp nhất/Cao nhất) - giá trị thấp nhất kể từ khi điều kiện được đáp ứng (AFL 1.4)
236. LowestSinceBars (Thấp nhất/Cao nhất) - barssince giá trị thấp nhất kể từ khi điều kiện được đáp ứng (AFL 1.4)
237. LowestVisibleValue (Các chỉ báo) - lấy giá trị thấp nhất trong vùng biểu đồ có thể nhìn thấy (AFL 3.30)
238. MA (Các đường trung bình động, tổng) - đường trung bình động đơn giản
239. MACD (Các chỉ báo) - sự hội tụ/phân kỳ của đường trung bình động
240. MapCreate (Các hàm bản đồ) - tạo đối tượng bản đồ/từ điển (giữ cặp khóa-giá trị) (AFL 4.90)
241. MarketID (Thông tin/Danh mục) - ID/tên thị trường (AFL 1.8)
242. Matrix (Các hàm ma trận) - tạo ma trận mới (AFL 4.0)
243. Max (Các hàm toán học) - giá trị lớn nhất của hai số/mảng
244. MDI (Các chỉ báo) - chỉ báo chuyển động theo hướng trừ (-DI) (AFL 1.3)
245. Median (Các hàm thống kê) - tính trung vị (phần tử ở giữa) (AFL 2.5)
246. MFI (Các chỉ báo) - chỉ số dòng tiền
247. MicroSec (Ngày/Giờ) - lấy phần micro giây của thanh trong dấu thời gian
248. MilliSec (Ngày/Giờ) - lấy phần mili giây của thanh trong dấu thời gian
249. Min (Các hàm toán học) - giá trị nhỏ nhất của hai số/mảng
250. Minute (Ngày/Giờ) - lấy phút của thanh hiện tại (AFL 2.0)
251. Month (Ngày/Thời gian) - Tháng (AFL 1.4)
252. mtRandom (Các hàm thống kê) - Bộ sinh số ngẫu nhiên Mersene Twister (AFL 3.0)

253. mtRandomA (Các hàm thống kê) - Bộ sinh số ngẫu nhiên Mersene Twister (phiên bản mảng) (AFL 3.0)
254. MxCopy (Hàm ma trận) - sao chép khối hình chữ nhật từ một ma trận sang ma trận khác (AFL 4.40)
255. MxDet (Hàm ma trận) - Tính định thức của ma trận (AFL 4.10)
256. MxFromString (Hàm ma trận) - Tạo một ma trận mới từ chuỗi (AFL 4.10)
257. MxGetBlock (Hàm ma trận) - Chỉ định khối hình chữ nhật của các mục từ ma trận (AFL 4.10)
258. MxGetSize (Hàm ma trận) - Xác định kích thước của ma trận (AFL 4.0)
259. MxIdentity (Hàm ma trận) - Tạo ma trận đơn vị (AFL 4.0)
260. MxInverse (Hàm ma trận) - Tính ma trận nghịch đảo (AFL 4.10)
261. MxSetBlock (Hàm ma trận) - Đặt các giá trị trong khối hình chữ nhật của các ô ma trận (AFL 4.10)
262. MxSolve (Hàm ma trận) - Giải hệ phương trình tuyến tính A @ X = B (AFL 4.10)
263. MxSort (Hàm ma trận) - Sắp xếp ma trận (AFL 4.10)
264. MxSortRows (Hàm ma trận) - Sắp xếp các hàng của ma trận (AFL 4.10)
265. MxSum (Hàm ma trận) - Tính tổng cộng của ma trận (AFL 4.20)
266. MxToString (Hàm ma trận) - Chuyển đổi ma trận thành chuỗi (AFL 4.10)
267. MxTranspose (Hàm ma trận) - Tạo chuyển vị của ma trận đầu vào (AFL 4.0)
268. Name (Thông tin / Danh mục) - Biểu tượng giao dịch (AFL 1.1)
269. NormDist (Các hàm thống kê) - Hàm phân phối chuẩn (AFL 4.20)
270. NoteGet (Các hàm khác) - Truy xuất text của ghi chú (AFL 2.6)
271. NoteSet (Các hàm khác) - Đặt text cho ghi chú (AFL 2.6)
272. Now (Date/Time) - lấy ngày/giờ hệ thống hiện tại (AFL 2.3)
273. NullCount (Các hàm khác nhau) - đếm các giá trị Null liên tiếp (AFL 3.90)
274. NumToStr (Thao tác chuỗi) - chuyển đổi số thành chuỗi (AFL 2.5)
275. NVI (Các chỉ báo) - chỉ số khối lượng âm
276. Nz (Các hàm khác nhau) - Null (Null/Nan/Infinity) thành 0 (AFL 2.3)
277. OBV (Các chỉ báo) - khối lượng cân bằng
278. Optimize (Hộp công cụ hệ thống giao dịch) - xác định biến tối ưu hóa (AFL 1.7)
279. OptimizerSetEngine (Hộp công cụ hệ thống giao dịch) - chọn công cụ tối ưu hóa bên ngoài (AFL 3.20)
280. OptimizerSetOption (Hộp công cụ hệ thống giao dịch) - đặt giá trị của tham số công cụ tối ưu hóa bên ngoài (AFL 3.20)
281. OscP (Chỉ báo) - dao động giá
282. OscV (Chỉ báo) - dao động khối lượng
283. Outside (Phát hiện mô hình giá cơ bản) - thanh bên ngoài
284. Param (Khám phá / Chỉ báo) - thêm tham số số do người dùng định nghĩa (AFL 2.3)
285. ParamColor (Khám phá / Chỉ báo) - thêm tham số màu do người dùng định nghĩa (AFL 2.3)
286. ParamDate (Khám phá / Chỉ báo) - thêm tham số ngày do người dùng định nghĩa (AFL 2.60)
287. ParamField (Khám phá / Chỉ báo) - tạo tham số trường giá (AFL 2.70)
288. ParamList (Khám phá / Chỉ báo) - tạo tham số bao gồm danh sách các lựa chọn (AFL 2.70)
289. ParamStr (Khám phá / Chỉ báo) - thêm tham số chuỗi do người dùng định nghĩa (AFL 2.3)
290. ParamStyle (Khám phá / Chỉ báo) - chọn kiểu áp dụng cho biểu đồ (AFL 2.70)

291. ParamTime (Khám phá / Chỉ báo) - thêm tham số thời gian do người dùng định nghĩa (AFL 2.60)
292. ParamToggle (Khám phá / Chỉ báo) - tạo tham số Có/Không (AFL 2.70)
293. ParamTrigger (Khám phá / Chỉ báo) - tạo một trình kích hoạt (nút) trong hộp thoại tham số (AFL 2.70)
294. PDI (Chỉ báo) - cộng với chỉ báo chuyển động theo hướng (AFL 1.3)
295. Peak (Phát hiện mô hình giá cơ bản) - đỉnh (AFL 1.1)
296. PeakBars (Phát hiện mô hình giá cơ bản) - các thanh kể từ đỉnh (AFL 1.1)
297. Percentile (Hàm thống kê) - tính toán phần trăm (AFL 2.5)
298. PercentRank (Các chỉ số) - tính toán thứ hạng phần trăm (AFL 3.40)
299. PlaySound (Các hàm khác) - phát lại tệp .WAV đã chỉ định (AFL 3.40)
300. Plot (Khám phá / Chỉ số) - vẽ biểu đồ chỉ số (AFL 1.8)
301. PlotForeign (Tham chiếu dữ liệu biểu tượng khác) - vẽ dữ liệu bảo mật nước ngoài (AFL 2.2)
302. PlotGrid (Khám phá / Chỉ số) - Vẽ đường lưới ngang (AFL 2.3)
303. PlotOHLC (Khám phá / Chỉ số) - vẽ biểu đồ OHLC tùy chỉnh (AFL 2.2)
304. PlotShapes (Khám phá / Chỉ số) - vẽ mũi tên và hình dạng khác (AFL 2.3)
305. PlotText (Chỉ số) - viết văn bản trên biểu đồ (AFL 2.80)
306. PlotTextSetFont (Phát hiện mô hình giá cơ bản) - viết văn bản trên biểu đồ với phông chữ do người dùng định nghĩa (AFL 2.80)
307. PlotVAPOverlay (Khám phá / Chỉ số) - vẽ biểu đồ Vùng-At-Gía overlay (AFL 2.4)
308. PlotVAPOverlayA (Chỉ số) - vẽ biểu đồ Volume-At-Price nhiều phân đoạn (AFL 3.20)
309. PopupWindow (Chức năng linh tinh) - hiển thị cửa sổ pop-up (AFL 3.0)
310. Prec (Hàm toán học) - điều chỉnh số lượng điểm thập phân của số điểm số (AFL 4.2)
311. Prefs (Chức năng linh tinh) - truy xuất cài đặt tùy chọn (AFL 1.4)
312. PriceVolDistribution (Hàm ma trận) - hàm phân phối đa dụng (AFL 4.20)
313. printf (Sự thay đổi chuỗi) - In đầu ra được định dạng lên cửa sổ đầu ra (AFL 2.5)
314. Prod (Trung bình di chuyển, tổng) - sản phẩm tích lũy của mảng trên phạm vi chỉ định (AFL 4.20)
315. ProdSince (Trung bình di chuyển, tổng) - sản phẩm tích lũy kể từ khi điều kiện được đáp ứng (AFL 4.20)
316. PVI (Chỉ số) - chỉ số khối lượng tích cực
317. Random (Chức năng thống kê) - số ngẫu nhiên (AFL 1.9)
318. Ref (Bộ công cụ hệ thống giao dịch) - tham chiếu các giá trị quá khứ/tương lai của mảng
319. RelStrength (Tham chiếu dữ liệu biểu tượng khác) - sức mạnh tương đối so sánh (AFL 1.3)
320. Remap (Hàm toán học) - remapping một phạm vi sang một phạm vi khác (AFL 4.30)
321. RequestTimedRefresh (Chỉ số) - buộc làm mới tấm bảng chỉ báo theo chu kỳ (AFL 2.90)
322. RequestMouseMoveRefresh (Chỉ báo) - yêu cầu biểu đồ được làm mới khi người dùng di chuyển con trỏ chuột (AFL 4.30)
323. RestorePriceArrays (Tham chiếu dữ liệu biểu tượng khác) - trả lại
324. Reverse (Các hàm khác nhau) - đảo ngược thứ tự các phần tử trong mảng (AFL 3.90)
325. RMI (Các chỉ báo) - Chỉ số động lượng tương đối (AFL 2.1)
326. ROC (Các chỉ báo) - tỷ lệ phần trăm thay đổi
327. Round (Các hàm toán học) - làm tròn số đến số nguyên gần nhất

328. RSI (Các chỉ báo) - chỉ số sức mạnh tương đối
329. RWI (Các chỉ báo) - chỉ số bước đi ngẫu nhiên
330. RWIHi (Các chỉ báo) - chỉ số bước đi ngẫu nhiên của mức cao
331. RWILo (Các chỉ báo) - chỉ số bước đi ngẫu nhiên của mức thấp
332. SafeDivide (Các hàm toán học) - phép chia có bảo vệ chia cho không (AFL 4.40)
333. SAR (Các chỉ báo) - dừng và đảo ngược parabol (AFL 1.3)
334. Say (Các hàm chuyển văn bản thành giọng nói) - đọc văn bản được cung cấp (AFL 2.90)
335. Second (Ngày/Giờ) - lấy thanh hiện tại second (AFL 2.0)
336. SectorID (Thông tin/Danh mục) - lấy ID/tên sector (AFL 1.8)
337. SelectedValue (Khám phá/Chỉ báo) - truy xuất giá trị của mảng tại thời điểm ngày/giờ hiện tại được chọn (AFL 2.1)
338. SendEmail (Các hàm khác) - gửi tin nhắn email (AFL 3.90)
339. SetBacktestMode (Hộp công cụ hệ thống giao dịch) - Đặt chế độ làm việc của trình kiểm tra ngược (AFL 3.0)
340. SetBarFillColor (Các chỉ báo) - đặt màu tô biểu đồ thanh/nến/đám mây (AFL 3.1)
341. SetBarsRequired (Các hàm khác) - đặt số lượng thanh trước đó và tương lai cần thiết để tập lệnh/DLL thực thi đúng (AFL 2.1)
342. SetChartBkColor (Các chỉ báo) - đặt màu nền của biểu đồ (AFL 2.80)
343. SetChartBkGradientFill (Chỉ báo) - cho phép tô màu nền gradient trong các chỉ báo (AFL 2.90)
344. SetChartOptions (Khám phá / Chỉ báo) - đặt/xóa/ghi đè các giá trị mặc định cho các tùy chọn khung biểu đồ (AFL 2.70)
345. SetCustomBacktestProc (Hộp công cụ Hệ thống Giao dịch) - định nghĩa tệp công thức cho quy trình backtest tùy chỉnh (AFL 2.70)
346. SetForeign (Tham chiếu dữ liệu biểu tượng khác) - thay thế mảng giá hiện tại với mảng giá của bảo mật nước ngoài (AFL 2.5)
347. SetFormulaName (Hộp công cụ Hệ thống Giao dịch) - đặt tên cho công thức (AFL 2.5)
348. SetGradientFill (Chỉ số) - đặt màu sắc cho biểu đồ điền màu gradient (AFL 3.60)
349. SetOption (Hộp công cụ Hệ thống Giao dịch) - đặt các tùy chọn trong cài đặt phân tích tự động (AFL 2.3)
350. SetPositionSize (Hộp công cụ Hệ thống Giao dịch) - đặt kích thước giao dịch (AFL 2.70)
351. SetSortColumns (Khám phá / Chỉ số) - đặt các cột sẽ được sắp xếp trong cửa sổ AA (AFL 2.90)
352. SetStopPrecedence (Hộp công cụ Hệ thống Giao dịch) - đặt sự ưu tiên của các dừng tích hợp (AFL 4.0)
353. SetTradeDelays (Hộp công cụ Hệ thống Giao dịch) - cho phép kiểm soát các trễ giao dịch được áp dụng bởi backtester (AFL 2.1)
354. ShellExecute (Patter giá cơ bản) - thực hiện một tệp (AFL 3.40)
355. sign (Hàm Toán) - trả lại dấu của số / mảng (AFL 2.50)
356. Signal (Các chỉ báo) - đường tín hiệu macd
357. sin (Các hàm toán học) - hàm sin
358. sinh (Các hàm toán học) - hàm sin hyperbolic (AFL 2.80)
359. Skewness (Các hàm toán học) - tính độ lệch (AFL 4.20)
360. Sort (Các hàm khác) - thực hiện sắp xếp nhanh mảng (AFL 3.90)
361. SparseCompress (Các hàm khác) - nén mảng thưa (AFL 4.0)

362. SparseExpand (Các hàm khác) - mở rộng mảng đã nén thành mảng thưa (AFL 4.0)
363. SparseInterpolate (Các hàm khác) - nội suy các giá trị giữa các điểm thưa được đưa vào làm đầu vào (AFL 4.90)
364. sqrt (Các hàm toán học) - căn bậc hai
365. StaticVarAdd (Các hàm khác) - phép cộng "nguyên tử" cho các biến tĩnh (AFL 4.10)
366. StaticVarCompareExchange (Các hàm khác nhau) - hoạt động so sánh-trao đổi biến tĩnh liên kết nguyên tử (AFL 3.50)
367. StaticVarCount (Các hàm khác nhau) - lấy tổng số biến tĩnh trong bộ nhớ (AFL 3.30)
368. StaticVarGenerateRanks (Các hàm khác nhau) - tạo thứ hạng của nhiều ký hiệu và lưu trữ vào các biến tĩnh (AFL 3.70)
369. StaticVarGet (Các hàm khác nhau) - lấy giá trị của biến tĩnh (AFL 2.60)
370. StaticVarGetRankedSymbols (Các hàm khác nhau) - lấy danh sách các ký hiệu được xếp hạng từ các biến tĩnh (AFL 3.70)
371. StaticVarGetText (Các hàm khác nhau) - lấy giá trị của biến tĩnh dưới dạng chuỗi (AFL 2.60)
372. StaticVarInfo (Các hàm khác nhau) - lấy thông tin về biến tĩnh (AFL 3.60)
373. StaticVarRemove (Các hàm khác nhau) - xóa biến tĩnh (AFL 2.80)
374. StaticVarSet (Các hàm khác nhau) - đặt giá trị của biến tĩnh (AFL 2.60)
375. StaticVarSetText (Các hàm khác nhau) - Đặt giá trị của biến chuỗi tĩnh. (AFL 2.60)
376. Status (Các hàm khác nhau) - lấy thông tin trạng thái AFL thời gian chạy (AFL 1.65)
377. StdErr (Các hàm thống kê) - lỗi chuẩn (AFL 1.4)
378. StDev (Các hàm thống kê) - độ lệch chuẩn (AFL 1.4)
379. StochD (Các chỉ số) - chậm ngẫu nhiên %D
380. StochK (Các chỉ số) - chậm ngẫu nhiên %K
381. StrCount (Điều khiển chuỗi) - đếm số lần xuất hiện của chuỗi con trong một chuỗi (AFL 3.20)
382. StrExtract (Điều khiển chuỗi) - trích xuất mục đã cho (chuỗi con) từ chuỗi được phân tách bằng dấu phẩy (AFL 2.4)
383. StrFind (Điều khiển chuỗi) - tìm chuỗi con trong một chuỗi (AFL 2.5)
384. StrFormat (Điều khiển chuỗi) - Ghi đầu ra được định dạng vào chuỗi (AFL 2.5)
385. StrLeft (Điều khiển chuỗi) - trích xuất phần ngoài cùng bên trái (AFL 2.0)
386. StrLen (Thao tác chuỗi) - độ dài chuỗi (AFL 1.5)
387. StrMatch (String manipulation) - khớp mô hình/chuỗi ký tự (AFL 4.0)
388. StrMid (String manipulation) - trích xuất một phần của chuỗi (AFL 2.0)
389. StrReplace (String manipulation) - thay thế chuỗi (AFL 2.90)
390. StrRight (String manipulation) - trích xuất phần cuối của chuỗi (AFL 2.0)
391. StrSort (String manipulation) - sắp xếp danh sách các mục phân tách bởi dấu phẩy (AFL 3.90)
392. StrToDateTime (String manipulation) - chuyển đổi chuỗi thành datetime (AFL 2.80)
393. StrToLower (String manipulation) - chuyển đổi thành chữ thường (AFL 2.80)
394. StrToNum (String manipulation) - chuyển đổi chuỗi thành số (AFL 2.5)
395. StrToUpper (String manipulation) - chuyển đổi thành chữ hoa (AFL 2.80)
396. StrTrim (String manipulation) - loại bỏ khoảng trắng từ chuỗi (AFL 3.90)
397. Study (Miscellaneous functions) - tham khảo nghiên cứu vẽ tay (AFL 1.5)
398. Sum (Moving averages, summation) - tổng dữ liệu sau một số lượng cụ thể của thanh
399. SumSince (Moving averages, summation) - tổng của các phần tử mảng kể từ khi điều kiện đúng (AFL 4.10)

400. tan (Math functions) - hàm tangent (AFL 1.0)
401. tanh (Math functions) - hàm tangent siêu cực (AFL 2.80)
402. TEMA (Moving averages, summation) - trung bình di chuyển ba lần mũ (AFL 2.0)
403. ThreadSleep (Miscellaneous functions) - tạm ngừng luồng cho một số lượng mili giây cụ thể (AFL 3.50)
404. TimeFrameCompress (Time Frame functions) - nén mảng đơn thành khung thời gian đã cho (AFL 2.5)
405. TimeFrameExpand (Time Frame functions) - mở rộng mảng khung thời gian đã nén (AFL 2.5)
406. TimeFrameGetPrice (Time Frame functions) - lấy giá trị O, H, L, C, V từ khung thời gian khác (AFL 2.5)
407. TimeFrameMode (Time Frame functions) - chuyển đổi chế độ nén khung thời gian (AFL 2.80)
408. TimeFrameRestore (Time Frame functions) - khôi phục mảng giá thành khung thời gian ban đầu (AFL 2.5)
409. TimeFrameSet (Time Frame functions) - chuyển mảng giá sang khung thời gian khác (AFL 2.5)
410. TimeNum (Date/Time) - lấy thời gian thanh hiện tại (AFL 2.0)
411. TrimResultRows (Exploration/Indicators) - cắt đối tượng phân tích kết quả thành một số hàng cụ thể (AFL 4.90)
412. Trin (Composites) - Traders (Arms) index (AFL 1.2)
413. TRIX (Indicators) - giá mịn hoá ba lần mũ
414. Trough (Basic price pattern detection) - đỉnh (AFL 1.1)
415. TroughBars (Basic price
416. TSF (Các hàm thống kê) - dự báo chuỗi thời gian (AFL 2.2)
417. Ultimate (Các chỉ báo) - dao động cuối cùng
418. UncIssues (Các hợp chất) - các vấn đề không thay đổi (AFL 1.2)
419. UncVolume (Các hợp chất) - khối lượng vấn đề không thay đổi (AFL 1.2)
420. ValueWhen (Hộp công cụ hệ thống giao dịch) - lấy giá trị của mảng khi điều kiện được đáp ứng (AFL 1.1)
421. VarGet (Các hàm khác nhau) - lấy giá trị của biến động (AFL 2.60)
422. VarGetText (Các hàm khác nhau) - lấy giá trị văn bản của biến động (AFL 2.80)
423. VarSet (Các hàm khác nhau) - đặt giá trị của biến động (AFL 2.60)
424. VarSetText (Các hàm khác nhau) - đặt biến động của kiểu chuỗi (AFL 2.80)
425. Version (Các hàm khác nhau) - lấy thông tin phiên bản (AFL 1.9)
426. VoiceCount (Chức năng Text-to-Speech) - lấy số lượng giọng nói SAPI (AFL 4.20)
427. VoiceSelect (Chức năng Text-to-Speech) - chọn giọng nói SAPI (AFL 4.20)
428. VoiceSetRate (Chức năng Text-to-Speech) - đặt tốc độ giọng nói (AFL 4.30)
429. VoiceSetVolume (Chức năng Text-to-Speech) - đặt âm lượng giọng nói (AFL 4.30)
430. VoiceWaitUntilDone (Chức năng Text-to-Speech) - đợi cho đến khi giọng nói TTS nói xong (AFL 4.30)
431. Wilders (Trung bình động, tổng) - Làm mịn Wilder (AFL 1.4)
432. WMA (Trung bình động, tổng) - Trung bình động có trọng số (AFL 2.0)
433. WriteIf (Khám phá / Chỉ báo) - đầu ra văn bản có điều kiện bình luận
434. WriteVal (Khám phá / Chỉ số) - chuyển đổi số thành chuỗi
435. XYChartAddPoint (Khám phá / Chỉ số) - thêm điểm vào biểu đồ phân tán (XY) cho việc khám phá (AFL 3.60)

436. XYChartSetAxis (Khám phá / Chỉ số) - đặt tên cho trục X và Y trong biểu đồ phân tán khi khám phá (AFL 3.60)
437. Year (Ngày/Thời gian) - năm (AFL 1.4)
438. ZIG (Nhận biết mô hình giá cơ bản) - chỉ số zig-zag (AFL 1.1)
439. _DEFAULT_NAME (Khám phá / Chỉ số) - lấy tên mặc định của plot (AFL 2.70)
440. _DT (Ngày/Thời gian) - chuyển đổi chuỗi thành datetime (AFL 3.40)
441. _exit (Các chức năng khác) - kết thúc việc thực thi công thức AFL (AFL 4.40)
442. _N (Khám phá / Chỉ số) - không xuất văn bản (AFL 2.1)
443. _PARAM_VALUES (Khám phá / Chỉ số) - truy xuất chuỗi giá trị param (AFL 2.70)
444. _SECTION_BEGIN (Khám phá / Chỉ số) - dấu bắt đầu phần (AFL 2.70)
445. _SECTION_END (Khám phá / Chỉ số) - dấu kết thúc phần (AFL 2.70)
446. _SECTION_NAME (Khám phá / Chỉ số) - truy xuất tên phần hiện tại (AFL 2.70)
447. _TRACE (Các chức năng khác) - in văn bản ra trình xem gỡ lỗi hệ thống (AFL 2.4)
448. _TRACEF (Các chức năng khác) - in văn bản định dạng ra trình xem gỡ lỗi hệ thống (AFL 4.0)

Danh Sách Tham Chiếu Hàm Afl - Danh Sách Chức Năng Nhóm Cơ Bản Nhận Biết Mô Hình Giá

Phát hiện mô hình giá cơ bản

- FFT - thực hiện Biến đổi Fourier Nhanh (AFL 2.90)
- GapDown - khoảng trống xuống
- GapUp - khoảng trống lên
- Inside - trong ngày
- Outside - bar ngoài
- Peak - peak (AFL 1.1)
- PeakBars - số thanh kể từ peak (AFL 1.1)
- PlotTextSetFont - viết văn bản trên biểu đồ bằng font được định nghĩa bởi người dùng (AFL 2.80)
- ShellExecute - thực thi một tệp (AFL 3.40)
- Trough - trough (AFL 1.1)
- TroughBars - số thanh kể từ trough (AFL 1.1)
- ZIG - chỉ số zig-zag (AFL 1.1)

Composites

- AddToComposite - thêm giá trị vào composite ticker (AFL 2.0)
- ADLine - dòng tiến/triệt (AFL 1.2)
- AdvIssues - vấn đề tiến bộ (AFL 1.2)
- AdvVolume - khối lượng vấn đề tiến bộ (AFL 1.2)
- DecIssues - các vấn đề đang giảm (AFL 1.2)
- DecVolume - khối lượng vấn đề đang giảm (AFL 1.2)
- Trin - chỉ số của nhà giao dịch (Arms) (AFL 1.2)
- UncIssues - các vấn đề không thay đổi (AFL 1.2)
- UncVolume - khối lượng vấn đề không thay đổi (AFL 1.2)

Ngày/Giờ

- BarIndex - lấy số thanh bắt đầu từ số không (AFL 2.3)
- BeginValue - Giá trị của mảng ở đầu phạm vi (AFL 2.3)
- Date - ngày (AFL 1.1)
- DateNum - số thứ tự ngày (AFL 1.4)
- DateTime - truy xuất thời gian ngày mã hóa (AFL 2.3)
- DateTimeAdd - thêm số lượng giây/phút/giờ/ngày vào thời gian (AFL 3.40)
- DateTimeConvert - chuyển đổi format ngày/thời gian (AFL 2.90)
- DateTimeDiff - lấy thời gian chênh lệch trong giây giữa hai giá trị thời gian (AFL 3.30)
- DateTimeFormat - chuyển đổi thời gian sang chuỗi (AFL 4.20)
- DateTimeToStr - chuyển đổi thời gian thành chuỗi (AFL 2.8)
- Day - ngày trong tháng (AFL 1.4)
- DayOfWeek - ngày trong tuần (AFL 1.4)
- DayOfYear - lấy số thứ tự ngày trong năm (AFL 2.4)
- DaysSince1900 - lấy số ngày kể từ ngày 1/1/1900 (AFL 3.20)
- EndValue - giá trị của mảng ở cuối phạm vi đã chọn (AFL 2.3)
- GetPlaybackDateTime - lấy vị trí phát lại thanh ngày/giờ (AFL 3.0)
- Hour - lấy giờ hiện tại (AFL 2.0)
- Interval - lấy khoảng thời gian (theo giây) (AFL 2.1)
- Lookup - tìm mảng cho thanh với ngày/giờ đã chỉ định (AFL 3.40)
- MicroSec - lấy phần micro giây của thời gian chuỗi
- MilliSec - lấy phần mili giây của thời gian chuỗi
- Minute - lấy phút hiện tại (AFL 2.0)
- Month - tháng (AFL 1.4)
- Now - lấy ngày/giờ hệ thống hiện tại (AFL 2.3)
- Second - lấy giây hiện tại (AFL 2.0)
- TimeNum - lấy thời gian hiện tại (AFL 2.0)
- Year - năm (AFL 1.4)
- _DT - chuyển đổi chuỗi sang datetime (AFL 3.40)

Chỉ báo

- AccDist - tích lũy/phân phối
- ADX - chỉ báo chuyển động trung bình (AFL 1.3)
- ATR - phạm vi thực chính xác trung bình (AFL 1.3)
- BBandBot - vị trí dưới cùng của bollinger band
- BBandTop - vị trí trên cùng của bollinger band
- CCI - chỉ số kênh hàng hóa
- Chaikin - dao động Chaikin
- ColorBlend - pha trộn (kết hợp) hai màu (AFL 3.30)
- FirstVisibleValue - lấy giá trị đầu tiên nhìn thấy trong mảng (AFL 3.40)
- GetChartBkColor - lấy giá trị màu RGB của nền biểu đồ (AFL 3.20)
- GetCursorMouseButtons - lấy trạng thái hiện tại của nút chuột (AFL 2.80)
- GetCursorXPosition - lấy vị trí
- GetCursorYPosition - lấy vị trí Y hiện tại của con trỏ chuột (AFL 2.80)

- HighestVisibleValue - lấy giá trị cao nhất trong vùng biểu đồ có thể nhìn thấy (AFL 3.30)
- LastVisibleValue - lấy giá trị có thể nhìn thấy cuối cùng của mảng (AFL 3.40)
- LowestVisibleValue - lấy giá trị thấp nhất trong vùng biểu đồ có thể nhìn thấy (AFL 3.30)
- MACD - hội tụ/phân kỳ trung bình động
- MDI - chỉ báo chuyển động theo hướng trừ (-DI) (AFL 1.3)
- MFI - chỉ số dòng tiền
- NVI - chỉ số khối lượng âm
- OBV - khối lượng cân bằng
- OscP - dao động giá
- OscV - dao động khối lượng
- PDI - chỉ báo chuyển động theo hướng cộng (AFL 1.3)
- PercentRank - tính toán xếp hạng phần trăm (AFL 3.40)
- PlotText - viết văn bản trên biểu đồ (AFL 2.80) - PlotVAPOverlayA - vẽ biểu đồ Volume-At-Price đa phân đoạn (AFL 3.20) - PVI - chỉ số khối lượng tích cực
- RequestMouseMoveRefresh - yêu cầu làm mới biểu đồ khi người dùng di chuyển con trỏ chuột (AFL 4.30)
- RequestTimedRefresh - buộc làm mới theo chu kì khung chỉ số (AFL 2.90)
- RMI - Chỉ số động lực tương đối (AFL 2.1)
- ROC - tỷ lệ thay đổi theo phần trăm
- RSI - chỉ số sức mạnh tương đối
- RWI - chỉ số đi dạo ngẫu nhiên
- RWIHi - chỉ số đi dạo ngẫu nhiên của điểm cao nhất
- RWILo - chỉ số đi dạo ngẫu nhiên của điểm thấp nhất
- SAR - điểm dừng và đảo ngược parabolic (AFL 1.3)
- SetBarFillColor - thiết lập màu chất lỏng cho biểu đồ bar/candlestick/cloud (AFL 3.1)
- SetChartBkColor - thiết lập màu nền cho một biểu đồ (AFL 2.80)
- SetChartBkGradientFill - kích hoạt chức năng tạo biểu đồ màu nền dạng gradient cho các chỉ số (AFL 2.90)
- SetGradientFill - thiết lập màu sắc cho dạng biểu đồ gradient fill (AFL 3.60) - Signal - dòng tín hiệu macd
- StochD - %D slow stochastic
- StochK - %K slow stochastic
- TRIX - giá triple exponential smoothed
- Ultimate - oscilator ultimate

Thông tin / Danh mục

- CategoryAddSymbol - thêm một biểu tượng vào một danh mục (AFL 2.5)
- CategoryCreate - thêm danh mục mới (như danh sách theo dõi) (AFL 3.70)
- CategoryFind - tìm kiếm danh mục theo tên (AFL 3.0)
- CategoryGetName - lấy tên của một danh mục (AFL 2.5)
- CategoryGetSymbols - lấy danh sách các biểu tượng được phân tách bằng dấu phẩy thuộc về danh mục đã cho (AFL 2.5)
- CategoryRemoveSymbol - rút một biểu tượng ra khỏi một danh mục (AFL 2.5)
- CategorySetName - đặt tên cho danh mục (nhóm, thị trường, danh sách theo dõi, ngành) (AFL 3.20)
- FullName - tên đầy đủ của biểu tượng (AFL 1.1)

- GetCategorySymbols - lấy danh sách các biểu tượng được phân tách bằng dấu phẩy thuộc về danh mục đã cho (AFL 2.4)
- GetDatabaseName - lấy tên thư mục của cơ sở dữ liệu hiện tại (AFL 2.3)
- GetFnData - lấy dữ liệu cơ bản (AFL 2.90)
- GetFnDataForeign - lấy dữ liệu cơ bản cho biểu tượng đã chỉ định (AFL 4.20)
- GicsID - lấy thông tin danh mục GICS (AFL 3.40) - GroupID - lấy ID/ tên nhóm (AFL 4.20)
- GroupID - lấy ID/tên nhóm (AFL 1.8)
- IcbID - lấy thông tin danh mục ICB (AFL 3.60)
- IndustryID - lấy ID/tên ngành (AFL 1.8)
- InGICS - kiểm tra tư cách thành viên GICS (AFL 3.40)
- InICB - kiểm tra tư cách thành viên ICB (AFL 3.60)
- InWatchList - kiểm tra tư cách thành viên danh sách theo dõi (theo số thứ tự)
- InWatchListName - kiểm tra tư cách thành viên danh sách theo dõi (theo tên) (AFL 3.0)
- IsContinuous - kiểm tra trạng thái cờ 'trích dẫn liên tục' (AFL 2.60)
- IsFavorite - kiểm tra xem ký hiệu hiện tại có thuộc mục yêu thích không (AFL 2.5)
- IsIndex - kiểm tra xem ký hiệu hiện tại có phải là chỉ mục không (AFL 2.5)
- MarketID - ID/tên thị trường (AFL 1.8)
- Tên - ticker symbol (AFL 1.1)
- SectorID - nhận ID / tên ngành (AFL 1.8)

Thấp nhất / Cao nhất

- HHV - giá trị cao nhất
- HHVBars - số thanh kể từ khi đạt giá trị cao nhất
- Highest - giá trị cao nhất
- HighestBars - số thanh từ khi đạt giá trị cao nhất
- HighestSince - giá trị cao nhất kể từ khi điều kiện được thỏa mãn (AFL 1.4)
- HighestSinceBars - số thanh từ khi đạt giá trị cao nhất kể từ khi điều kiện được thỏa mãn (AFL 1.4)
- LLV - giá trị thấp nhất
- LLVBars - số thanh từ khi đạt giá trị thấp nhất
- Lowest - giá trị thấp nhất
- LowestBars - số thanh từ khi đạt giá trị thấp nhất
- LowestSince - giá trị thấp nhất kể từ khi điều kiện được thỏa mãn (AFL 1.4)
- LowestSinceBars - số thanh từ khi đạt giá trị thấp nhất kể từ khi điều kiện được thỏa mãn (AFL 1.4)

Hàm Toán học

- abs - giá trị tuyệt đối
- acos - hàm arccosine
- AlmostEqual - so sánh không nhạy cảm với lỗi làm tròn (AFL 2.80)
- asin - hàm arcsin
- atan - arc tan
- atan2 - tính arctangent của y/x (AFL 2.90)
- ceil - giá trị ceil
- cos - cosine

- cosh - hàm cosine hyperbolic (AFL 2.80)
- erf - tính hàm lỗi Gauss (erf) (AFL 4.40)
- EXP - hàm mũ
- floor - giá trị sàn
- frac - phần phân số
- Int - phần nguyên
- inverf - hàm erf Gauss nghịch đảo (AFL 4.40)
- Kurtosis - tính độ nhọn (AFL 4.20)
- log - logarit tự nhiên
- log10 - logarit thập phân
- Max - giá trị lớn nhất của hai số / mảng
- Min - giá trị nhỏ nhất của hai số / mảng
- Prec - điều chỉnh số dấu thập phân của số dấu phẩy động
- Remap - ánh xạ lại một phạm vi thành một phạm vi khác (AFL 4.30)
- Round - làm tròn số thành số nguyên gần nhất
- SafeDivide - phép chia có bảo vệ chia cho không (AFL 4.40)
- sign - trả về dấu của số/mảng (AFL 2.50)
- sin - hàm sin
- sinh - hàm sin hyperbolic (AFL 2.80)
- Skewness - tính độ lệch (AFL 4.20)
- sqrt - căn bậc hai
- tan - hàm tanent (AFL 1.0)
- tanh - hàm tanent hyperbolic (AFL 2.80)

Các Hàm Khác

- #include - lệnh include tiền xử lý (AFL 2.2)
- #include_once - lệnh include tiền xử lý (once) (AFL 2.70)
- #pragma - đặt tùy chọn tiền xử lý AFL (AFL 2.4)
- ClipboardGet - truy xuất nội dung hiện tại của clipboard Windows (AFL 2.60)
- ClipboardSet - sao chép văn bản vào clipboard Windows (AFL 2.6)
- ColorHSB - chỉ định màu bằng Hue-Saturation-Brightness (AFL 2.80)
- ColorRGB - chỉ định màu bằng các thành phần Red-Green-Blue (AFL 2.80)
- CreateObject - tạo đối tượng COM (AFL 1.8)
- CreateStaticObject - tạo đối tượng COM tĩnh (AFL 1.8)
- EnableScript - bật công cụ tạo tập lệnh
- EnableTextOutput - cho phép bật hoặc tắt đầu ra văn bản (AFL 2.20)
- Error - hiển thị thông báo lỗi do người dùng xác định và dừng thực thi (AFL 3.7)
- FindIndex - tìm chỉ mục của mục mảng khớp với giá trị đã chỉ định (AFL 4.40)
- GetAsyncKeyState - truy vấn trạng thái hiện tại của các phím bàn phím (AFL 3.60)
- GetExtraData - lấy dữ liệu bổ sung từ nguồn dữ liệu bên ngoài (AFL 1.9)
- GetExtraDataForeign - lấy dữ liệu bổ sung từ nguồn dữ liệu bên ngoài cho ký hiệu đã chỉ định (AFL 4.20)
- GetFormulaPath - lấy đường dẫn tệp của công thức hiện tại (AFL 3.90)
- GetLastOSError - lấy văn bản lỗi hệ điều hành (Windows) gần đây nhất (AFL 4.30)
- GetObject - lấy xử lý đối tượng OLE đang chạy (AFL 4.40)

- GetPerformanceCounter - truy xuất giá trị hiện tại của bộ đếm hiệu suất độ phân giải cao (AFL 2.90)
- GetRTData - truy xuất các trường dữ liệu thời gian thực (AFL 2.60)
- GetRTDataForeign - truy xuất các trường dữ liệu thời gian thực (cho ký hiệu đã chỉ định) (AFL 2.80)
- GetScriptObject - truy xuất quyền truy cập vào tập lệnh Đối tượng COM (AFL 1.8)
- IsEmpty - kiểm tra giá trị rỗng (AFL 1.5)
- IsFinite - kiểm tra xem giá trị có vô hạn không (AFL 2.3)
- IsNan - kiểm tra NaN (không phải số) (AFL 2.3)
- IsNull - kiểm tra giá trị Null (rỗng) (AFL 2.3)
- IsTrue - kiểm tra giá trị true (không rỗng và không bằng 0) (AFL 1.5)
- NoteGet - truy xuất văn bản của ghi chú (AFL 2.6)
- NoteSet - đặt văn bản của ghi chú (AFL 2.6)
- NullCount - đếm các giá trị Null liên tiếp (AFL 3.90)
- Nz - Null (Null/Nan/Infinity) thành 0 (AFL 2.3)
- PlaySound - phát lại tệp .WAV đã chỉ định (AFL 3.40)
- PopupWindow - hiển thị cửa sổ bật lên (AFL 3.0)
- Prefs - truy xuất cài đặt tùy chọn (AFL 1.4)
- Reverse - đảo ngược thứ tự các phần tử trong mảng (AFL 3.90)
- SendEmail - gửi tin nhắn email (AFL 3.90)
- SetBarsRequired - đặt số lượng thanh trước và sau cần thiết để tập lệnh/DLL thực thi đúng (AFL 2.1)
- Sort - thực hiện sắp xếp nhanh mảng (AFL 3.90)
- SparseCompress - nén mảng thưa thớt (AFL 4.0)
- SparseExpand - mở rộng mảng đã nén thành mảng thưa thớt (AFL 4.0)
- SparseInterpolate - nội suy các giá trị giữa các điểm thưa thớt được đưa ra làm đầu vào (AFL 4.90)
- StaticVarAdd - phép cộng "nguyên tử" cho các biến tĩnh (AFL 4.10)
- StaticVarCompareExchange - hoạt động so sánh-trao đổi biến tĩnh liên kết nguyên tử (AFL 3.50)
- StaticVarCount - lấy tổng số biến tĩnh biến trong bộ nhớ (AFL 3.30)
- StaticVarGenerateRanks - tạo thứ hạng của nhiều ký hiệu và lưu trữ nó vào các biến tĩnh (AFL 3.70)
- StaticVarGet - lấy giá trị của biến tĩnh (AFL 2.60)
- StaticVarGetRankedSymbols - lấy danh sách các biểu tượng được xếp hạng từ biến tĩnh (AFL 3.70)
- StaticVarGetText - lấy giá trị của biến tĩnh dưới dạng chuỗi (AFL 2.60)
- StaticVarInfo - lấy thông tin về biến tĩnh (AFL 3.60)
- StaticVarRemove - xoá biến tĩnh (AFL 2.80)
- StaticVarSet - đặt giá trị cho biến tĩnh (AFL 2.60)
- StaticVarSetText - Đặt giá trị cho biến chuỗi tĩnh. (AFL 2.60)
- Status - lấy thông tin trạng thái AFL thời gian thực (AFL 1.65)
- Study - tham khảo nghiên cứu vẽ bằng tay (AFL 1.5)
- ThreadSleep - tạm dừng luồng cho số lượng mili giây chỉ định (AFL 3.50)
- VarGet - lấy giá trị của biến động (AFL 2.60)
- VarGetText - lấy giá trị văn bản của biến động (AFL 2.80)

- VarSet - đặt giá trị cho biến động (AFL 2.60)
- VarSetText - đặt biến động của loại chuỗi (AFL 2.80)
- Version - lấy thông tin phiên bản (AFL 1.9)
- _exit - kết thúc quá trình thực thi công thức AFL (AFL 4.40)
- _TRACE - in văn bản vào trình xem gỡ lỗi hệ thống (AFL 2.4)
- _TRACEF - in văn bản được định dạng vào trình xem gỡ lỗi hệ thống (AFL 4.0)

Trung bình di chuyển, Tổng

- AMA - trung bình di chuyển thích ứng (AFL 1.5)
- AMA2 - trung bình di chuyển thích ứng (AFL 1.5) - Cum - tổng lũy kế
- CumProd - tích lũy của tất cả các phần tử mảng (AFL 4.20)
- DEMA - trung bình di chuyển đôi mũ (AFL 2.0)
- EMA - trung bình di chuyển mũ
- FIR - Bộ lọc đáp ứng xung hữu hạn (AFL 3.40)
- HMA - Trung bình di chuyển Hull (AFL 3.40)
- IIR - bộ lọc đáp ứng xung vô hạn (AFL 4.0)
- MA - trung bình di chuyển đơn giản
- Prod - tổng lũy kế của mảng trên phạm vi được chỉ định (AFL 4.20)
- ProdSince - tổng lũy kế từ khi điều kiện được thỏa mãn (AFL 4.20)
- Sum - tổng dữ liệu trên số lượng thanh chỉ định
- SumSince - tổng của các phần tử mảng từ khi điều kiện đúng (AFL 4.10)
- TEMA - trung bình di chuyển mũ gấp ba (AFL 2.0)
- Wilders - làm mượt Wilder (AFL 1.4)
- WMA - trung bình động có trọng số (AFL 2.0)

Các hàm thống kê

- Correlation - tương quan (AFL 1.4)
- LinearReg - điểm cuối hồi quy tuyến tính (AFL 2.2)
- LinRegIntercept - (AFL 2.2)
- LinRegSlope - độ dốc hồi quy tuyến tính (AFL 1.4)
- Median - tính trung vị (phần tử ở giữa) (AFL 2.5)
- mtRandom - Trình tạo số ngẫu nhiên Mersene Twister (AFL 3.0)
- mtRandomA - Trình tạo số ngẫu nhiên Mersene Twister (phiên bản mảng) (AFL 3.0)
- NormDist - hàm phân phối chuẩn (AFL 4.20)
- Percentile - tính phần trăm (AFL 2.5)
- Random - số ngẫu nhiên (AFL 1.9)
- StdErr - lỗi chuẩn (AFL 1.4)
- StDev - độ lệch chuẩn (AFL 1.4)
- TSF - dự báo dãy thời gian (AFL 2.2)

Thao tác chuỗi

- Asc - lấy mã ASCII của ký tự (AFL 2.80)
- Chr - lấy chuỗi có mã ASCII cho trước (AFL 4.40)
- NumToStr - chuyển đổi số thành chuỗi (AFL 2.5)
- printf - In đầu ra đã định dạng vào cửa sổ đầu ra (AFL 2.5)

- StrCount - đếm số lần xuất hiện của chuỗi con trong chuỗi (AFL 3.20)
- StrExtract - trích xuất mục đã cho (chuỗi con) từ chuỗi có dấu phẩy phân tách (AFL 2.4)
- StrFind - tìm chuỗi con trong chuỗi (AFL 2.5)
- StrFormat - Viết đầu ra đã định dạng vào chuỗi (AFL 2.5)
- StrLeft - trích xuất phần trái nhất của chuỗi (AFL 2.0)
- StrLen - độ dài chuỗi (AFL 1.5)
- StrMatch - khớp mẫu/chuỗi dùng ký tự đại diện (AFL 4.0)
- StrMid - trích xuất một phần của chuỗi (AFL 2.0)
- StrReplace - thay thế chuỗi (AFL 2.90)
- StrRight - trích xuất phần phải nhất của chuỗi (AFL 2.0)
- StrSort - sắp xếp danh sách mục có dấu phẩy phân tách (AFL 3.90)
- StrToDateTime - chuyển đổi chuỗi thành ngày giờ (AFL 2.80)
- StrToLower - chuyển đổi thành chữ thường (AFL 2.80)
- StrToNum - chuyển đổi chuỗi thành số (AFL 2.5)
- StrToUpper - chuyển đổi thành chữ in hoa (AFL 2.80)
- StrTrim - cắt bỏ khoảng trắng từ chuỗi (AFL 3.90)

Bộ công cụ Hệ thống Giao dịch

- AlertIf - kích hoạt cảnh báo (AFL 2.1)
- ApplyStop - áp dụng dừng built-in (AFL 1.7)
- BarsSince - thanh từ BarsSinceCompare - thanh từ so sánh giữa giá trị mảng quá khứ và hiện tại là đúng (AFL 4.40)
- Cross - kiểm tra crossover
- EnableRotationalTrading - Chuyển đổi chế độ trading xoay vòng của backtester (AFL 2.5)
- Equity - tính toán đường công bằng cho một biểu tượng (AFL 2.0)
- ExRem - loại bỏ tín hiệu dư thừa (AFL 1.5)
- ExRemSpan - loại bỏ tín hiệu dư thừa trải rộng qua số lượng thanh cho trước (AFL 2.0)
- Flip - (AFL 1.5)
- GetBacktesterObject - lấy quyền truy cập vào object backtester (AFL 2.60)
- GetOption - nhận giá trị của tùy chọn trong cài đặt phân tích tự động (AFL 2.60)
- GetTradingInterface - truy xuất đối tượng tự động hóa để giao dịch tự đ
- Hold - giữ tín hiệu cảnh báo
- IIf - hàm IF tức thời
- LastValue - giá trị cuối cùng của mảng
- Optimize - định nghĩa biến tối ưu hóa (AFL 1.7)
- OptimizerSetEngine - chọn công cụ tối ưu hóa bên ngoài (AFL 3.20)
- OptimizerSetOption - đặt giá trị của tham số công cụ tối ưu hóa bên ngoài (AFL 3.20)
- Ref - tham chiếu các giá trị trong quá khứ/tương lai của mảng
- SetBacktestMode - Đặt chế độ làm việc của backtester (AFL 3.0)
- SetCustomBacktestProc - định nghĩa tệp công thức thủ tục backtest tùy chỉnh (AFL 2.70)
- SetFormulaName - đặt tên của công thức (AFL 2.5)
- SetOption - thiết lập các tùy chọn trong cài đặt phân tích tự động (AFL 2.3)
- SetPositionSize - thiết lập kích thước giao dịch (AFL 2.70)
- SetStopPrecedence - thiết lập thứ tự ưu tiên cho dừng tích hợp (AFL 4.0)

- SetTradeDelays - cho phép kiểm soát thời gian chậm giao dịch được áp dụng bởi bộ kiểm tra lỗi (AFL 2.1)
- ValueWhen - lấy giá trị của mảng khi điều kiện được đáp ứng (AFL 1.1)

Khám phá / Chỉ số

- AddColumn - thêm cột khám phá số (AFL 1.8)
- AddMultiTextColumn - thêm cột văn bản khám phá dựa trên mảng (AFL 4.20)
- AddRankColumn - thêm cột xếp hạng theo thứ tự hiện tại được đặt bởi SetSortColumns (AFL 5.70)
- AddRow - thêm hàng văn bản gốc vào khám phá (AFL 4.0)
- AddSummaryRows - thêm hàng(t) tóm tắt vào kết quả khám phá (AFL 3.2)
- AddTextColumn - thêm cột văn bản khám phá (AFL 1.8)
- EncodeColor - mã hóa màu cho tiêu đề chỉ số (AFL 2.2)
- GetChartID - lấy ID biểu đồ hiện tại (AFL 2.3)
- GetPriceStyle - lấy kiểu biểu đồ giá hiện tại (AFL 2.70)
- LineArray - tạo mảng dự đồ thị xu hướng (AFL 2.5)
- Param - thêm tham số số có thể xác định bởi người dùng (AFL 2.3)
- ParamColor - thêm tham số màu có thể xác định bởi người dùng (AFL 2.3)
- ParamDate - thêm tham số ngày do người dùng định nghĩa (AFL 2.60)
- ParamField - tạo tham số trường giá (AFL 2.70)
- ParamList - tạo tham số bao gồm danh sách các lựa chọn (AFL 2.70)
- ParamStr - thêm tham số chuỗi do người dùng định nghĩa (AFL 2.3)
- ParamStyle - chọn kiểu áp dụng cho biểu đồ (AFL 2.70)
- ParamTime - thêm tham số thời gian do người dùng định nghĩa (AFL 2.60)
- ParamToggle - tạo tham số Có/Không (AFL 2.70)
- ParamTrigger - tạo trình kích hoạt (nút) trong hộp thoại tham số (AFL 2.70)
- Plot - biểu đồ chỉ báo vẽ (AFL 1.8)
- PlotGrid - vẽ đường lưới ngang (AFL 2.3)
- PlotOHLC - vẽ biểu đồ OHLC tùy chỉnh (AFL 2.2)
- PlotShapes - vẽ mũi tên và các hình dạng khác (AFL 2.3)
- PlotVAPOverlay - vẽ biểu đồ chồng Volume-At-Price (AFL 2.4)
- SelectedValue - truy xuất giá trị của mảng tại thời điểm ngày/giờ hiện được chọn (AFL 2.1)
- SetChartOptions - đặt/xóa/ghi đè mặc định cho các tùy chọn ngăn biểu đồ (AFL 2.70)
- SetSortColumns - đặt các cột sẽ được sử dụng để sắp xếp trong cửa sổ AA (AFL 2.90)
- TrimResultRows - cắt danh sách kết quả Phân tích thành số hàng đã chỉ định (AFL 4.90)
- WriteIf - đầu ra văn bản có điều kiện bình luận
- WriteVal - chuyển đổi số thành chuỗi
- XYChartAddPoint - thêm điểm vào biểu đồ phân tán khám phá (XY) (AFL 3.60)
- XYChartSetAxis - đặt tên trục X và Y trong biểu đồ phân tán khám phá (AFL 3.60)
- _DEFAULT_NAME - truy xuất tên mặc định của biểu đồ (AFL 2.70)
- _N - không có đầu ra văn bản (AFL 2.1)
- _PARAM_VALUES - lấy chuỗi giá trị tham số (AFL 2.70)
- _SECTION_BEGIN - đánh dấu bắt đầu phần (AFL 2.70)
- _SECTION_END - đánh dấu kết thúc phần (AFL 2.70)

- _SECTION_NAME - lấy tên phần hiện tại (AFL 2.70)

Hàm đọc/ghi file

- fclose - đóng một file (AFL 2.5)
- fdelete - xóa tệp (AFL 2.70)
- fdir - liệt kê nội dung thư mục (AFL 3.70)
- feof - kiểm tra cuối tệp (AFL 2.5)
- fgetcwd - lấy thư mục làm việc hiện tại (AFL 4.10)
- fgets - lấy một chuỗi từ tệp (AFL 2.5)
- fgetstatus - truy xuất trạng thái/tính năng của tệp (AFL 2.90)
- fmkdir - tạo (làm) một thư mục (AFL 2.70)
- fopen - mở một tệp (AFL 2.5)
- fputs - viết một chuỗi vào tệp (AFL 2.5)
- frmdir - xóa thư mục (AFL 2.70)
- InternetClose - đóng Internet file handle (AFL 4.20)
- InternetGetStatusCode - trả về mã trạng thái HTTP của lần gọi Internet cuối cùng (AFL 4.40)
- InternetOpenURL - mở tài nguyên web Internet (URL) (AFL 4.20)
- InternetPostRequest - gửi yêu cầu HTTP Post đến tài nguyên web Internet (URL) (AFL 4.30)
- InternetReadString - đọc một chuỗi từ tài nguyên Internet (AFL 4.20)
- InternetSetAgent - đặt chuỗi agent cho chức năng Internet (AFL 4.30)
- InternetSetHeaders - đặt tiêu đề HTTP tùy chỉnh cho các yêu cầu web tiếp theo (AFL 4.40)
- InternetSetOption - đặt tùy chọn HTTP cho phiên Internet (AFL 4.40)

Chức năng Gui

- GuiButton - tạo nút kiểm soát trên biểu đồ (AFL 4.30)
- GuiCheckBox - tạo hộp kiểm trên biểu đồ (AFL 4.30)
- GuiDateTime - tạo bộ chọn ngày-giờ trên biểu đồ (AFL 4.30)
- GuiEdit - tạo kiểm soát chỉnh sửa trên biểu đồ (AFL 4.30)
- GuiEnable - kích hoạt hoặc vô hiệu hóa kiểm soát trên biểu đồ (AFL 4.30)
- GuiGetCheck - lấy trạng thái đã kiểm tra của điều khiển (AFL 4.30)
- GuiGetEvent - lấy sự kiện GUI (AFL 4.30)
- GuiGetText - lấy văn bản từ điều khiển trên biểu đồ (AFL 4.30)
- GuiGetValue - lấy giá trị số từ điều khiển trên biểu đồ (AFL 4.30)
- GuiRadio - tạo nút radio trên biểu đồ (AFL 4.30)
- GuiSendKeyEvents - yêu cầu thông báo GUI cho các phím đã chỉ định (AFL 4.40)
- GuiSetCheck - đặt trạng thái đã kiểm tra của điều khiển trên biểu đồ (AFL 4.30)
- GuiSetColors - đặt màu sắc cho điều khiển GUI (AFL 2.30)
- GuiSetFont - đặt font chữ cho điều khiển trên biểu đồ (AFL 4.30)
- GuiSetRange - đặt phạm vi điều khiển thanh trượt (AFL 4.30)
- GuiSetText - đặt giá trị văn bản của điều khiển trên biểu đồ (AFL 4.30)
- GuiSetValue - đặt giá trị số của điều khiển trên biểu đồ (AFL 4.30)
- GuiSetVisible - hiển thị hoặc ẩn điều khiển trên biểu đồ (AFL 4.30)
- GuiSlider - tạo điều khiển thanh trượt trên biểu đồ (AFL 4.30)

Đồ họa cấp thấp

- GfxArc - vẽ cung tròn (AFL 3.0)
- GfxChord - vẽ dây cung (AFL 3.0)
- GfxCircle - vẽ hình tròn (AFL 3.0)
- GfxDrawImage - vẽ ảnh bitmap (AFL 4.20)
- GfxDrawText - vẽ văn bản (cắt thành hình chữ nhật) (AFL 3.0)
- GfxEllipse - vẽ hình elip (AFL 3.0)
- GfxFillSolidRect - tô hình chữ nhật bằng màu đặc (AFL 4.10)
- GfxGetTextWidth - lấy chiều rộng pixel của văn bản (AFL 2.80)
- GfxGradientRect - vẽ hình chữ nhật với tô màu chuyển sắc (AFL 3.0)
- GfxLineTo - Vẽ một đường tới một điểm cụ thể (AFL 3.0).
- GfxMoveTo - Di chuyển con trỏ đồ họa tới vị trí mới (AFL 3.0).
- GfxPie - Vẽ một hình bánh (AFL 3.0).
- GfxPolygon - Vẽ một đa giác (AFL 3.0).
- GfxPolyline - Vẽ một đường polyline (AFL 3.0).
- GfxRectangle - Vẽ một hình chữ nhật (AFL 3.0).
- GfxRoundRect - Vẽ hình chữ nhật có góc cạnh bo tròn (AFL 3.0).
- GfxSelectFont - Tạo/mở chọn font đồ họa (AFL 3.0).
- GfxSelectHatchBrush - Chọn chổi cọ kiểu Hatch (AFL 4.0).
- GfxSelectPen - Tạo/mở chọn bút đồ họa (AFL 3.0).
- GfxSelectSolidBrush - Tạo/mở chọn chổi cọ đồ họa (AFL 3.0).
- GfxSelectStockObject - Chọn một đối tượng đồ họa có sẵn (AFL 4.00).
- GfxSetBkColor - Đặt màu nền đồ họa (AFL 3.0).
- GfxSetBkMode - Đặt chế độ nền đồ họa (AFL 3.0).
- GfxSetCoordsMode - Thiết lập chế độ toạ độ đồ họa cấp thấp (AFL 2.80).
- GfxSetOverlayMode - Thiết lập chế độ overlay đồ họa mức thấp (AFL 3.0).
- GfxSetPixel - Đặt pixel ở vị trí cụ thể với màu cụ thể (AFL 3.0).
- GfxSetTextAlign - Đặt căn lề văn bản (AFL 3.0).
- GfxSetTextColor - Đặt màu chữ của đồ họa (AFL 3.0).
- GfxSetZOrder - Đặt tầng lớp Z-order đồ họa mức thấp hiện tại (AFL 2.80).
- GfxTextOut - Ghi văn bản tại vị trí chỉ định (AFL 3.0).

Hàm bản đồ

- MapCreate - Tạo một đối tượng bản đồ/danh sách (giữ các cặp key-value) (AFL 4.90).

Hàm ma trận

- Matrix - Tạo một ma trận mới (AFL 4.0).
- MxCopy - Sao chép khối hình chữ nhật từ một ma trận sang một ma trận khác (AFL 4.40).
- MxDet - Tính toán định thức của ma trận (AFL 4.10).
- MxFromString - Tạo một ma trận mới từ chuỗi (AFL 4.10).
- MxGetBlock - Lấy khối hình chữ nhật của các mục từ ma trận (AFL 4.10).
- MxGetSize - Lấy kích thước của ma trận (AFL 4.0).
- MxIdentity - Tạo một ma trận đơn vị (AFL 4.0).
- MxInverse - Tính toán ma trận nghịch đảo (AFL 4.10).

- MxSetBlock - thiết lập giá trị trong khối hình chữ nhật của các ô ma trận (AFL 4.10)
- MxSolve - giải hệ phương trình tuyến tính A @ X = B (AFL 4.10)
- MxSort - sắp xếp ma trận (AFL 4.10)
- MxSortRows - sắp xếp các hàng của ma trận (AFL 4.10)
- MxSum - tính tổng lớn của ma trận (AFL 4.20)
- MxToString - chuyển đổi ma trận thành chuỗi (AFL 4.10)
- MxTranspose - tạo chuyển vị của ma trận đầu vào (AFL 4.0)
- PriceVolDistribution - hàm phân phối mục đích chung (AFL 4.20)

Tham chiếu dữ liệu ký hiệu khác

- Foreign - truy cập dữ liệu chứng khoán nước ngoài (AFL 1.5)
- GetBaseIndex - truy xuất ký hiệu của chỉ số cơ sở sức mạnh tương đối (AFL 2.1)
- PlotForeign - vẽ dữ liệu chứng khoán nước ngoài (AFL 2.2)
- RelStrength - sức mạnh tương đối so sánh (AFL 1.3)
- RestorePriceArrays - khôi phục mảng giá về ký hiệu gốc (AFL 2.5)
- SetForeign - thay thế mảng giá hiện tại bằng mảng giá của bảo mật nước ngoài (AFL 2.5)

Các chức năng Văn bản-sang-Lời nói (Text-To-Speech Functions)

- Say - phát biểu văn bản được cung cấp (AFL 2.90)
- VoiceCount - lấy số lượng giọng SAPI (AFL 4.20)
- VoiceSelect - chọn giọng SAPI (AFL 4.20)
- VoiceSetRate - đặt tốc độ lời nói (AFL 4.30)
- VoiceSetVolume - đặt âm lượng của lời nói (AFL 4.30)
- VoiceWaitUntilDone - chờ đến khi giọng TTS hoàn tất việc nói (AFL 4.30)

Các hàm Khung thời gian

- TimeFrameCompress - nén mảng đơn vào khung thời gian đã cho (AFL 2.5)
- TimeFrameExpand - mở rộng khung thời gian nén mảng (AFL 2.5)
- TimeFrameGetPrice - lấy giá trị O, H, L, C, V từ khung thời gian khác (AFL 2.5)
- TimeFrameMode - chuyển chế độ nén khung thời gian (AFL 2.80)
- TimeFrameRestore - phục hồi mảng giá thành khung thời gian gốc (AFL 2.5)
- TimeFrameSet - chuyển mảng giá sang khung thời gian khác (AFL 2.5)

Lệnh #include - Lệnh gồm nhiều chức năng

Cú pháp

#include

Đầu ra

Không trả về gì

Chức năng

Đưa các tệp AFL bên ngoài vào công thức của bạn. Vui lòng lưu ý:

- Ghi chú 1: câu lệnh include cần dấu gạch chéo đơn trong đường dẫn (điều này hoàn toàn trái ngược với việc phân tích chuỗi AFL bình thường)
- Ghi chú 2: việc sử dụng lệnh #include có thể làm chậm thực hiện công thức, ngay cả khi AmiBroker cố gắng chỉ đưa vào một lần và lưu trữ văn bản đã qua xử lý
- Ghi chú 3: hiện tại không có thông báo lỗi nếu #include thất bại và mã này đang trong giai đoạn thử nghiệm
- Ghi chú 4: việc lồng các lệnh #include hiện đã được hỗ trợ (phiên bản 5.10 trở lên)
- Ghi chú 5: Theo mặc định, các tệp được #include đều được lưu trữ bởi AmiBroker. Để tắt lưu trữ, sử dụng #pragma nocache trước bất kỳ lệnh #include nào

#include giờ đây chấp nhận cách mới để chỉ định tên tệp để đưa vào: #include (chú ý đến dấu ngoặc đơn <> thay vì "") Nếu bạn chỉ định tên tệp theo cách này, AmiBroker sẽ tìm kiếm tệp trong "đường dẫn bao gồm tiêu chuẩn" có thể định rõ thông qua cài đặt mới trong Tools->Preferences->AFL

Xem thêm

Hàm #pragma()

Ví dụ

Ví dụ	#include "C:\Program Files\AmiBroker\AFL\common.afl"

Tham khảo

Hàm #include được sử dụng trong các công thức sau trong thư viện AFL trực tuyến:

Thông tin thêm

Xem phiên bản cập nhật/mở rộng trực tuyến.

#include_once - lệnh bao gồm (một lần) của trình tiền xử lý

CÚ PHÁP #include "đường dẫn tệp công thức"

KHÔNG TRẢ VỀ gì

CHỨC NĂNG Bao gồm các tệp AFL bên ngoài vào công thức của bạn. Tương tự như #include nhưng #include_once chỉ thực hiện việc bao gồm một lần cho mỗi công thức. Vì vậy, nếu một công thức đơn có nhiều lệnh #include_once cho cùng một tệp (ví dụ do kéo và thả phủ lên) thì nó ngăn chặn các lỗi cú pháp có thể xảy ra do việc định nghĩa lại các hàm trong tệp được bao gồm. Thông tin chi tiết hơn có thể tìm thấy trong tài liệu lệnh #include.

VÍ DỤ #include_once "myfile.afl"

XEM THÊM Hàm #include()

Chức năng #include_once được sử dụng trong các công thức sau trong thư viện trực tuyến AFL.

Thông tin thêm: Xem phiên bản cập nhật/mở rộng trực tuyến.

#pragma - thiết lập tùy chọn tiền xử lý AFL

CÚ PHÁP #pragma optionname TRẢ VỀ KHÔNG CÓ GÌ

FUNCTION Thiết lập nhiều tùy chọn tiền xử lý AFL khác nhau. Tiền xử lý là một phần của công cụ AFL xử lý các công thức TRƯỚC KHI chúng được thực thi. #pragma cho phép thay đổi hành vi của tiền xử lý. Tiền xử lý chịu trách nhiệm đưa các tệp bên ngoài vào thông qua lệnh #include. Bộ tiền xử lý #pragma cũng có thể ảnh hưởng đến hành vi của cửa sổ Phân tích và các tiền tố biến tĩnh đã khai báo

Hiện tại #pragma hỗ trợ các lệnh sau:

• #pragma nocache - khiến các tệp #include không được lưu vào bộ nhớ đệm nên chúng được đọc lại sau mỗi lần thực thi

#pragma nocache phải được đặt trước bất kỳ lệnh #include nào Lưu ý: giữa '#pragma' và 'nocache' phải có CHÍNH XÁC MỘT khoảng trắng

Lưu ý 2: việc tắt bộ nhớ đệm có thể làm chậm quá trình thực thi công thức (đặc biệt là trong các chỉ báo) !!!

• #pragma maxthreads N - khiến cửa sổ Phân tích giới hạn số luồng song song

là N

• #pragma enable_static_decl "prefix_string" - cho phép khai báo biến tĩnh với tiền tố được chỉ định

• #pragma sequence(list_of_actions) - xác định trình tự các hành động được thực hiện tự động khi bạn nhấn nút "Chạy trình tự" trong cửa sổ Phân tích

VÍ DỤ #pragma nocache

#include "myfile.afl"

XEM THÊM hàm #include()

Tham khảo:

Hàm #pragma được sử dụng trong các công thức sau trong thư viện trực tuyến AFL:

Thông tin thêm:

Xem phiên bản cập nhật/mở rộng trực tuyến.

DEFAULT NAME- lấy tên mặc định của biểu đồ

SYNTAX _DEFAULT_ NAME() TRẢ VỀ CHUỖI
HÀM Hàm này trả về tên mặc định của biểu đồ trong phần kéo-thả. Tên mặc định bao gồm tên phần và danh sách các giá trị của tham số số được phân tách bằng dấu phẩy được xác định trong phần đã cho.
VÍ DỤ _SECTION_BEGIN("MA1");
P = ParamField("Trường giá");
Periods = Param("Kỳ", 15, 2, 200, 1, 10);
Plot(MA(P, Periods), _DEFAULT_NAME(), ParamColor("Color", colorCycle), ParamStyle("Style"));
_SECTION_END();

DEFAULT NAM sẽ đánh giá thành chuỗi "MA1(Close,15)".
XEM THÊM hàm _SECTION_BEGIN(), hàm _SECTION_NAME(), hàm _SECTION_END()
Tài liệu tham khảo:

Hàm _DEFAULT_NAME được sử dụng trong các công thức sau trong thư viện trực tuyến AFL:

• Hệ thống MA nâng cao
• Ví dụ AFL
• Ví dụ AFL - Nâng cao
• Liên kết COM từ AFL sang Python
• AO+Momentum
• Giao dịch tự động từng bước
• Giao dịch tự động bằng cách sử dụng Exploration
• Luồng cơ bản của AutoTrader - cập nhật ngày 15 tháng 4 năm 2009
• Luồng cơ bản của AutoTrader - cập nhật ngày 18 tháng 11 năm 2008
• Giao cắt giá trung bình
• babaloo chapora
• Giao dịch nút bằng giao diện giao dịch tự động AB
• Quét PullBack của Dave Landry
• DEBAJ
• Đường trung bình động dịch chuyển 3x DiNapolis
• DPO có tô bóng
• Màn hình ba Elder Hệ thống giao dịch
• Nhiệt kế thị trường Elder
• Bộ dao động sóng Elliott
• Đường trung bình động Fibonacci
• Máy vẽ mức Gann
• Heikin-Ashi (Koma-Ashi) với Đường trung bình động
• Herman
• Trình lấp đầy IB
• Hệ thống phá vỡ xu hướng Fibonacii trong ngày
• HEIKIN ASHI trong ngày mới
• Hệ thống phá vỡ xu hướng trong ngày
• Trình giao dịch lệnh khung thủ công
• mitalpradip
• Mndahoo ADX
• Modified-DVI
• Nadaraya-Watson Envelope
• RSI dự đoán/mượt mà do mạng nơ-ron cung cấp V2
• Noor_Doodie

• ParabXO
• prakash
• PVT Trend Decider
• Chỉ số đi bộ ngẫu nhiên
• RI - Hệ thống giao dịch tự động
• Chỉ báo RSI với Vùng trên và dưới Thanh
• RSI styleClipMinMax
• shailu lunia
• Simple Momentum
• Đường trung bình động tam giác mới
• Dòng tiền Twiggs
• Ultimate plus
• Tệp word tô sáng trình soạn thảo UltraEdit
• Hệ thống trong ngày Vikram's Floor Pivot
• Hệ thống giao dịch trực quan của Turtle
• Bộ lọc ZigZag được viết lại từ đầu trong AFL
• ZigZag Hi Lo Barcount

Thông tin thêm:

Xem phiên bản cập nhật/mở rộng trực tuyến.

_DT- chuyển đổi chuỗi thành datetime

CÚ PHÁP _DT("chuỗi ngày"); TRẢ VỀ SỐ
HÀM Chuyển đổi chuỗi biểu diễn giá trị ngày/giờ thành số DateTime tương ứng (sau đó có thể so sánh với đầu ra của hàm DateTime() chẳng hạn).

Hàm này là từ đồng nghĩa/phím tắt cho hàm StrToDateTime.
VÍ DỤ
XEM THÊM Hàm StrToDateTime()
Tham khảo:

Hàm _DT được sử dụng trong các công thức sau trong thư viện trực tuyến AFL:

Thông tin thêm:

Xem phiên bản cập nhật/mở rộng trực tuyến.

_exit - chấm dứt việc thực thi công thức AFL

CÚ PHÁP _exit() KHÔNG TRẢ VỀ
HÀM Hàm này sẽ kết thúc việc thực thi AFL một cách bình thường tại thời điểm gọi. Mã sau lệnh gọi _exit() sẽ không được thực thi.

Nó hữu ích nếu bạn muốn chấm dứt quá trình xử lý AFL nhanh hơn bình thường trong một số tình huống nhất định
VÍ DỤ if(Status("action" == actionScan)
{
// nếu chúng ta đang chạy scan, chúng ta chỉ cần tạo composite AddToComposite(Volume, "~~~TotalVol", "X");
// và kết thúc quá trình thực thi nhanh chóng
_exit();
}

// phần còn lại của mã sẽ chỉ được thực thi nếu KHÔNG chạy SCAN
XEM THÊM
Tài liệu tham khảo:

Hàm _exit được sử dụng trong các công thức sau trong thư viện trực tuyến AFL:

• Thêm các tham số SL/TGT khác vào bất kỳ chiến lược nào
• TSP của Ed Seykota: Hệ thống giao cắt EMA
• TSP của Ed Seykota: Hỗ trợ và kháng cự
• MFE và MAE và vẽ giao dịch dưới dạng chỉ báo

Thông tin thêm:

Xem phiên bản cập nhật/mở rộng trực tuyến.

_N- không có đầu ra văn bản

CÚ PHÁP _N(chuỗi) KHÔNG TRẢ VỀ
CHỨC NĂNG Bảo vệ khỏi việc in chuỗi vào cửa sổ đầu ra chú thích
VÍ DỤ _N(ticker = name()); // nhờ hàm _N, ký hiệu ticker không được in
XEM THÊM hàm ENABLETEXTOUTPUT()
Tham khảo:

Hàm _N được sử dụng trong các công thức sau trong thư viện trực tuyến AFL:

• Pivots And Prices And Swing Volume
• 'R' Channel
• 10-20 Indicator
• 2 Timeframes Candlestick Bar Chart
• 3 cách sử dụng RMI trong một tập lệnh
• 30 Week Hi Indicator - Hiển thị
• 3TF Candlestick Bar Chart
• 4% Model - Xác định hướng thị trường chứng khoán
• 52 Week New High-New Low Index
• Một hệ thống kiểm soát sửa đổi AFL đơn giản
• Abhimanyu
• AccuTrack
• Adaptive Laguerre Filter, từ John Ehlers
• Tự động thêm Nifty 50 IB Equity Symbol
• Thêm SL/TGT các tham số khác vào bất kỳ chiến lược nào
• Advanced MA hệ thống
• Tìm kiếm và Tìm kiếm Nâng cao
• Đường xu hướng Nâng cao với S & R
• Hiển thị biểu đồ giá NRx tư vấn.
• ADXbuy
• ADXVMA
• Ví dụ AFL
• Ví dụ AFL - Nâng cao
• Liên kết COM từ AFL sang Python
• AFL_Glossary_1
• AFL_Glossary_Converter
• Đầu ra cảnh báo dưới dạng Truy xuất nhanh
• ALJEHANI
• AllinOneAlerts - Mô-đun
• Chỉ báo Alpha và Beta và R_Squared
• Alphatrend
• Hàm ZIG thay thế
• Hàm loại ZIG thay thế, nhiều TF
• Andrews Pitchfork
• Andrews PitchforkV3.3
• Nền động
• Nền động 1.1
• AO+Momentum
• ROC của Appel hoặc Mô hình thời gian động ba
• Chỉ báo Aroon
• Aroon The Advisor
• AR_Prediction.afl
• Nghiên cứu ATR
• Tự động mở rộng và thoái lui Fib

• Tự động giao dịch từng bước
• Khung tối ưu hóa tự động
• Kênh xu hướng tuyến tính tự động
• Kênh xu hướng tuyến tính tự động 2
• Kênh xu hướng tự động
• Đường xu hướng tự động
• Đường xu hướng tự động sử dụng các mẫu fractal
• Đường xu hướng tự động sử dụng nhiều khung thời gian
• Tự động giao dịch bằng cách khám phá
• Luồng cơ bản của AutoTrader - cập nhật ngày 15 tháng 4 năm 2009
• Luồng cơ bản của AutoTrader - cập nhật ngày 18 tháng 11 năm 2008
• Khám phá biến động giá đô la trung bình
• Giao cắt giá trung bình
• B-Xtrender
• babaloo chapora
• Dữ liệu sao lưu của khoảng thời gian 1 phút
• Cắt giảm tích xấu trên cơ sở dữ liệu 5 giây
• Biểu đồ danh sách theo dõi hiệu suất tương đối cơ sở V2
• Hệ thống giao dịch giỏ hàng T101
• BBAreacolor&TGLCROSSNEW
• BEANS-Tóm tắt về các khoản nắm giữ
• Bảng thông tin giá mua so với giá bán
• Bộ chuyển đổi nhị phân sang thập phân
• Giá quyền chọn Black Scholes
• HaDiffCO của Bman
• Chỉ báo xu hướng thị trường trung hạn BMTRIX
• Chuẩn hóa dải Bollinger
• Bóp dải Bollinger
• Độ rộng dải Bollinger
• Khám phá đáy Fisher
• Bull Fear / Bear Fear
• Chỉ số phần trăm tăng giá
• Chỉ số phần trăm tăng giá 2 tệp kết hợp
• Chỉ số phần trăm tăng giá 2004
• Giao dịch bằng nút sử dụng giao diện giao dịch tự động AB
• Tính toán các hợp chất cho mã chứng khoán trong tệp danh sách
• Caleb Lawrence
• Mẫu cốc và tay cầm CAMSLIM AFL
• Nhận dạng nến
• Chức năng mẫu nến
• Phân tích nến
• Bản demo nến
• Biểu đồ nến cho Khối lượng/RSI/OBV
• Bình luận về CandleStick--Cần trợ giúp
• Bình luận về Candlestick
• Bình luận về Candlestick đã sửa đổi
• Bình luận về Candlestick đã sửa đổi
• CCI 14 DrBobStyle
• CCI 50 DrBob Style
• Chỉ báo phân kỳ CCI(20)
• Chandelier Exit
• Chandelier Exit
• trung bình động thay đổi theo chu kỳ
• chỉ báo kênh
• Kênh/S&R và đường xu hướng

• Thu phóng biểu đồ
• CoinToss ver 1
• Cole
• Kết hợp màu sắc
• Hiển thị màu sắc.afl
• Thanh giá màu - Hệ thống xung lực
• Thanh giá màu với các thay đổi về biểu đồ MACD
• Giá đầy màu sắc
• com-out
• Chỉ số lựa chọn hàng hóa (CSI)
• Bảng điều khiển tích hợp đồng thời và ADF
• Tắc nghẽn phát hiện
• Connors TPS
• Chuyển đổi hợp đồng liên tục
• Biểu đồ Histogram Coppock
• Tín hiệu giao dịch Coppock trên Biểu đồ giá
• Tín hiệu giao dịch Coppock v1.1
• correlerror
• BÁO CÁO COT
• Đếm mã chứng khoán trong Danh sách theo dõi
• Tạo danh sách các hàm trong chương trình của bạn
• crMathLib
• CVR--bộ lọc nghiêm trọng
• Sóng sin Hilbert Cybernertic
• Công cụ tô sáng chu kỳ
• Công cụ tô sáng chu kỳ (tự động phù hợp nhất)
• Chu kỳ
• Cao thấp hàng ngày trong Dự báo
• Độ thâm nhập danningham
• Darvas Amibroker
• Nghiên cứu Darvas Johndeo
• Darvas Wisestocktrader
• DateNum_DateStr
• Date_To_Num(), Time_To_Num()
• Quét lùi Dave Landry
• Thanh ngày Không
• DEBAJ
• Chỉ số cầu
• Đường cong vốn chủ sở hữu chi tiết
• Chỉ báo kỹ thuật số
• Đường trung bình động dịch chuyển DiNapolis 3x
• Chỉ báo phân kỳ
• Phân kỳ
• Chỉ số chênh lệch DMI
• Giai đoạn chu kỳ chi phối
• Hệ thống siêu xu hướng kép
• Phát hiện đỉnh kép
• Chỉ số động lượng động
• TSP của Ed Seykota: Hệ thống giao cắt EMA
• TSP của Ed Seykota: Hỗ trợ và kháng cự
• Chỉ báo biến áp Hilbert Ehlers
• Chỉ báo phản xạ Ehlers
• Chỉ báo xu hướng Ehlers
• Biểu đồ giá ekeko
• EKEKO SAR-MF

• Chỉ báo xung lực Elder V2
• Elder Ray - Bull Bear
• Bộ dao động Elder Ray với MA
• Vùng an toàn Elder Long + Short
• Hệ thống giao dịch Elder Triple Screen
• Nhiệt kế thị trường Elder
• Nhãn hiệu thủ công sóng Elliott
• Bộ dao động sóng Elliott
• Giao cắt EMA
• Giao dịch cuối năm
• Hệ thống Envelope
• Đánh giá các mẫu nến trong hệ thống giao dịch
• Chỉ báo sóng sin thậm chí còn tốt hơn
• Ngày/ngày hết hạn - Thứ năm cuối cùng của tháng
• Xuất tất cả dữ liệu hàng ngày sang TXT với định dạng nhập MS
• Xuất EOD hoặc Intraday sang tệp .csv
• Xuất dữ liệu Intraday
• Trích xuất các dòng mã cụ thể từ chương trình của bạn
• Biểu đồ dao động KAGI được làm mới nhanh (Biến động giá)
• FastStochK FullStochK-D
• Fib Fan dựa trên ZZ
• Tính toán Fibonacci & Kháng cự tốc độ
• Các đường thoái lui bên trong và bên ngoài Fibonacci
• Đường trung bình động Fibonacci
• FirstBarIndex(), LastBarIndex()
• Bộ dao động Fisher
• Theo dõi người dẫn đầu
• Để thiết lập giao dịch tự động
• Chỉ số lực
• Fre
• FTWHMS - CHIẾN LƯỢC ĐỘNG LƯỢNG CAO NĂM MƯƠI HAI TUẦN
• Công cụ sàng lọc quỹ
• Dự báo MA tương lai
• Vẽ biểu đồ tương lai về thời gian và Giá
• Vẽ đồ thị tương lai của thời gian và giá
• Chỉ báo góc hồi quy tuyến tính Gabriel
• Chỉ báo và hệ thống Gann HiLo
• Máy vẽ mức Gann
• Biểu đồ Gann Swing
• Biểu đồ Gann Swing v41212
• Biểu đồ Gann Swing ở 3 chế độ có văn bản
• Trung bình hình học của khối lượng
• Nhận các đoạn trích tin tức Moneycontrol vào Amiboker
• Bộ công cụ Gfx
• Mẹo công cụ GFX
• Gordon Rose
• Phân tích ngành đồ họa
• Phân tích cổ phiếu ngành đồ họa
• Guppy Cloud
• Hệ thống giao dịch bán tự động
• Halftrend
• Phát hiện mô hình hài hòa
• Mô hình hài hòa
• Mô hình đầu và vai
• Bản đồ nhiệt V1

• Nến Heikin Ashi
• Heikin Ashi Delta
• Hệ thống Heikin Ashi
• Heikin-Ashi(Koma-Ashi) với Đường trung bình động
• Heinkin-Ashi
• Herman
• HH-LL-PriceBar
• Mã phát hiện Cao Thấp
• Sóng hình sin Hilbert
• Hỗ trợ & Kháng cự của Sóng hình sin Hilbert
• Sóng hình sin Hilbert với Đường trung bình động Hull
• Nghiên cứu Hilbert
• Chỉ số biến động lịch sử
• Quét biến động lịch sử - 6/100
• Quét biến động lịch sử - 50 ngày
• HLspread
• Cách thêm Biểu tượng tùy chọn IB
• Chỉ báo tỷ lệ hoàn vốn Hull
• Trình lấp đầy IB
• Xếp hạng cơ sở dữ liệu sức mạnh tương đối của IBD
• Trình xem cơ sở dữ liệu sức mạnh tương đối của IBD
• ICHIMOKU SIGNAL TRADER
• Hệ thống Ichimoku
• IchimokuBrianViorelRO
• Quét/chỉ báo NH-NH được cải thiện
• Trong Watch Danh sách
• Chỉ số 30 tuần cao so với thấp
• Trình khám phá chỉ báo (ZigZag)
• Thanh bên trong
• Thanh bên trong
• Phân kỳ hồi quy tuyến tính lợi suất liên thị trường
• kiểm tra tương tác giao dịch tùy ý
• Khối lượng trung bình trong ngày
• Hệ thống phá vỡ xu hướng Fibonacii trong ngày
• HEIKIN ASHI trong ngày mới
• Sức mạnh trong ngày
• Hệ thống phá vỡ xu hướng trong ngày
• Chỉ báo chồng khối lượng được vẽ ngược
• Công thức Javascript để skalp và cập nhật danh mục đầu tư
• CHAKRA SRI CỦA JEEVAN
• John Ehler
• Tiêu chuẩn Kelly
• Hôn và chạm với Phạm vi thực đã sửa đổi
• Bảng điều khiển kết quả năm giao dịch gần nhất – Mã AFL
• Bộ dao động Laugerre PPO
• Chỉ báo kênh bình phương nhỏ nhất
• Nến tuyến tính
• Pha Luna
• LunarPhase
• Chỉ báo MACD BB
• Bình luận về MACD
• Biểu đồ Histogram MACD - Thay đổi hướng
• Hiển thị chỉ báo MACD
• Manual Bracket Order Trader
• Biểu đồ độ rộng thị trường trong biểu đồ

• Chỉ số tạo điều kiện thị trường so với khối lượng
• Chỉ số trung bình thị trường
• Hồ sơ thị trường
• MCDX (Rồng nhiều màu mở rộng)
• MFE và MAE và giao dịch biểu đồ như chỉ báo
• mitalpradip
• MOCS
• Mô hình đầu và vai đã sửa đổi
• DVI đã sửa đổi
• Chỉ báo giá khối lượng động lượng (MVP)
• Biểu đồ thanh hàng tháng
• Kiểm tra "sự cố" của đường trung bình động
• MO_CrashZone
• Hộp MS Darvas với Exploration
• Khối lượng nhiều màu theo giá (VAP)
• MultiCycle 1.0
• Bản demo nhiều dải ruy băng
• Nhiều nhiễu xoang
• NABIN SINHA
• Nadaraya-Watson Envelope
• NASDAQ 100 Volume
• Neural Network Powered Smooth/Predictive RSI V2
• New HL Scanner
• Nick
• Nifty
• Đường Zigzag không vẽ lại
• Noor_Doodie
• Bây giờ Gửi Thông báo Đẩy Từ Amibroker
• NRx Exploration
• Thứ tự thứ n (1 - 8) Phù hợp Đa thức
• OBV với Hồi quy Tuyến tính
• Hệ thống Giao dịch Phá vỡ Phạm vi Mở
• Quyền chọn Mua, Bán và ngày cho đến thứ sáu tuần thứ ba.
• Ord Volume
• Biểu đồ P&F - Giá cao/thấp tháng 9 năm 2003
• Biểu đồ P&F với các kích thước hộp phạm vi
• Parabolic SAR trong VBScript
• Điểm xu hướng Chande tham số
• Pattenz
• Mẫu hình - Đột phá cơ sở hình chữ nhật trên High Vol
• Khám phá nhận dạng mẫu hình
• Mẫu hình_-_Rectangle_Base_Breakout_on_High_Vol 2
• Bộ dao động giá phần trăm
• Perceptron
• Kiểm tra hiệu suất
• Cân bằng lại danh mục đầu tư MUA & GIỮ theo định kỳ
• Peter Cooper
• Peterson
• Biểu đồ PF - Đóng cửa - Tháng 4 năm 2004
• Hệ thống giao dịch cuối ngày Pivot
• Công cụ tìm Pivot
• Pivot và giá
• Pivot cho biểu đồ Forex trong ngày
• Vẽ đường cong vốn chủ sở hữu mà không cần kiểm tra ngược?
• vẽ trục xoay ngày mai trên cơ sở dữ liệu trong ngày

• Vẽ tỷ lệ dừng/mục tiêu trực quan.
• Biểu đồ Point & Figure India Securities
• Đường Polyfit
• Position Sizer vers2, cổ phiếu và CFD
• Phần trăm thanh dương
• Prakash
• Prashanth
• Độ bền của giá
• OHLC hàng ngày trước
• Mật độ xác suất & Phân phối chuẩn Gauss
• Bảng lợi nhuận (Mã màu)
• Bộ dao động dự báo
• Hệ thống thoái lui số 1
• Bộ quyết định xu hướng PVT
• Biểu đồ ngành qp2 dưới dạng bảng trong biểu đồ cổ phiếu
• R-Squared
• Bộ dao động cầu vồng
• Chỉ số bước ngẫu nhiên
• Bộ lọc phạm vi - Chiến lược giao dịch
• Xếp hạng và sắp xếp cổ phiếu
• Xếp hạng danh sách theo dõi mã cổ phiếu
• Cân bằng lại Kiểm tra ngược tránh đòn bẩy
• Kết nối lại TWS
• Sức mạnh tương đối
• Chỉ số sức mạnh tương đối
• Biểu đồ đa sức mạnh tương đối lên đến 10 tickers
• Khối lượng tương đối
• Rene Rijnaars
• Biểu đồ Renko
• Biểu đồ Renko
• Đảo ngược MFI Crossover
• Biểu đồ Renko đã sửa đổi
• RI - Hệ thống giao dịch tự động
• Robert Antony
• RSI + Avgs
• RSI Double-Bottom
• Chỉ báo RSI với Thanh vùng trên và dưới
• RSI của Mảng giá hàng tuần
• Đường xu hướng và nêm RSI
• Tín hiệu thời gian RUTVOL với thói quen ghi điểm BB
• Sainath Sidgiddi
• SAR-ForNextBarStop
• Ghi chú về Say
• Mở rộng quy mô: Hợp đồng tương lai
• Quét mức cao mới và mức thấp mới
• Đường Schiff
• Theo dõi ngành
• SectorRSI
• Gửi cảnh báo từ Amibroker đến Telgram
• shailu lunia
• Tín hiệu nhiễu
• Khám phá nến đơn giản
• Chỉ báo sóng sin
• Động lượng thích ứng được làm mịn
• Sony

• Hệ số tương quan thứ hạng Spearman
• Biểu đồ lộ trình Square of Nine
• Chiến lược của Stan Weinstein
• STD_STK Multi
• Kênh Cum. Vol. Float + Cum. Vol. của Steve Woods
• Steve Woods' Cumulative Vol. Chỉ báo phần trăm
• Stochastic %J - KDJ
• Phân kỳ ngẫu nhiên, tiêu cực
• Phân kỳ ngẫu nhiên, tích cực
• Phân kỳ ngẫu nhiên, PDI, NDI
• Phân kỳ ngẫu nhiên của Mảng giá hàng tuần
• Dao động ngẫu nhiên
• Đường xu hướng ngẫu nhiên
• StochD_StochK Single.afl
• Dừng triển khai trong AFS
• Dừng theo phần trăm
• Sức mạnh và điểm yếu
• Căng thẳng với SuperSmoother
• ĐIỂM SIÊU TRỤC
• Chỉ báo siêu xu hướng
• Chỉ báo siêu xu hướng
• Hỗ trợ và kháng cự
• Hỗ trợ và kháng cự
• Mức hỗ trợ và kháng cự
• Mức hỗ trợ và kháng cự
• suresh
• ĐỌC TAPE
• Khám phá phương pháp giao dịch TAZ
• Kênh TD-1
• Kênh TD-2
• Đường trung bình động TD 2
• TD Đường trung bình động I
• TD REI
• TD tuần tự
• TD tuần tự
• thử nghiệm nhiều hệ thống đồng thời
• D_oscillator
• Hành vi Fibonaccian
• Điểm cân bằng ba ngày
• Ba đường ngắt - TLB
• Ba cực Butterworth
• Khung thời gian Thanh hàng tuần
• Thời gian còn lại trong thanh
• tomy_frenchy
• Lỗi theo dõi
• Lỗi theo dõi
• Chỉ báo giao dịch mỗi giây
• Phân tích xu hướng_Bình luận
• Phát hiện xu hướng
• Khám phá xu hướng với nhiều khung thời gian
• Khám phá xu hướng: Đếm số lượng mức cao mới
• Khám phá xu hướng: Độ dốc Đường trung bình động
• Hệ thống theo dõi xu hướng
• Đường xu hướng từ 2 điểm
• Dải ruy băng xu hướng

• Mũi tên Dải ruy băng xu hướngADX
• Khám phá tam giác bằng Biểu đồ P&F
• Tìm kiếm tam giác mở rộng
• Tam giác Moving Average mới
• Trigonometric Fit - TrigFit với AR cho cos / sin
• TSV
• TTM Squeeze
• Twiggs Money Flow
• Twiggs money flow weekly
• TWS auto-export Executions-file parser
• TWS trade plotter
• Ultimate plus
• UltraEdit editor highlight wordfile
• Cập nhật Renko Chart
• Sử dụng ngày From và To từ Auto Analysis trong Code
• Vikram's Floor Pivot Intraday System
• Visi-Trade
• Visible Min và Max Value Demo
• Visual Turtle Trading system
• Visual image stoploss và profit trong biểu đồ
• Volatility Quality Index
• Volatility System
• Volume - so với Moving Avg (100%)
• Volume based support resistance price finder
• Volume Charts
• Volume Occilator
• Volume Spread cho VSA
• Bộ dao động vùng khối lượng
• Bộ lọc dự đoán Voss (Một cái nhìn thoáng qua về tương lai)
• VSTOP (2)
• VSTOP (3)
• VWAP - Giá trung bình có trọng số theo khối lượng
• Biểu đồ hàng tuần
• Xu hướng hàng tuần trong Biểu đồ hàng ngày
• Chỉ số có trọng số
• Chủ đề màu trắng
• Hệ thống Alligator II của William
• Hệ thống Alligator của Williams
• KÊNH GIÁ TƯƠNG ĐỐI CỦA WILSON
• Quy trình WLBuild
• Các mẫu sóng Wolfe
• Bảng điều khiển CCI của Woodie Cơ bản
• Bảng điều khiển CCI của Woodie Đầy đủ số liệu thống kê
• Bảng điều khiển Heikin-Ashi của Woodie
• Bảng điều khiển giá của Woodie với các điểm xoay của Woodie
• Zig Zag
• Chỉ báo Zig Zag với Điểm vào và Điểm thoát hợp lệ
• Zig-Hi Zap-Lo
• ZigZag - Ngày, Khối lượng trung bình (Trật tự) và Kênh
• Bộ lọc ZigZag được viết lại từ đầu trong AFL
• ZigZag Hi Lo Barcount
• ZigZag Retracements
• Hệ thống giao dịch chỉ mua ZLEMA ATR

Thông tin thêm:Xem phiên bản cập nhật/mở rộng trực tuyến.

_PARAM_VALUES- lấy chuỗi giá trị tham số

SYNTAX _PARAM_VALUES() TRẢ VỀ CHUỖI
HÀM _PARAM_VALUES lấy các giá trị của tham số được xác định trong phần kéo-thả hiện tại. Nó hoạt động giống như _DEFAULT_NAME ngoại trừ không có tên phần nào được bao gồm (do đó chỉ có danh sách các giá trị tham số được trả về).
VÍ DỤ
XEM THÊM hàm _DEFAULT_NAME()
Tham khảo:

Hàm _PARAM_VALUES được sử dụng trong các công thức sau trong thư viện trực tuyến AFL:

• Hệ thống MA nâng cao
• ALJEHANI
• babaloo chapora
• BBAreacolor&TGLCROSSNEW
• HaDiffCO của Bman
• bolingerbands
• bonlinger bands
• Tín hiệu giao dịch Coppock trên biểu đồ giá
• Tín hiệu giao dịch Coppock v1.1
• Hệ thống giao dịch Elder Triple Screen
• Guppy Cloud
• Hệ thống phá vỡ xu hướng Fibonacii trong ngày
• HEIKIN ASHI trong ngày mới
• prakash
• Hệ số tương quan thứ hạng Spearman
• ZigZag - Ngày, Khối lượng trung bình (Trật tự) và Kênh

Thông tin thêm:

Xem phiên bản cập nhật/mở rộng trực tuyến.

_SECTION_BEGIN- dấu bắt đầu phần

CÚ PHÁP _SECTION_BEGIN("tên phần") KHÔNG TRẢ VỀ
HÀM Đánh dấu phần bắt đầu của phần kéo-thả.

VÍ DỤ XEM THÊM
Tham khảo:

QUAN TRỌNG: "tên phần" phải là một chuỗi ký tự hằng số, được đặt trong dấu ngoặc kép. Bạn KHÔNG ĐƯỢC sử dụng biến ở đây.

Hàm _SECTION_BEGIN được sử dụng trong các công thức sau trong thư viện trực tuyến AFL:

• Pivots And Prices And Swing Volume
• 10-20 Indicator
• 2 Timeframes Candlestick Bar Chart
• 3 cách sử dụng RMI trong một tập lệnh
• 3TF Candlestick Bar Chart
• Tự động thêm Nifty 50 IB Equity Symbol
• Thêm SL/TGT các tham số khác vào bất kỳ chiến lược nào
• Hệ thống MA nâng cao
• Đường xu hướng nâng cao với S & R
• ADXVMA
• Ví dụ AFL
• Liên kết AFL sang Python COM
• AFL_Glossary_Converter
• ALJEHANI
• AllinOneAlerts - Module
• Alphatrend
• Animated BackGround
• Animated BackGround 1.1
• AO+Momentum
• Appel's ROC hoặc Mô hình thời gian động lượng ba
• Aroon The Advisor
• Tự động mở rộng và thoái lui Fib
• Tự động giao dịch từng bước
• Kênh xu hướng tuyến tính tự động 2
• Kênh xu hướng tự động
• Đường xu hướng tự động sử dụng nhiều khung thời gian
• AutoTrader Basic Flow - cập nhật ngày 15 tháng 4 năm 2009
• AutoTrader Basic Flow - cập nhật ngày 18 tháng 11 năm 2008
• Giao cắt giá trung bình
• B-Xtrender
• babaloo chapora
• Bad Tick Trim trên cơ sở dữ liệu 5 giây
• Hệ thống giao dịch giỏ hàng T101
• BBAreacolor&TGLCROSSNEW
• Bảng điều khiển giá mua so với giá bán
• HaDiffCO của Bman
• Độ rộng dải Bollinger
• Giao dịch nút bằng giao diện giao dịch tự động AB
• Caleb Lawrence
• Nhận dạng nến

• Bản demo nến
• Đèn chùm Thoát
• Thoát Chandelier
• thay đổi chu kỳ di chuyển trung bình
• Biểu đồ thu phóng
• CoinToss ver 1
• Giá Colorfull
• com-out
• Chỉ số lựa chọn hàng hóa (CSI)
• Bảng điều khiển đồng tích hợp tính toán và ADF
• Chuyển đổi hợp đồng liên tục
• Biểu đồ Histogram Coppock
• Tín hiệu giao dịch Coppock trên Biểu đồ giá
• Tín hiệu giao dịch Coppock v1.1
• Cao thấp hàng ngày trước
• Darvas Amibroker
• Nghiên cứu Darvas Johndeo
• Darvas Wisestocktrader
• DateNum_DateStr
• Quét PullBack của Dave Landry
• Thanh ngày Không
• DEBAJ
• Chỉ báo kỹ thuật số
• Đường trung bình động dịch chuyển DiNapolis 3x
• Chỉ số động lượng động
• Chỉ báo xung lực Elder V2
• Elder Ray - Bull Bear
• Elder Ray Oscillator với MA
• Elder safe Zone Long + short
• Elder Triple Screen Trading System
• Elder's Market Thermometer
• Elliott Wave labelling manual
• Elliott Wave Oscillator
• Ngày/ngày hết hạn - Thứ năm cuối cùng của tháng
• FastStochK FullStochK-D
• Tính toán Fibonacci & Speed Resistance
• Fibonacci Internal and External Retracements
• Fibonacci Moving averages
• Đối với thiết lập giao dịch tự động
• Force index
• Fre
• FTWHMS - CHIẾN LƯỢC ĐỘNG LƯỢNG CAO TRONG NĂM MƯƠI HAI TUẦN
• Dự báo MA trong tương lai
• Vẽ biểu đồ tương lai của thời gian và giá
• Vẽ biểu đồ tương lai của thời gian và giá
• Gann level plotter
• Gann Swing chart v41212
• Gann Swing Charts in 3 mode with text
• Geometric Khối lượng trung bình
• Nhận Moneycontrol News Snippets vào Amiboker
• GFX ToolTip
• Phân tích ngành theo đồ họa
• Phân tích cổ phiếu theo ngành theo đồ họa
• Guppy Cloud
• Halftrend

• Phát hiện mẫu hình hài hòa
• Mẫu hình hài hòa
• Heatmap V1
• Nến Heikin Ashi
• Heikin Ashi Delta
• HH-LL-PriceBar
• Mã phát hiện High Low
• Chỉ số biến động lịch sử
• Quét biến động lịch sử - 6/100
• Quét biến động lịch sử - 50 ngày
• Cách thêm Biểu tượng tùy chọn IB
• Chỉ báo tỷ lệ hoàn vốn của Hull
• IB Backfiller
• IchimokuBrianViorelRO
• Quét/chỉ báo NH-NH được cải thiện
• Trong danh sách theo dõi
• Thanh bên trong
• Thanh bên trong
• Trong ngày Hệ thống phá vỡ xu hướng Fibonacii
• INTRADAY HEIKIN ASHI mới
• Sức mạnh trong ngày
• Hệ thống phá vỡ xu hướng trong ngày
• Chỉ báo chồng khối lượng được vẽ ngược
• JEEVAN'S SRI CHAKRA
• Hôn và chạm với Phạm vi thực đã sửa đổi
• Bảng điều khiển kết quả năm giao dịch gần nhất – mã AFL
• Bộ dao động Laugerre PPO
• Nến tuyến tính
• Giai đoạn Luna
• Chỉ báo MACD BB
• Biểu đồ Histogram MACD - Thay đổi hướng
• Nhà giao dịch lệnh khung thủ công
• Biểu đồ độ rộng thị trường trong biểu đồ
• Chỉ số tạo thuận lợi cho thị trường so với khối lượng
• MCDX (Rồng nhiều màu mở rộng)
• mitalpradip
• Mô hình đầu và vai đã sửa đổi
• DVI đã sửa đổi
• Hộp MS Darvas với Exploration
• Bản demo nhiều dải ruy băng
• Nadaraya-Watson Envelope
• NASDAQ 100 Volume
• Mạng nơ-ron cung cấp RSI dự đoán/mượt mà V2
• Nick
• Đường Zigzag không vẽ lại
• Noor_Doodie
• Bây giờ gửi thông báo đẩy từ Amibroker
• Hệ thống giao dịch đột phá phạm vi mở
• Quyền chọn mua, bán và ngày cho đến thứ sáu tuần thứ ba.
• Điểm xu hướng Chande tham số
• pattenz
• Bộ dao động giá phần trăm
• Peter Cooper
• Điểm xoay và giá
• Vẽ đường cong vốn chủ sở hữu mà không cần kiểm tra ngược?

• Vẽ tỷ lệ dừng/mục tiêu trực quan.
• Biểu đồ Point & figure India Securities
• Position Sizer vers2, cổ phiếu và CFD
• Phần trăm thanh dương
• Prakash
• OHLC hàng ngày trước
• Chỉ số Random Walk
• Bộ lọc phạm vi - Chiến lược giao dịch
• Kết nối lại TWS
• Khối lượng tương đối
• Rene Rijnaars
• RI - Hệ thống giao dịch tự động
• Robert Antony
• Ghi chú về bài phát biểu
• Mở rộng quy mô: Hợp đồng tương lai
• Gửi cảnh báo từ Amibroker đến Telgram
• Biểu đồ lộ trình Square of Nine
• Chiến lược của Stan Weinstein
• Stochastic %J - KDJ
• Stochastic Oscillator
• Dừng theo phần trăm
• Căng thẳng với SuperSmoother
• ĐIỂM SIÊU TRỤC
• Chỉ báo siêu xu hướng
• ĐỌC BĂNG
• Kênh TD-1
• Kênh TD-2
• Đường trung bình động TD 2
• Đường trung bình động TD I
• TD REI
• TD Sequential
• Chỉ báo giao dịch mỗi giây
• Dải ruy băng xu hướng
• Mũi tên Dải ruy băng xu hướng ADX
• Đường trung bình động tam giác mới
• TSV
• TTM Squeeze
• Dòng tiền Twiggs
• Công cụ vẽ biểu đồ giao dịch TWS
• Ultimate plus
• Tệp từ tô sáng của trình chỉnh sửa UltraEdit
• Biểu đồ Renko được cập nhật
• Hệ thống trong ngày Floor Pivot của Vikram
• Giá trị tối thiểu và tối đa có thể nhìn thấy Bản demo
• Hệ thống giao dịch Visual Turtle
• Volume Occilator
• Volume Spread cho VSA
• Volume Zone Oscillator
• VWAP - Giá trung bình có trọng số theo khối lượng
• William's Alligator System II
• ZigZag - Ngày, Khối lượng trung bình (Ord) và Kênh
• Bộ lọc ZigZag được viết lại từ đầu trong AFL
• ZigZag Hi Lo Barcount
• ZigZag Retracements
• Hệ thống giao dịch chỉ mua ZLEMA ATR

Thông tin thêm:

Xem phiên bản cập nhật/mở rộng trực tuyến.

_SECTION_END- đánh dấu phần kết thúc

CÚ PHÁP _SECTION_END() KHÔNG TRẢ VỀ GÌ
HÀM đánh dấu phần kết thúc của phần kéo-thả
VÍ DỤ XEM THÊM
Tham khảo:

Hàm _SECTION_END được sử dụng trong các công thức sau trong thư viện trực tuyến AFL:

• Pivots And Prices And Swing Volume
• Chỉ báo 10-20
• Biểu đồ thanh nến 2 khung thời gian
• 3 cách sử dụng RMI trong một tập lệnh
• Biểu đồ thanh nến 3TF
• Tự động thêm Biểu tượng cổ phiếu Nifty 50 IB
• Hệ thống MA nâng cao
• Đường xu hướng nâng cao với S & R
• ADXVMA
• Ví dụ về AFL
• Liên kết COM từ AFL sang Python
• AFL_Glossary_Converter
• ALJEHANI
• AllinOneAlerts - Module
• Alphatrend
• Animated BackGround
• Animated BackGround 1.1
• AO+Momentum
• Aroon The Advisor
• Auto Fib Ext&Retracement
• Automatic Linear Trend Channel 2
• Automatic trend channel
• Automatic trend lines using multiple frames
• AutoTrader Basic Flow - cập nhật ngày 15 tháng 4 năm 2009
• AutoTrader Basic Flow - cập nhật ngày 18 tháng 11 năm 2008
• Average Price Crossover
• B-Xtrender
• babaloo chapora
• Bad Tick Trim trên cơ sở dữ liệu 5 giây
• Basket Trading System T101
• BBAreacolor&TGLCROSSNEW
• Bid Vs Ask Dashboard
• Bman's HaDiffCO
• Bollinger Band Width
• Button trading sử dụng giao diện giao dịch tự động AB
• Caleb Lawrence
• Nhận dạng nến
• Biểu đồ nến
• Thoát Chandelier
• Thoát Chandelier
• Thay đổi trung bình động theo chu kỳ
• Thu phóng biểu đồ
• CoinToss ver 1
• Giá đầy màu sắc

• com-out
• Chỉ số lựa chọn hàng hóa (CSI)
• Tính toán đồng tích hợp và Bảng điều khiển ADF
• Chuyển đổi hợp đồng liên tục
• Biểu đồ Histogram Coppock
• Tín hiệu giao dịch Coppock trên Biểu đồ giá
• Tín hiệu giao dịch Coppock v1.1
• Cao thấp hàng ngày trong Dự báo
• Darvas Amibroker
• Nghiên cứu Darvas Johndeo
• Darvas Wisestocktrader
• Quét PullBack của Dave Landry
• Thanh ngày Không
• DEBAJ
• Chỉ báo kỹ thuật số
• Đường trung bình động dịch chuyển 3x DiNapolis
• Chỉ số động lượng động
• Chỉ báo xung lực Elder V2
• Elder Ray - Bull Bear
• Bộ dao động Elder Ray với MA
• Vùng an toàn Elder Long + short
• Hệ thống giao dịch Elder Triple Screen
• Nhiệt kế thị trường Elder
• Nhãn thủ công sóng Elliott
• Bộ dao động sóng Elliott
• Ngày/ngày hết hạn - Thứ năm cuối cùng của tháng
• FastStochK FullStochK-D
• Tính toán Fibonacci & Sức cản tốc độ
• Đường thoái lui bên trong và bên ngoài Fibonacci
• Đường trung bình động Fibonacci
• Dành cho thiết lập giao dịch tự động
• Chỉ số lực
• Fre
• FTWHMS - CHIẾN LƯỢC ĐỘNG LƯỢNG CAO TRONG NĂM MƯƠI HAI TUẦN
• Dự báo MA trong tương lai
• Máy vẽ biểu đồ mức Gann
• Biểu đồ Gann Swing v41212
• Biểu đồ Gann Swing ở 3 chế độ có văn bản
• Trung bình hình học của Khối lượng
• Nhận Moneycontrol News Snippets vào Amiboker
• GFX ToolTip
• Phân tích ngành theo đồ họa
• Phân tích cổ phiếu theo ngành theo đồ họa
• Guppy Cloud
• Halftrend
• Phát hiện mẫu hình Harmonic
• Mẫu hình Harmonic
• Heatmap V1
• Nến Heikin Ashi
• Heikin Ashi Delta
• HH-LL-PriceBar
• Mã phát hiện High Low
• Chỉ số biến động lịch sử
• Quét biến động lịch sử - 6/100
• Quét biến động lịch sử - 50 ngày

• Cách thêm Biểu tượng tùy chọn IB
• Chỉ báo tỷ lệ hoàn vốn của Hull
• IB Backfiller
• ICHIMOKU SIGNAL TRADER
• IchimokuBrianViorelRO
• Quét/chỉ báo NH-NH được cải thiện
• Trong danh sách theo dõi
• Bên trong Thanh
• Thanh bên trong
• Hệ thống phá vỡ xu hướng Fibonacii trong ngày
• HEIKIN ASHI trong ngày mới
• Sức mạnh trong ngày
• Hệ thống phá vỡ xu hướng trong ngày
• Chỉ báo chồng khối lượng được vẽ ngược
• CHAKRA SRI CỦA JEEVAN
• Hôn và chạm với Phạm vi thực đã sửa đổi
• Bảng điều khiển kết quả năm giao dịch gần nhất – mã AFL
• Bộ dao động Laugerre PPO
• Nến tuyến tính
• Pha Luna
• Chỉ báo MACD BB
• Biểu đồ MACD Histogram - Thay đổi hướng
• Nhà giao dịch lệnh khung thủ công
• Biểu đồ độ rộng thị trường trong biểu đồ
• Chỉ số tạo thuận lợi cho thị trường so với khối lượng
• MCDX (Rồng nhiều màu mở rộng)
• mitalpradip
• Mẫu đầu và vai đã sửa đổi
• DVI đã sửa đổi
• Hộp MS Darvas với Exploration
• Nhiều dải ruy băng Bản demo
• Nadaraya-Watson Envelope
• NASDAQ 100 Volume
• Mạng nơ-ron hỗ trợ Smooth/Predictive RSI V2
• Đường Zigzag không vẽ lại
• Noor_Doodie
• Bây giờ Gửi thông báo đẩy từ Amibroker
• Hệ thống giao dịch đột phá phạm vi mở
• Quyền chọn mua, bán và ngày cho đến thứ sáu tuần thứ ba.
• Điểm xu hướng Chande tham số
• pattenz
• Bộ dao động giá phần trăm
• Peter Cooper
• Trục và giá
• Vẽ đường cong vốn chủ sở hữu mà không cần kiểm tra ngược?
• Vẽ tỷ lệ dừng lỗ/mục tiêu trực quan.
• Biểu đồ Point & Figure India Securities
• Position Sizer vers2, cổ phiếu và CFD
• Phần trăm thanh dương
• Prakash
• OHLC hàng ngày trước
• Chỉ số Random Walk
• Bộ lọc phạm vi - Chiến lược giao dịch
• Kết nối lại TWS
• Khối lượng tương đối

• Rene Rijnaars
• RI - Hệ thống giao dịch tự động
• Robert Antony
• Ghi chú về Say
• Mở rộng quy mô: Hợp đồng tương lai
• Gửi cảnh báo từ Amibroker đến Telgram
• Biểu đồ lộ trình Square of Nine
• Chiến lược Stan Weinstein
• Stochastic %J - KDJ
• Bộ dao động Stochastic
• Dừng theo phần trăm
• Căng thẳng với SuperSmoother
• ĐIỂM SIÊU TRỤC
• Chỉ báo siêu xu hướng
• ĐỌC BĂNG
• Kênh TD-1
• TD Kênh-2
• Đường trung bình động TD 2
• Đường trung bình động TD I
• TD REI
• TD Sequential
• Chỉ báo giao dịch mỗi giây
• Dải ruy băng xu hướng
• Mũi tên Dải ruy băng xu hướng ADX
• Đường trung bình động tam giác mới
• TSV
• TTM Squeeze
• Dòng tiền Twiggs
• Công cụ vẽ biểu đồ giao dịch TWS
• Ultimate plus
• Tệp word tô sáng trình soạn thảo UltraEdit
• Biểu đồ Renko được cập nhật
• Hệ thống trong ngày Vikram's Floor Pivot
• Visi-Trade
• Bản demo giá trị tối thiểu và tối đa Visible
• Hệ thống giao dịch Visual Turtle
• Công cụ tìm giá kháng cự hỗ trợ dựa trên khối lượng
• Volume Occilator

• Chênh lệch khối lượng cho VSA
• Bộ dao động vùng khối lượng
• VWAP - Giá trung bình có trọng số theo khối lượng
• Hệ thống Alligator II của William
• Bảng điều khiển CCI của Woodie Thống kê đầy đủ
• ZigZag - Ngày, Khối lượng trung bình (Ord) và Kênh
• Bộ lọc ZigZag được viết lại từ đầu trong AFL
• ZigZag Hi Lo Barcount
• ZigZag Retracements
• Hệ thống giao dịch chỉ mua ZLEMA ATR

Thông tin thêm:

Xem phiên bản cập nhật/mở rộng trực tuyến.

_SECTION_NAME- lấy tên phần hiện tại

SYNTAX _SECTION_NAME() TRẢ VỀ CHUỖI
HÀM Hàm cung cấp tên của phần kéo-thả (được cung cấp trong lệnh gọi _SECTION_BEGIN trước đó).
VÍ DỤ
XEM THÊM hàm _SECTION_BEGIN()
Tham khảo:

Hàm _SECTION_NAME được sử dụng trong các công thức sau trong thư viện trực tuyến AFL:

• ALJEHANI
• Elder Triple Screen Trading System
• UltraEdit editor highlight wordfile

Thông tin thêm:

Xem phiên bản cập nhật/mở rộng trực tuyến.

_TRACE- in văn bản vào trình xem gỡ lỗi hệ thống

CÚ PHÁP _TRACE("string") KHÔNG TRẢ VỀ GÌ
CHỨC NĂNG Ghi thông báo gỡ lỗi từ mã AFL vào trình xem gỡ lỗi hệ thống (nó gọi hàm API OutputDebugString Win bên trong) hoặc vào cửa sổ Nhật ký bên trong (Cửa sổ->Nhật ký) Để xem thông báo gỡ lỗi được gửi đến trình gỡ lỗi hệ thống, bạn phải chạy chương trình phần mềm miễn phí DebugView từ https://learn.microsoft.com/en-us/sysinternals/downloads/debugview Để xem thông báo được gửi đến cửa sổ nhật ký bên trong, bạn cần hiển thị cửa sổ nhật ký (menu Cửa sổ->Nhật ký)

Lưu ý đối với trình xem bên trong: bạn có thể chỉ định _TRACE("!CLEAR!"); để xóa cửa sổ nhật ký bên trong
VÍ DỤ _TRACE("Đây là một bài kiểm tra");
_TRACE("Đây là giá trị đã chọn của lệnh đóng: " + Đóng);
_TRACE("Đây là phần tử đầu tiên của mảng đóng: " + Đóng[0]);

XEM THÊM
Tài liệu tham khảo:

Hàm _TRACE được sử dụng trong các công thức sau trong thư viện trực tuyến AFL:

• Pivots And Prices And Swing Volume
• Auto Trade Step by Step
• AutoTrader Basic Flow - cập nhật ngày 15 tháng 4 năm 2009
• AutoTrader Basic Flow - cập nhật ngày 18 tháng 11 năm 2008
• babaloo chapora
• Button trading sử dụng giao diện giao dịch tự động AB
• Tính toán hợp chất cho các mã chứng khoán trong tệp danh sách
• Chỉ báo phân kỳ CCI(20)
• Phát hiện tắc nghẽn
• Tạo danh sách các hàm trong chương trình của bạn
• Đường cong vốn chủ sở hữu chi tiết
• TSP của Ed Seykota: Hệ thống giao cắt EMA
• TSP của Ed Seykota: Hỗ trợ và kháng cự
• Trích xuất các dòng mã cụ thể từ chương trình của bạn
• Để thiết lập giao dịch tự động
• Herman
• lastNDaysBeforeDate
• Chỉ số Market Meanness
• MFE và MAE và vẽ biểu đồ giao dịch như một chỉ báo
• Ord Volume
• Pivot Finder
• Vẽ biểu đồ tỷ lệ dừng/mục tiêu trực quan.
• Bảng lợi nhuận (Mã màu)
• Scale Out: Hợp đồng tương lai
• shailu lunia
• suresh
• Màu khối lượng với Giới hạn động
• WLBuildProcess
Thông tin thêm:
Xem phiên bản cập nhật/mở rộng trực tuyến.

_TRACEF- in văn bản được định dạng vào trình xem gỡ lỗi hệ thống

CÚ PHÁP _TRACEF("format string", arg1,)
TRẢ VỀ KHÔNG CÓ GÌ
HÀM Hàm này giống như _TRACE nhưng đối số đầu tiên là chuỗi định dạng theo kiểu printf và cho phép xuất các số được định dạng như sự kết hợp của _TRACE và StrFormat.

Hàm này ghi các thông báo gỡ lỗi từ mã AFL vào trình xem gỡ lỗi hệ thống (nó gọi hàm API OutputDebugString Win bên trong) hoặc vào cửa sổ Nhật ký bên trong (Window->Log)

Để xem các thông báo gỡ lỗi được gửi đến trình gỡ lỗi hệ thống, bạn phải chạy chương trình phần mềm miễn phí DebugView từ https://learn.microsoft.com/en-us/sysinternals/downloads/debugview

Để xem các thông báo được gửi đến cửa sổ nhật ký bên trong, bạn cần hiển thị cửa sổ nhật ký (menu Window->Log)

Lưu ý đối với trình xem bên trong: bạn có thể chỉ định _TRACE("!CLEAR!"); để xóa cửa sổ nhật ký nội bộ
VÍ DỤ _TRACEF("Giá đóng là = %g", Đóng);
XEM THÊM hàm _TRACE()
Tham khảo:

Hàm _TRACEF được sử dụng trong các công thức sau trong thư viện trực tuyến AFL:

Thông tin thêm:

Xem phiên bản cập nhật/mở rộng trực tuyến.

Danh sách lỗi AFL

• Lỗi 1. Không cho phép thao tác. Kiểu toán tử/toán hạng không khớp.

• Lỗi 2. Kiểu đối số không đúng. Đang mong đợi số hoặc mảng ở đây.

• Lỗi 3. Kiểu không khớp, toán tử trừ một ngôi yêu cầu số hoặc mảng

• Lỗi 4. Kiểu đối số không đúng. Đang mong đợi số ở đây.

• Lỗi 5. Đối số #1 có kiểu không đúng (hàm mong đợi kiểu đối số khác ở đây)

• Lỗi 6. Điều kiện trong các câu lệnh IF, WHILE, FOR phải là kiểu Số hoặc Boolean. Bạn không thể sử dụng mảng ở đây

• Lỗi 7. Điều kiện trong các câu lệnh IF, WHILE, FOR phải là kiểu Số hoặc Boolean. Bạn không thể sử dụng STRING ở đây

• Lỗi 8. Kiểu không khớp, giá trị được gán cho phần tử mảng phải là số. Bạn không thể sử dụng mảng...

• Lỗi 9. Chỉ số mảng phải là số

• Lỗi 10. Chỉ số nằm ngoài phạm vi. Bạn không được truy cập các phần tử mảng nằm ngoài phạm vi 0..(BarCount-1)

• Lỗi 11. Toán tử chỉ số [] yêu cầu kiểu mảng hoặc số. Không thể sử dụng chuỗi ở đây.

• Lỗi 12. Toán tử chỉ số [] yêu cầu kiểu mảng hoặc số.

• Lỗi 13. Phát hiện vòng lặp vô tận trong vòng lặp WHILE

• Lỗi 14. Phát hiện vòng lặp vô tận trong vòng lặp DO-WHILE

• Lỗi 15. Phát hiện vòng lặp vô tận trong vòng lặp FOR

• Lỗi 16. Quá nhiều đối số

• Lỗi 17. Thiếu đối số

• Lỗi 18. Biến đối tượng COM không được khởi tạo hoặc có kiểu không hợp lệ (yêu cầu xử lý đối tượng COM hợp lệ)

• Lỗi 19. Gọi 'tên hàm' của phương thức/hàm COM không thành công.

• Lỗi 20. Phương thức/hàm COM '%s' không tồn tại

• Lỗi 21. Không tìm thấy ký hiệu cơ sở cường độ tương đối

• Lỗi 22. Kiểu đối số 'format' không hợp lệ - định dạng phải là một số (không phải mảng)

• Lỗi 23. Cuộc gọi GetExtraData không thành công

• Lỗi 24. Công thức này yêu cầu AmiBroker phiên bản '...' (hoặc cao hơn).

• Lỗi 25. SetVariable() được gọi từ plug-in không thành công. Mã định danh đã được sử dụng trong một ngữ cảnh khác

• Lỗi 26. Xử lý tệp được truyền cho hàm không hợp lệ (bằng không).

• Lỗi 27. Số lượng đối số được truyền cho Hàm gọi (..) từ DLL của plugin không hợp lệ

• Lỗi 28. Hết bộ nhớ

• Lỗi 29. Biến được sử dụng mà không được khởi tạo.

• Lỗi 30. Lỗi cú pháp

• Lỗi 31. Lỗi cú pháp, mong đợi 'danh sách mã thông báo'

• Lỗi 32. Lỗi cú pháp, có thể thiếu dấu chấm phẩy ở cuối dòng trước đó

• Lỗi 33. Mã định danh đã được sử dụng. Bạn đã cố gắng gán giá trị cho mã định danh đại diện cho một hàm.

• Lỗi 34. Mã định danh đã được sử dụng. Bạn đã cố gắng xác định hàm có cùng mã định danh với biến toàn cục.

• Lỗi 35. Shift+BREAK được nhấn. Vòng lặp đã kết thúc.

• Lỗi 36. Đối số thứ N của lệnh gọi hàm không có giá trị được đặt

• Lỗi 37. Trường không được hỗ trợ trong SetOptions

• Lỗi 38. Trường không được hỗ trợ trong GetOptions

• Lỗi 39. CategoryAddSymbol: Thiết lập sector không được hỗ trợ, hãy thay thế bằng industry

• Lỗi 40. CategoryRemoveSymbol: Xóa khỏi sector không được hỗ trợ, hãy thay thế bằng remove khỏi industry

• Lỗi 41. Trường không được hỗ trợ trong GetRTData

• Lỗi 42. #include không thành công vì tệp không tồn tại: 'filename' (thư mục làm việc hiện tại là '...')

• Lỗi 43. Các điểm dừng biến không được hỗ trợ trong chế độ Giao dịch luân phiên

• Lỗi 44. SectorID() nằm ngoài phạm vi 0..63.

• Lỗi 45. Không khởi chạy được giao diện giao dịch

• Lỗi 46. Thiếu dấu phẩy

• Lỗi 47. Đã xảy ra ngoại lệ trong quá trình thực thi công thức AFL

• Lỗi 48. Thanh N khối lượng nén dữ liệu dài hơn khung thời gian cơ sở. Sử dụng hệ số nén cao hơn.

• Lỗi 49. Tên tham số tối ưu hóa không được để trống.

• Lỗi 50. Giá trị tối thiểu của tham số tối ưu hóa phải nhỏ hơn hoặc bằng giá trị tối đa và tham số bước phải lớn hơn không.

• Lỗi 51. Chỉ số mảng có giá trị Null. Chỉ số phải nằm trong phạm vi 0...BarCount-1.

• Lỗi 52. Giá trị đối số không hợp lệ cho hàm name(). Đối số 'paramname' phải là số dương (và không phải Null)

• Lỗi 53. Bạn có N tệp đang mở mà bạn quên đóng. Bạn phải gọi hàm fclose() cho mọi tệp được mở bằng Lỗi 54. Chuỗi espace không đúng. Các chuỗi được hỗ trợ là \n, \r, \t, \'' và \\ (cho một dấu gạch chéo ngược)

• Lỗi 90. Không tìm thấy Công cụ tối ưu hóa được chỉ định

• Lỗi 91. OptimizerSetOption mong đợi giá trị STRING

• Lỗi 92. OptimizerSetOption mong đợi giá trị NUMBER (số vô hướng)

• Lỗi 93. Tùy chọn được chỉ định không được hỗ trợ / không khả dụng bởi trình tối ưu hóa đã chọn

• Lỗi 94. Trình tối ưu hóa bên ngoài không được chọn. Bạn phải gọi OptimizerSetEngine() trước khi gọi OptimizerSetOption()

• Cảnh báo 501. Gán trong điều kiện. Ý bạn là == thay vì = ?

• Cảnh báo 502. Bạn đang gọi hàm Plot()/PlotOHLC() hơn 500 lần, điều này rất kém hiệu quả. Giảm số lượng Cảnh báo 503. Sử dụng OLE / CreateObject / CreateStaticObject không thân thiện với đa luồng.

• Lỗi 701. Thiếu các chỉ định biến mua/bán

• Lỗi 702. Thiếu các chỉ định biến bán khống/bảo hiểm

• Lỗi 703. Giao dịch luân phiên yêu cầu biến PositionScore.

• Lỗi 704. Trong giao dịch luân phiên, bạn KHÔNG ĐƯỢC sử dụng các tín hiệu mua/bán/bán khống/bảo hiểm.

• Lỗi 705. Bạn không thể sử dụng HoldMinBars cùng với Allow Same Bar exit.

• Lỗi 706. Tính năng Show Arrows cần có danh sách Trade.

Lỗi 1. Thao tác Không Được Phép. Loại Toán Tử/Toán Hạng Không Khớp.

Ban đang sử dụng một toán tử số học, chuỗi, logic hoặc so sánh với một kiểu dữ liệu không hợp lệ. Lỗi này sẽ xảy ra, ví dụ, nếu bạn cố gắng nhân hai giá trị chuỗi.

```
a = "x" * 5; // sai, không thể nhân chuỗi với số
 b = "x" - "y"; // sai, không thể trừ hai chuỗi
 z = "x" + "y"; // đúng, việc nối các chuỗi là chấp nhận được
```

Lỗi 2. Loại đối số không chính xác. Đang đợi số hoặc mảng ở đây.

Lỗi này xảy ra khi gọi hàm toán học đơn đối số như sin() mà chỉ chấp nhận số và mảng, nhưng người dùng đã chỉ định chuỗi cho ví dụ.

x= sin("test"); // sin yêu cầu số hoặc mảng

Lỗi 3. Không khớp kiểu dữ liệu, toán tử âm phân biệt đòi hỏi số hoặc mảng.

Lỗi này xảy ra khi cố gắng sử dụng toán tử âm phân biệt cho chuỗi ký tự.

text2 = - "test"; // không thể sử dụng toán tử âm phân biệt (phủ định) với văn bản

Lỗi số 4: Loại tham số không đúng. Ở đây đang mong đợi một số.

Lỗi này xảy ra trong hàm ApplyStop khi Type, Mode, ExitAtStop hoặc tham số Volatile là một mảng.

Điều này sai - chế độ dừng (phần trăm/ điểm) không thể là mảng.

ApplyStop(stopTypeLoss, IIf(C > O, stopModePercent, stopModePoint), 5);

Lỗi số 5. Đối số #1 có Kiểu không chính xác (Hàm ở đây Yêu cầu một Kiểu đối số Khác)

Lỗi này xảy ra khi đối số được truyền trong lời gọi hàm có kiểu không hợp lệ. Ví dụ, khi bạn truyền chuỗi thay vì mảng

MA("test", 5); // sai, Moving average yêu cầu mảng là đối số đầu tiên

AddColumn("Test", "Caption"); // sai, AddColum yêu cầu mảng là đối số đầu tiên

AddTextColumn("Test", "Caption"); // đúng, AddTextColumn yêu cầu văn bản

Lỗi 6. Điều kiện trong các câu lệnh IF, WHILE, FOR phải là kiểu Số hoặc Boolean. Bạn không thể sử dụng mảng ở đây, vui lòng sử dụng [] (toán tử chỉ mục mảng) để truy cập vào các phần tử mảng.

Từ khóa if thực thi statement1 nếu biểu thức đúng (khác không); nếu else có mặt và biểu thức sai (bằng không), nó sẽ thực thi statement2. Sau khi thực thi statement1 hoặc statement2, quyền kiểm soát sẽ chuyển sang câu lệnh tiếp theo. Biểu thức phải có kiểu boolean (True/False) (do đó nó KHÔNG THỂ là ARRAY vì sẽ không có cách nào quyết định xem có thực thi statement1 hay không, nếu ví dụ như mảng là: [True,True,False,.....,False,True])

```
if( expression )
statement1;
else
statement2;
```

Ví dụ

```
 if( Close > Open ) // SAI
   Color = colorGreen; //statement 1
 else
   Color = colorRed; //statement 2

Plot(Close,"Colored Price",Color,styleCandle);
```

Ví dụ trên là SAI, vì Open và Close đều là mảng và biểu thức như Close > Open cũng là một ARRAY. Giải pháp phụ thuộc vào câu lệnh. Có thể thực hiện nó trên cơ sở từng thanh, với việc sử dụng vòng lặp FOR:

```
 for( i = 0; i < BarCount; i++ )

{
   if( Close[ i ] > Open[ i ] ) // ĐÚNG
         Color[ i ] = colorGreen;
   else
         Color[ i ] = colorRed;
 }

Plot( Close, "Colored Price", Color, styleCandle );
```

Trong trường hợp này cũng có thể sử dụng hàm IIf():

```
Color = IIf( Close > Open, colorGreen, colorRed ); // CŨNG ĐÚNG - làm việc trực tiếp trên mảng

Plot( Close, "Colored Price", Color, styleCandle );
```

Lỗi 7. Điều kiện trong các lệnh IF, WHILE, FOR phải là loại số học (Numeric) hoặc Boolean. Bạn không thể sử dụng STRING ở đây.

Lỗi này xảy ra khi bạn cố gắng sử dụng chuỗi làm điều kiện trong các lệnh if/while/for.

Ví dụ:

```
if( "text" ) // sai
{
// làm gì đó x = 1;
}
```

Điều kiện trong if/while/for nên được đánh giá là true/false:

```
if( "text" != "someothertext" ) // đúng, toán tử != (không bằng) cung cấp giá trị true/false
{
// làm gì đó x = 1;
}
```

Lỗi 8. Loại không phù hợp, giá trị được gán cho phần tử mảng phải là một số. Bạn không thể sử dụng mảng ở phía bên phải của phép gán này. Lỗi này xảy ra khi cố gắng gán toàn bộ mảng cho một phần tử đơn lẻ của mảng khác.

Đoạn mã dưới đây là một ví dụ:

```
test = 0;
for( i = 0; i < BarCount; i++ )
{
test[ i ] = Close ; // sai, một phần tử mảng chỉ có thể chứa một giá trị, không phải mảng
test[ i ] = Close[ i ]; // đúng
}
```

Lỗi 9. Chỉ số mảng phải là một số

Bạn chỉ có thể sử dụng số làm chỉ số mảng; chuỗi và mảng không được chấp nhận.

```
table[ 1 ] = 10; // đúng
 table[ "text" ] = 10; // sai
 table[ Close ] = 10; // sai
```

Lỗi 10: Subscript nằm ngoài phạm vi. Bạn không được phép truy cập vào các phần tử mảng ngoài phạm vi 0..(BarCount-1).

Điều này xảy ra khi bạn cố gắng truy cập các phần tử mảng có chỉ số dưới 0 (không) hoặc trên BarCount-1.

```
// không đúng
 for( bar = 0; bar < BarCount; bar++ ) {
  a[ bar ] = C[ bar - 1]; // khi i == 0, chúng ta đang truy cập C[-1] là sai
 }


 // đúng
 for( bar = 0; bar < BarCount; bar++ ) {
 if( bar > 0 )
  a[ bar ] = C[ bar - 1 ]; // chỉ truy cập C[ i - 1 ] khi i lớn hơn không
  else a[ bar ] = C[ 0 ];
 }
```

Một trong những lỗi phổ biến nhất là sử dụng giới hạn trên cứng của vòng lặp và giả sử rằng tất cả các ký hiệu đều có đủ trích dẫn.

Ví dụ:

```
 MyPeriod = 10;
 for( i = 0; i < MyPeriod; i++ ) // SAI! Giả định là bạn luôn có ít nhất 10 trích dẫn
 {
 // ... thực hiện một hành động nào đó
 }
```

Điều này sẽ luôn thất bại với các ký hiệu có ít trích dẫn hơn 10 và có thể cũng thất bại nếu bạn phóng to biểu đồ của mình để ít hơn 10 trích dẫn thực sự được hiển thị trên biểu đồ.

Để đảm bảo hoạt động không lỗi, bạn phải luôn kiểm tra chỉ số trên dưới BarCount, như được minh họa ở dưới đây:

```
 MyPeriod = 10;
 for( i = 0; i < MyPeriod AND i < BarCount; i++ ) // ĐÚNG - đã thêm kiểm tra giới hạn trên
 {
 // ... thực hiện một hành động nào đó
 }
```

Một cách khác, bạn chỉ có thể nhập vòng lặp khi có đủ thanh, như được minh họa trong đoạn mã sau:

```
 MyPeriod = 10;

if( MyPeriod <= BarCount ) // ĐÚNG - kiểm tra xem có đủ số thanh để chạy vòng lặp hay không
```

```
{
	for( i = 0; i < MyPeriod; i++ )
	{
	// ... thực hiện một số công việc
	}
}
```

Lỗi 11. Toán tử Subscript [] yêu cầu kiểu mảng hoặc kiểu số. Chuỗi không thể được sử dụng ở đây.

Lỗi này xảy ra khi bạn cố gắng sử dụng toán tử Subscript [] trên strings. Ví dụ:

```
tt = "Test";
x = tt[ 0 ];
```

Lỗi số 12: "Subscript operator []" đòi hỏi kiểu dữ liệu mảng hoặc số.

Đây là lỗi xảy ra khi "subscript operator []" được áp dụng cho các kiểu dữ liệu không được hỗ trợ (như đối tượng phân phối COM):

```
 a=CreateObject("Broker.Application");

b = a[0];
```

Lỗi 13. Phát hiện vòng lặp không kết thúc trong While Loop

Lỗi này xuất hiện khi máy chủ AFL phát hiện ra một vòng lặp while không bao giờ kết thúc (việc phát hiện dựa trên số lần lặp lại, AmiBroker chỉ đơn giản là đếm số lần lặp và nếu nó vượt quá giới hạn ngưỡng được đặt trong Tools->Preferences->AFL: Endless loop detection threshold - mặc định là 100000 lần lặp), hệ thống sẽ hiển thị thông báo lỗi này). Ví dụ:

i = 0; while(i < 5) x = i; // Biến i không được tăng lên, vì vậy vòng lặp không bao giờ kết thúc.

Lỗi 14. Phát hiện vòng lặp vô tận trong vòng lặp Do-While

Lỗi này xảy ra khi AFL engine phát hiện ra vòng lặp do-while không bao giờ kết thúc (việc phát hiện dựa trên số lần lặp, AmiBroker đơn giản chỉ đếm số lần lặp và nếu nó vượt quá giới hạn ngưỡng được đặt trong Tools->Preferences->AFL: Endless loop detection threshold - theo mặc định là 100000 lần lặp), nó hiển thị thông điệp này. Ví dụ:

```
i = 0;

do

{

x = i;

}

while( i < 5 ); // biến i không tăng, vì vậy vòng lặp không bao giờ kết thúc.
```

Lỗi 15. Vòng lặp For không kết thúc

Lỗi này xảy ra khi động cơ AFL phát hiện ra vòng lặp for không bao giờ kết thúc (việc phát hiện dựa trên số lần lặp lại, AmiBroker đơn giản đếm số lần lặp lại và nếu nó vượt quá giới hạn đã đặt trong công cụ -> Sở thích -> AFL: Giới hạn phát hiện vòng lặp không kết thúc - mặc định là 100000 lần lặp lại) nó sẽ hiển thị thông báo này). Ví dụ:

for(i = 0; i < BarCount; i) // quên ++ (toán tử tăng) vì thế vòng lặp sẽ không bao giờ kết thúc.

{

x = i;

}

Lỗi 16. Quá nhiều đối số

Lỗi này xảy ra khi bạn chỉ định quá nhiều đối số khi gọi hàm. Ví dụ:

m = MACD(12, 26, 3); // lỗi: MACD chỉ cần 2 tham số, nhưng ở đây lại chỉ định 3

Lỗi 17. Thiếu đối số

Xảy ra khi số lượng đối số truyền vào khi gọi hàm quá ít. Ví dụ:

Plot(C, "Price"); // quá ít đối số, đối số thứ 3 là bắt buộc - màu của biểu đồ

Lỗi 18: Biến đối tượng COM không được khởi tạo hoặc có loại không hợp lệ (yêu cầu có handle đối tượng COM hợp lệ)

Lỗi này xảy ra khi cố gắng sử dụng biến chưa được khởi tạo hoặc biến có loại không đúng để gọi phương thức đối tượng COM:

```
Obj = 1; // khởi tạo dưới dạng số
Obj.Test(); // thử gọi phương thức thất bại vì Obj không phải là handle đối tượng COM
```

Lỗi 19. Gọi Phương Thức/Hàm Com Thất Bại.

Một ngoại lệ OLE đã xảy ra trong quá trình gọi phương thức/hàm/thuộc tính OLE/COM. có thể chứa mô tả ngoại lệ OLE.

Thông thường thông báo này được hiển thị khi các đối số được chuyển đến phương thức OLE/COM bị sai hoặc thiếu.

AB=CreateObject("Broker.Application");

AB.Import(); // <-- fails with error 19. Trong trường hợp này, đó là do thiếu đối số

Lỗi 21: Không tìm thấy ký hiệu cơ sở sức mạnh tương đối.

Hàm RelStrength() được gọi với ký hiệu không tồn tại như là một tham số:

x = RelStrength("NonExistingTicker"); // không thành công vì ký hiệu sai

Lỗi 22: Loại đối số 'format' sai - 'format' cần phải là một số (không phải mảng)

Tham số format được chỉ định trong lời gọi hàm AddColumn() cần phải là một số (không phải mảng). Đây là một số ví dụ:

```
AddColumn( C, "Close", IIf( C > 10, 1.2, 1.3 ) ); // Sai, tham số format (thứ 3) cần phải là một số
AddColumn( C, "Close", 1.3 ); // Đúng
```

Lỗi 23. Getextradata không thể hoạt động

Lệnh GetExtraData() không hoạt động (hoặc không trả về giá trị) có thể do plugin hiện tại không hỗ trợ chức năng GetExtraData hoặc trường được chỉ định không được plugin hỗ trợ:

```
x=GetExtraData("nonexistingfieldname"); // tên trường không tồn tại
```

Lỗi 24. Công thức này yêu cầu phiên bản AmiBroker <...> (hoặc cao hơn).

Công thức này được thiết kế để sử dụng trên phiên bản AmiBroker cao hơn, và bạn nên nâng cấp để sử dụng nó.

Version(9.7); // chưa có phiên bản 9.70 :-)

Lỗi 25. Việc gọi SetVariable() từ plugin không thành công. Định danh đã được sử dụng trong một ngữ cảnh khác.

Tình trạng này xảy ra nếu plugin từ bên ngoài thử gọi SetVariable với định danh mà đã được sử dụng cho mục đích khác

(như là hàm được xây dựng sẵn hoặc do người dùng định nghĩa)

Lỗi 26. Lỗi này xảy ra khi file handle được truyền vào hàm không hợp lệ (bằng với zero). Bạn cần kiểm tra xem liệu file handle trả về từ fopen có khác với zero hay không. Nếu nó là zero, có nghĩa là file không thể được mở.

Điều này thường xảy ra khi cố gắng gọi hàm ghi/đọc/đóng file trên null file handle.

Ví dụ với cách sử dụng không chính xác:

```
fh = fopen("nonexistingfile.txt", "r"); // file này không tồn tại


fputs("Test", fh ); // lỗi ở đây, fh có thể là null
fclose( fh ); // sai, fh có thể là null
```

Cách sử dụng đúng sẽ giống như thế này:

```
fh = fopen("nonexistingfile.txt", "r"); // file này không tồn tại


if( fh ) // đúng, chỉ gọi hàm ghi/đọc/đóng file tiếp theo khi file handle hợp lệ
{
fputs("Test", fh );
fclose( fh );
}
```

Lỗi 27. Số lượng đối số không hợp lệ được truyền vào Hàm Gọi (..) từ plugin DLL

Chỉ xảy ra khi plugin bên ngoài gọi các hàm nội bộ của AmiBroker với số lượng đối số không chính xác.

Lỗi 28. Hết bộ nhớ (Out of memory)

Xảy ra lỗi hết bộ nhớ (Out-of-memory) trong quá trình phân tích công thức (should not happen under normal circumstances). Rất khó xảy ra trong những hoàn cảnh bình thường.

Lỗi 29. Biến Được Sử Dụng Mà Không Được Khởi Tạo Trước.

Bạn không thể sử dụng (đọc) biến mà không được khởi tạo trước. Bạn nên gán giá trị cho biến trước khi đọc nó.

Dưới đây là ví dụ về cách sử dụng sai:

```
x = 1;
z = x + y; // sai, y chưa được khởi tạo
```

Cách sử dụng đúng sẽ như sau:

```
x = 1; // khởi tạo x
y = 2; // khởi tạo y
z = x + y; // đúng, cả x và y đều được khởi tạo
```

Lỗi 30. Lỗi cú pháp (Syntax Error)

Đây là lỗi cú pháp chung. Lỗi này xảy ra khi cú pháp không chính xác. Ví dụ như dấu ngoặc không đủ cặp, chứa các ký tự không được công nhận, toán tử không hợp lệ hoặc tên hàm không chính xác/undefined, thiếu dấu ngoặc hoặc dấu chấm phẩy.

```
x = 4;

y = 2;

z = x * ( 7 + y ; // Lỗi về cú pháp tại đây do thiếu dấu ngoặc đóng
```

Lỗi 31: Syntax Error, Expecting

Cú pháp có vấn đề bởi vì parser muốn tìm thấy các token cụ thể nhưng lại phát hiện ra cái gì đó khác.

Ví dụ:

while i < 5 // điều này tạo ra Error 31: Syntax error, parser đang chờ đợi dấu mở ngoặc '(' sau câu lệnh while.

Lỗi cũng có thể xảy ra nếu một dấu ngoặc đóng, dấu kết thúc dòng hoặc dấu chấm phẩy bị thiếu ở một trong những dòng trước đó và parser đang tìm kiếm dấu ngoặc hoặc dấu kết thúc dòng phù hợp cho đến khi nó tìm thấy cái gì đó không thực sự thuộc về câu lệnh không cân đối trước đó.

Lỗi 32. Lỗi cú pháp, có thể thiếu dấu chấm phẩy ở cuối dòng trước

Đã xảy ra lỗi cú pháp chung ngay từ đầu dòng. Trong hầu hết các trường hợp, đó là do thiếu dấu chấm phẩy ở cuối câu lệnh trong dòng trước, ví dụ như:

a=5

b=4; // <--- lỗi cú pháp ở đây, có thể thiếu dấu chấm phẩy.

Tuy nhiên, trong một số trường hợp, nguyên nhân có thể đơn giản là cú pháp không đúng ngay từ đầu dòng, như:

a=5;

+b=4; // <-- cùng một thông báo lỗi, nhưng vấn đề nằm ở việc có một ký tự + dư thừa ở đầu dòng

Lỗi số 33. Trình nhận dạng đã bị sử dụng. Bạn đã cố gắng gán giá trị cho trình nhận dạng đại diện cho một hàm. Nếu bạn muốn trả về giá trị từ hàm, bạn nên sử dụng câu lệnh RETURN thay thế.

Ví dụ 1:

```
function Test( x )

{
        return 2 * x;
}
VarSet( "Test", 7 ); // error, Trình nhận dạng 'Test' đã được sử dụng cho hàm
```

Ví dụ 2: (không chính xác)

```
function Test( x )

{
Test = 2 * x; // lỗi tại đây vì trình nhận dạng Test đã được sử dụng cho hàm,
// bạn nên sử dụng câu lệnh return để trả về giá trị từ hàm
}
x = Test(5);
```

Hàm trả về giá trị đúng sẽ trông như thế này:

```
function Test( x ) {
return 2 * x;
// Chính xác, trả về giá trị bằng cách sử dụng câu lệnh return
}
x = Test(5);
```

Lỗi 34: Định danh đã được sử dụng. Bạn đã thử định nghĩa một hàm mà có cùng định danh với biến toàn cục.

Lỗi này xảy ra khi định nghĩa hàm sử dụng cùng một định danh với biến toàn cục được định nghĩa trước đó trong công thức.

```
Test = 5; // biến toàn cục
 function Test( x ) // không chính xác, định danh 'Test' đã được sử dụng cho biến toàn cục
 {

return 2 * x;

 }
```

Lỗi 35. Nhấn Shift+BREAK. Vòng lặp bị kết thúc.

Lỗi này xảy ra khi người dùng kết thúc thực hiện vòng lặp một cách thủ công bằng cách nhấn các phím Shift+BREAK.

Lỗi 36. Đối số thứ N của lệnh gọi hàm không có giá trị đã được đặt

Điều này có thể xảy ra nếu đối số thứ N của hàm được đặt thành không có giá trị. Điều này không thể xảy ra trong điều kiện bình thường, chỉ khi gọi các hàm từ bên trong plugin hoặc đặt các biến từ bên trong plugin.

Lỗi 37. Trường không được hỗ trợ trong SetOptions.

Xảy ra khi tên trường sai (hoặc không được hỗ trợ) đã được sử dụng trong lời gọi SetOption, ví dụ:

SetOption("NoSuchOption", 1);

Lỗi 38. Trường không được hỗ trợ trong GetOptions

xảy ra khi tên trường sai (hoặc không được hỗ trợ) được sử dụng trong lệnh GetOption, ví dụ:

x = GetOption("NoSuchOption");

Lỗi 39. Categoryaddsymbol: Việc Thiết Lập Ngành Không Được Hỗ Trợ, Hãy Thiết Lập Công Nghiệp Thay Thế

Xảy ra khi categorySector được sử dụng trong hàm CategoryAddSymbol. Bạn không thể thêm biểu tượng vào ngành vì biểu tượng chỉ được liên kết với các nhóm công nghiệp và sau đó các nhóm công nghiệp được gán cho các ngành. Một ngành thường bao gồm nhiều nhóm công nghiệp và nếu bạn thiết lập công nghiệp thì ngành được giả định. Tham khảo Hướng Dẫn Sử Dụng AmiBroker để biết thêm thông tin về các hạng mục.

```
CategoryAddSymbol( "", categorySector, 2 ); // sai, bạn không thể sử dụng categorySector

CategoryAddSymbol( "", categoryIndustry, 2 ); // đúng
```

Lỗi 40: Categoryremovesymbol - Không Hỗ Trợ Xóa Từ Ngành, Hãy Xóa Từ Công Nghiệp Thay Thế

Lỗi này xảy ra khi categorySector được sử dụng trong hàm CategoryRemoveSymbol. Bạn không thể xóa biểu tượng từ ngành vì các biểu tượng chỉ được liên kết với các nhóm công nghiệp và sau đó các nhóm công nghiệp được gán vào các ngành. Một ngành thường bao gồm nhiều nhóm công nghiệp và nếu bạn đã thiết lập công nghiệp thì ngành được đặt theo mặc định. Hãy tham khảo Hướng dẫn sử dụng của AmiBroker để biết thêm thông tin chi tiết về các loại danh mục.

```
CategoryRemoveSymbol( "", categorySector, 2 ); // Sai, bạn không thể sử dụng categorySector

CategoryRemoveSymbol( "", categoryIndustry, 2 ); // Đúng
```

Lỗi 41. Trường không được hỗ trợ trong Getrtdata

Lỗi này xảy ra khi một trường không được hỗ trợ đã được chỉ định trong lời gọi GetRTData:

GetRTData("EPSRank"); // EPSRank không có sẵn từ các nguồn RT.

Chú ý rằng GetRTData chỉ được hỗ trợ cho các nguồn dữ liệu thời gian thực và chỉ trong phiên bản Professional. Nếu bạn cố gắng gọi nó mà không chạy nguồn dữ liệu thời gian thực hoặc sử dụng phiên bản Standard của AmiBroker, nó sẽ im lặng trả về giá trị Null mà không hiển thị bất kỳ thông báo lỗi nào.

Lỗi 42. Lệnh #include không thành công vì không tìm thấy tệp tin: (thư mục làm việc hiện tại là '...')
Điều này xảy ra khi tệp bao gồm được chỉ định không tồn tại.Ví dụ:

```
#include "not\existing\path\to\the\file."
```

Lỗi 43: Chế Độ Dừng Biến Đổi Không Được Hỗ Trợ Trong Các Chế Độ Giao Dịch Xoay Vòng và Thô

Lỗi này xảy ra khi cố gắng sử dụng mức số lượng biến đổi trong hàm ApplyStop() khi sử dụng chế độ backtester giao dịch xoay vòng hoặc một trong những chế độ Thô (như backtestRegularRaw hoặc backtestRegularRawMulti). Ví dụ:

```
EnableRotationalTrading();


ApplyStop( stopTypeLoss, stopModePoint, H-L ); // lượng dừng biến đổi không được hỗ trợ trong chế độ xoay vòng
```

Lỗi 44. Hàm SectorID() nằm ngoài phạm vi 0..63.

Lỗi này có thể xảy ra trong quá trình gọi hàm SectorID() nếu plugin dữ liệu bên ngoài thiết lập ID của sector không chính xác (ngoài phạm vi 0..63).

x = SectorID(); // công thức đúng, nhưng có thể gây ra lỗi 44 nếu plugin dữ liệu thiết lập sector không chính xác

Lỗi 45. Không thể khởi chạy giao diện giao dịch
Vấn đề xảy ra khi công thức gọi GetTradingInterface nhưng giao diện cần thiết không được cài đặt hoặc đăng ký đúng cách.

```
javascript
ti = GetTradingInterface("DUMMY"); // thất bại bởi vì giao diện DUMMY không được cài đặt
```

Lỗi 46. Thiếu dấu phẩy.

Vấn đề xảy ra khi thiếu dấu phẩy trong danh sách tham số chính thức của khai báo hàm

```
function MyFun( x y ) // missing comma in the formal parameter list

 {
 return x * y;
 }
```

Lỗi 47. Ngoại lệ Xảy ra Trong Quá trình Thực thi Công thức Afl

Lỗi này xảy ra khi công thức gây ra ngoại lệ hệ thống không được xử lý. Ngoại lệ hệ thống có thể do tràn bộ nhớ, truy cập sai tay nắm tệp, vi phạm truy cập bộ nhớ, v.v.

Ví dụ:

fclose(123); // đóng tay nắm tệp không hợp lệ gây ra ngoại lệ hệ thống

Lỗi 48. Dữ liệu Nén của Thanh N-Khối lượng Lớn hơn Khung Thời gian Gốc. Sử dụng Hệ số Nén lớn hơn.

Lỗi này xảy ra khi cài đặt nén N-khối lượng tạo ra dữ liệu dài hơn tập dữ liệu gốc.

Thanh N-khối lượng rất khác biệt so với thanh dựa trên thời gian (việc nén dữ liệu thành thanh N-khối lượng có thể thực sự cung cấp THÊM các thanh đầu ra - ví dụ: nếu mỗi tick là 1000 cổ phiếu và bạn đã chỉ định thanh 100V thì một tick đơn sẽ được mở rộng thành MƯỜI thanh 100V - gấp mười lần kích thước gốc).

Các hàm TimeFrame được bảo vệ khỏi array overrun và sẽ không giải nén vượt quá kích thước mảng gốc (bạn sẽ nhận được một "Lỗi 48"). Cũng như vậy, việc chuyển khung thời gian chính sang một giá trị thanh N-khối lượng lạ sẽ dẫn đến việc hạn chế đầu ra ở tối đa gấp đôi kích thước dữ liệu gốc (không có thông báo lỗi).

Bạn nên ghi nhớ điều này và tránh sử dụng các khoảng thời gian thanh N-khối lượng quá nhỏ có thể dẫn đến tình trạng như vậy. Cũng do tính chất của thanh N-khối lượng, chỉ hàm TimeFrameSet() sẽ cho ra giá trị thanh N-khối lượng chính xác, TimeFrameGetPrice() có thể cung cấp kết quả bị méo một chút.

Ví dụ:
```
 TimeFrameMode( 2 );

TimeFrameSet( 20 ); // có thể có Lỗi 48 - nén thanh 20-cổ phiếu có thể quá nhỏ
```

Lỗi 49. Tên tham số tối ưu hóa không được để trống.

Lỗi này xảy ra khi hàm Optimize() được gọi với đối số tên trống.

Ví dụ:

period = Optimize("", 10, 10, 20, 1); // SAI: tên KHÔNG được để trống

Lỗi 50. Giá trị tối thiểu của tham số tối ưu hóa phải nhỏ hơn hoặc bằng giá trị tối đa, còn tham số bước phải lớn hơn không.

Lỗi này xảy ra khi hàm Optimize() của AFL được gọi với giá trị tối thiểu lớn hơn giá trị tối đa hoặc bước nhỏ hơn không chứ không lớn hơn.

Ví dụ:

```
period = Optimize("Period", 1, 20, 10, 1 ); # SAI: giá trị tối thiểu > giá trị tối đa
period2 = Optimize("Period2", 1, 10, 20, 0 ); # SAI: bước = 0
```

Lỗi 51: Chỉ số mảng có giá trị Null. Chỉ số phải nằm trong phạm vi từ 0 đến BarCount-1.

Bạn đã cố gắng sử dụng giá trị Null như một chỉ số mảng:

```
index = Null;
value = Close[index]; // lỗi 51.
```

Lỗi 52. Giá trị đối số không hợp lệ cho hàm <name>. **Đối số <paramname> phải là số dương (và không được để trống).**

Lỗi 52. Giá trị đối số không hợp lệ cho hàm <name>(). Hàm không hỗ trợ <paramvalue> được chỉ định.

Bạn đã cố gắng truyền một giá trị âm làm tham số cho một hàm mà nguyên vận dụng muốn giá trị dương hoặc hàm không chấp nhận giá trị đã cho của tham số.

Ví dụ: Sum(Close, -15);// Lỗi 52 - phạm vi phải là số dương

Lỗi 53: Bạn có N file mở mà bạn đã quên không đóng lại. Bạn phải gọi hàm fclose() cho mỗi file được mở bằng hàm fopen().

Bạn đã gọi hàm fopen() để mở file nhưng lại không gọi hàm fclose.

```
fh = fopen("test.txt", "w" );
 if( fh )

 {
 // fclose( fh ); // Hàm fclose() NÊN được gọi
 }
```

Lỗi 54. Chuỗi escape không hợp lệ. Các chuỗi được hỗ trợ là \n, \r, \t, \" và \\ (để cho một dấu gạch chéo ngược đơn)

Bạn đã chỉ định chuỗi escape không đúng trong hằng số chuỗi. Các chuỗi được hỗ trợ là:

- \n - để xuống dòng

- \r - để quay về đầu dòng

- \t - để tab

- \" - để dấu ngoặc kép

- \\ - để một dấu gạch chéo ngược đơn

fh = fopen("C:\windows\test.txt", "w"); // Lỗi 54 - chuỗi escape không hợp lệ. Dấu gạch chéo ngược đơn phải được mã hóa là \\

fh = fopen("C:\\windows\\test.txt", "w"); // ĐÚNG

fclose(fh);

Lỗi 55. Phép gán không hợp lệ. Định danh chỉ có thể đọc (hằng số) và không thể ghi dữ liệu.

Lỗi này xảy ra khi bạn cố gắng gán giá trị cho định danh chỉ đọc (hằng số), ví dụ như

colorBlack = 1200; // SAI ! colorBlack là một hằng số và KHÔNG thể ghi dữ liệu.

Lỗi cũng có thể xảy ra nếu bạn thực hiện phép gán như một phần của lời gọi hàm:

Plot(Close, "Price", colorRed = 7); // SAI ! gán giá trị cho định danh chỉ đọc (colorRed) trong lời gọi hàm

Lỗi 56. Biến '' Có Kiểu Không Hợp Lệ

Điều này có thể xảy ra nếu biến có kiểu không xác định (không được hỗ trợ). Các loại được hỗ trợ là: number, array, string, object, file handle, matrix. Thông thường, lỗi này không xuất hiện vì AmiBroker không sử dụng các loại không xác định nội bộ. Tuy nhiên, điều này có thể xảy ra nếu bạn đang sử dụng DLL bên thứ ba trả về các giá trị có loại cấu trúc AmiVar không đúng (hoặc không được đặt). Trong trường hợp đó, vui lòng liên hệ với nhà cung cấp plugin thứ ba để sửa lỗi. Lỗi cũng có thể xảy ra nếu có một số lỗi xảy ra và biến bị để lại trong một trạng thái ngẫu nhiên / không khởi tạo.

Lỗi 57. Dấu ngoặc kép không hợp lệ. Hãy thay thế nó bằng dấu ngoặc kép thẳng ("").

Lỗi này nghĩa là bạn đã từ đâu đó (như trang web) và mã đã dán chứa dấu ngoặc kép không chính xác. Chỉ nên sử dụng dấu ngoặc kép thẳng " (mã ASCII Mỹ 0x22) trong mã/formula.

Thật không may, một số trang web và một số trình duyệt tự động thay đổi dấu ngoặc kép thành những ký tự khác trông giống như dấu ngoặc kép nhưng chúng kỹ thuật khác nhau (sử dụng mã khác nhau). Thực sự có 29 mã khác nhau cho các loại dấu ngoặc kép được sử dụng trên Internet.

Title = This is a text with weird (incorrect) quotation marks; // SAI
Title = "This is a text with correct quotation marks"; // ĐÚNG

Để khắc phục vấn đề, bạn cần thay thế dấu ngoặc kép bằng dấu ngoặc kép thẳng (trên bàn phím Mỹ, bạn nhấn phím dấu phẩy cùng với SHIFT).

Lỗi 58. Đang mong đợi dấu ngoặc đơn kết thúc (`') trong kí tự Literal.

Lỗi này xuất hiện khi bạn đã nhập literal kí tự đơn mà không có dấu ngoặc đơn kết thúc. Các literal kí tự đơn thay thế cho các mã ASCII của các chữ cái, ví dụ như 'a' cho mã ASCII của chữ cái a. Chúng có thể được sử dụng ví dụ trong các câu lệnh AddColumn để hiển thị mã chữ cái đơn thay vì các số:

```
Buy = Cross( C, MA( C, 10 ) );

AddColumn( IIf( Buy, 'B', 'S' ), "Trade", formatChar ); // đúng


AddColumn( IIf( Buy, 'B', 'S ), "Trade", formatChar ); // sai - thiếu dấu ngoặc đơn sau S
```

Lỗi 59: Quá nhiều chỉ số phụ (subscripts)

Lỗi này xảy ra khi bạn cố gắng sử dụng chỉ số phụ ba chiều (triple subscript) trên ma trận hai chiều (two dimensional matrix).

```
mx = Matrix( 2, 3 );
x = mx[ 1 ][ 2 ][ 3 ]; // mx chỉ có 2 chiều, bạn không thể dùng toán tử chỉ số phụ thứ 3 (3rd subscript operator)
```

Lỗi 60. Kích thước Ma trận Yêu cầu Quá Lớn (>2Gb)

Lỗi này xảy ra khi bạn cố gắng tạo ra một ma trận mà đòi hỏi hơn 2GB RAM. Vì giá trị dạng số thực đơn chính xác mất 4 bytes, điều này có nghĩa là bạn chỉ có thể tạo ra các ma trận lên đến 500 triệu phần tử (số hàng * số cột). Hãy lưu ý rằng quá trình 32-bit có bộ nhớ làm việc của nó bị giới hạn đến 2GB cho toàn bộ quá trình, vì vậy khi bạn muốn làm việc với các ma trận lớn, hãy sử dụng phiên bản 64-bit.

mx = Matrix(30000, 20000); // đoạn mã dưới đây là sai vì cố tạo ra một ma trận 30K*20K (600 triệu phần tử - quá lớn):

Lỗi 61. Số lượng ký hiệu định dạng (%) không khớp với số lượng đối số đã truyền.

Lỗi này có nghĩa là bạn đã có chuỗi định dạng không chính xác trong lệnh printf/StrFormat và/hoặc số lượng đối số truyền vào hàm không khớp với số lượng ký hiệu % trong chuỗi định dạng.

Khi bạn gọi printf/StrFormat, chuỗi đầu tiên là chuỗi định dạng, cho biết cách các đối số tiếp theo sẽ được định dạng để xuất. Việc danh sách các đối số tiếp theo khớp với số lượng ký hiệu % trong chuỗi định dạng là rất quan trọng. Trong các trường hợp không khớp, AmiBroker sẽ hiển thị Lỗi 61.

Lỗi 62. Chuỗi định dạng chứa chuỗi không được hỗ trợ. Chỉ có dạng số thực %g, %f và %e mới được hỗ trợ.

Lỗi này có nghĩa là bạn đã có chuỗi định dạng không chính xác trong lệnh gọi printf/StrFormat.

Các hàm printf/StrFormat sử dụng chuỗi tham số đầu tiên như là một chuỗi định dạng quyết định cách hiển thị các đối số tiếp theo. Hiện tại, bạn chỉ có thể truyền số và mảng làm đối số thứ 2 và các đối số tiếp theo vào printf/StrFormat, vì vậy chỉ có thể sử dụng các định dạng số thực, %f, %g, %e. Các định dạng khác không được hỗ trợ và sẽ tạo ra lỗi 62.

```
printf("Close price with 4 decimal digits %.4f\n", Close ); // OK

printf("Close price as integer (without decimal places) %.0f\n", Close ); // OK printf("Close price as integer %d\n", Close ); // WRONG, unsupported %d sequence
```

Nguồn gây lỗi phổ biến khác là quên rằng biểu tượng phần trăm thực sự phải được gõ là %% (hai dấu phần trăm) vì dấu phần trăm đơn là khởi đầu của chuỗi định dạng, không phải là biểu tượng phần trăm thực sự.

```
printf("Actual percent sign is not %." ); // WRONG use % instead of %% printf("Actual percent sign is %%." ); // correct use of %%
```

Lỗi 90. Optimizer Engine được chỉ định không tìm thấy.

Hàm OptimizeSetEngine đã được gọi với tên engine không tồn tại.

```
OptimizerSetEngine("test"); // Lỗi 90 - engine này không tồn tại
```

Lỗi 91. OptimizerSetOption yêu cầu giá trị STRING

Hàm OptimizerSetOption yêu cầu chuỗi cho tùy chọn đối số được đưa ra

OptimizerSetOption("a_string_field", 1); // Lỗi 91 - đối số không chính xác

Lỗi 92. OptimizerSetOption yêu cầu GIÁ TRỊ SỐ (số scalar)

OptimizerSetOption function yêu cầu số cho đối số tùy chọn đã cho

OptimizerSetOption("a_numeric_field", "string"); // Lỗi 91 - đối số không đúng

Lỗi 93. Tùy chọn đã chỉ định không được hỗ trợ / không có sẵn cho bộ tối ưu hóa đã chọn.

Hàm OptimizerSetOption được gọi với tên phương án không được hỗ trợ.

Ví dụ:
```
 OptimizerSetOption("NoSuchOption", 1); // Lỗi 93 - không có phương án nào như vậy trong công cụ bên ngoài
```

Lỗi số 94: Chưa chọn external optimizer. Bạn phải gọi hàm OptimizerSetEngine() trước khi gọi hàm OptimizerSetOption().
Bạn phải gọi hàm OptimizerSetEngine trước khi gọi bất kỳ hàm Optimizer... nào khác.

// OptimizerSetEngine("cmae"); // Không được gọi nhưng NÊN gọi
OptimizerSetOption("runs", 1); // Lỗi số 94 - bạn phải gọi OptimizerSetEngine trước khi gọi OptimizerSetOption

Cảnh báo 501. Gán giá trị trong câu điều kiện. Bạn có chắc chắn ý định sử dụng == thay vì = không?

Bạn đã thực hiện gán giá trị bên trong câu lệnh if/while/for hoặc bên trong hàm IIF. Điều này có thể là một lỗi vì khả năng cao bạn muốn so sánh các số, chứ không phải gán chúng. Hãy nhớ rằng == là kiểm tra sự bằng nhau, chứ không phải =. Khi sử dụng =, bạn không so sánh mà gán một giá trị cho biến. Nếu bạn thực sự muốn gán giá trị bên trong câu điều kiện, hãy sử dụng thêm dấu ngoặc nhọn.

```
if( x = 5 )  // cảnh báo, bạn đang gán 5 cho biến x
 {

 }
 if( x == 5 ) // đúng - bạn có lẽ muốn SO SÁNH (chú ý đến toán tử ==)
 {

}
```

Cảnh báo 502: Bạn đang gọi hàm Plot()/PlotOHLC() hơn 500 lần, điều này rất không hiệu quả. Hãy giảm số lượng lần gọi.

Việc gọi hàm Plot()/PlotOHLC() hơn 500 lần rất không hiệu quả. Bạn cần giảm số lần gọi.

Điều quan trọng là phải hiểu rằng hàm Plot() liên quan đến việc vẽ toàn bộ biểu đồ và điều này tốn kém. Nếu bạn muốn vẽ đường phân đoạn được tạo ra bằng cách sử dụng ví dụ như LineArray, đừng gọi hàm Plot() nhiều lần. Thay vào đó, hãy kết hợp tất cả LineArrays thành một và sử dụng một lần gọi Plot duy nhất.

```
for(i = 0; i < 600; i++)
 {
   x0 = ..;
   x1 = ..;
   y0 = ..;
   y1 = .. ;
   Plot( LineArray( x0, x1, y0, y1 ), "", colorRed ); // KHÔNG làm như thế này!
 }
```

Thay vào đó, hãy sử dụng một lần gọi Plot():

```
 CombinedLine = Null;

for(i = 0; i < 600; i++)
 {
   x0 = ..;
   x1 = ..;
   y0 = ..;
   y1 = .. ;


   La = LineArray( x0, x1, y0, y1 );
   CombinedLine = IIf( IsNull( la ), CombinedLine, la ); // kết hợp các đường
 }

Plot( CombinedLine, "", colorRed ); // Đây mới là CÁCH LÀM ĐÚNG - một lần gọi hàm Plot làm tất cả công việc vẽ biểu đồ
```

Để xem thêm các ví dụ, xem tại http://www.amibroker.org/userkb/2007/04/20/plotting-trade-zigzag-lines/

Cảnh báo 503. Việc sử dụng Ole / Createobject / Createstaticobject không thân thiện với Đa luồng.

Công thức của bạn chứa các lệnh CreateObject và/hoặc CreateStatic. Mặc dù chúng vẫn được hỗ trợ, nhưng hiệu suất của OLE trong ứng dụng đa luồng kém và bạn nên xem xét việc thay thế OLE bằng các lệnh AFL bản địa tương tự.

AB = CreateObject("Broker.Application"); // OLE is slow

AmiBroker hỗ trợ đầy đủ việc gọi các đối tượng OLE từ mức công thức AFL, và nó vẫn an toàn khi sử dụng, nhưng có những lý do kỹ thuật khuyến nghị không sử dụng OLE. Lý do chủ yếu là OLE chậm, đặc biệt khi không được gọi từ luồng "chủ".

OLE được phát triển bởi Microsoft vào thập kỷ 90, ngày 16 bit, đây là công nghệ cũ và nó ngăn cản hiệu quả các luồng chạy ở tốc độ cao nhất vì tất cả các cuộc gọi OLE phải được phục vụ bởi chỉ một luồng giao diện người dùng. Để biết thêm chi tiết xem bài viết sau:

http://blogs.msdn.com/b/oldnewthing/archive/2008/04/24/8420242.aspx

Vì lý do này, nếu có thể, bạn nên hoàn toàn tránh sử dụng OLE / CreateObject trong công thức của mình.

Nếu bạn không làm như vậy, hiệu suất sẽ bị ảnh hưởng. Bất kỳ cuộc gọi nào đến OLE từ luồng công nhân đều gây ra việc đăng tin nhắn vào cửa sổ ẩn của OLE và chờ luồng giao diện người dùng chính của ứng dụng xử lý yêu cầu. Nếu nhiều luồng làm điều tương tự, hiệu suất có thể dễ dàng giảm xuống mức luồng đơn, vì tất cả các cuộc gọi OLE đều được xử lý bởi luồng giao diện người dùng chính.

Bạn nên cân nhắc việc thay thế OLE bằng các lệnh AFL bản địa thực hiện tương tự.

Để biết thêm chi tiết, hãy xem Sử dụng hiệu quả đa luồng.

Lỗi 701. Thiếu Biến Gán Mua/Bán.

Công thức mà bạn đang cố gắng kiểm định hồi quy không chứa các quy tắc Mua và Bán chính xác. Quy tắc Mua và Bán nên được viết dưới dạng bài tập như được hiển thị dưới đây:

Mua = Cross(Close, MA(Close, 50));

Bán = Cross(MA(Close, 50), Close));

Để biết thêm chi tiết, xem Hướng dẫn: Kiểm định hồi quy ý tưởng giao dịch của bạn.

Lỗi 702. Thiếu Gán Biến Short/Cover.

Công thức bạn đang cố gắng backtest không chứa các quy tắc Short và Cover đúng. Các quy tắc Short và Cover nên được viết dưới dạng gán như sau:

Short = Cross(MA(Close, 50), Close));

Cover = Cross(Close, MA(Close, 50));

Nếu bạn không muốn kiểm tra phía giao dịch short, vui lòng đi đến Cài đặt và chọn Long từ combo-box Positions. Điều này sẽ cho phép bạn kiểm tra chỉ phía giao dịch long.

Để biết thêm chi tiết, xem Hướng dẫn: Backtest các ý tưởng giao dịch của bạn.

Lỗi 703. Giao dịch xoay vòng đòi hỏi biến Positionscore.

Khi cố gắng sử dụng chế độ kiểm tra phản hồi Rotational, bạn phải xác định biến PositionScore quyết định cách các biểu tượng được xếp hạng và sắp xếp. Ví dụ về việc gán PositionScore được trình bày bên dưới:

EnableRotationalTrading();

SetOption("WorstRankHeld",5);

PositionSize = -25; // đầu tư 25% quyền chọn vào một bảo mật duy nhất

PositionScore = 50 - RSI(); // ĐIỀU NÀY LÀ BẮT BUỘC cho chế độ xoay vòng

Để biết thêm chi tiết, xem chế độ Giao dịch xoay vòng

Lỗi 704. Trong Giao Dịch Xoay Vòng, Bạn Không Được Sử Dụng Tín Hiệu Buy/Sell/Short/Cover.

Khi cố gắng sử dụng chế độ backtest Xoáy Vòng, bạn phải không sử dụng phép gán biến buy/sell/short/cover.

```
EnableRotationalTrading();

SetOption("WorstRankHeld",5);

PositionSize = -25; // đầu tư 25% vốn vào một loại cổ phiếu

PositionScore = 50 - RSI(); // bắt buộc cho chế độ xoáy vòng

Buy = Cross( C, MA( C, 50 ) ); // SAI - không thể sử dụng trong chế độ xoáy vòng.
```

Nhưng nếu bạn muốn sử dụng Các Tín Hiệu Buy/Sell/Short/Cover và chỉ cần ưu tiên chúng - hãy dùng chế độ backtest thông thường. Đối với điều đó, hãy loại bỏ lời gọi hàm EnableRotationalTrading.

Để biết thêm chi tiết, hãy tham khảo chế độ Giao Dịch Xoáy Vòng.

Lỗi 705: Bạn không thể sử dụng HoldMinBars cùng với Allow Same Bar exit.

Bạn không thể kết hợp tùy chọn HoldMinBars với AllowSameBarExit.

SetOption("AllowSameBarExit", True);

SetOption("HoldMinBars", 5); Cảnh báo: Bạn không thể kết hợp cả hai tùy chọn này, sẽ xảy ra lỗi - hai tùy chọn này không tương thích.

Lỗi 706. Tính năng "Show Arrows" Cần Một "Trade List".

Bạn đã thử sử dụng tính năng "Show arrows" (hoặc nhấp đúp vào danh sách kết quả Analysis) nhưng AmiBroker không thể hiển thị mũi tên thương mại trừ khi bạn chạy backtest với chế độ Report được đặt thành Trade List. Bạn sẽ cần truy cập vào cửa sổ Settings, chuyển sang tab Report, và thay đổi 'Result list shows' thành Trade List (đây là cài đặt mặc định).

Để biết thêm thông tin, xem Tutorial: Sử dụng cửa sổ "New Analysis" và cửa sổ "Settings".

Lỗi 707. Động cơ tối ưu hóa được chọn không tìm thấy hoặc không thể khởi tạo.

Động cơ tối ưu hóa được chỉ định bởi cuộc gọi OptimizerSetEngine() không thể tải hoặc khởi tạo.

Lỗi 708: Số lượng bước tối ưu hóa không đủ cho Quá trình tối ưu hóa thông minh (Non-Exhaustive). Hãy loại bỏ lời gọi hàm OptimizerSetEngine().

Bạn đã cố gắng sử dụng engine tối ưu hóa thông minh (non-exhaustive) cho một không gian tìm kiếm quá nhỏ (quá ít biến tối ưu hóa và / hoặc quá ít bước). Các optimizer không kiệt quệ cần ít nhất 100 bước (tổ hợp tham số). Các vấn đề nhỏ nên được giải quyết bằng cách sử dụng tối ưu hóa kiệt quệ (chỉ cần loại bỏ hoặc chú thích lời gọi OptimizerSetEngine).

```
OptimizerSetEngine("spso");

x = Optimize( "x", 1, 1, 10, 1 ); // Chỉ được tối ưu hoá 10 bước - quá ít cho PSO

Mua = Cross( C, MA( C, x ) );

Bán = 0;
```

Lỗi 709. Không gian tìm kiếm tối ưu hóa quá lớn (hơn 10^19 tổ hợp). Giảm số lượng lệnh Optimize().

Bạn đã cố gắng sử dụng tối ưu hóa trên không gian tìm kiếm lớn hơn 10.000.000.000.000.000.000 tổ hợp bằng phương pháp toàn diện. Bạn cần giảm số lượng lệnh `Optimize()` và/hoặc tăng kích thước bước và/hoặc giảm khoảng min-max của các tham số tối ưu hóa để giảm kích thước không gian tìm kiếm.

x = Optimize("x", 1, 1, 1000, 1); // 10^3

y = Optimize("y", 1, 1, 1000, 1); // 10^6

z = Optimize("z", 1, 1, 1000, 1); // 10^9

i = Optimize("i", 1, 1, 1000, 1); // 10^12

j = Optimize("j", 1, 1, 1000, 1); // 10^15

k = Optimize("k", 1, 1, 1000, 1); // 10^18

m = Optimize("m", 1, 1, 1000, 1); // 10^21 -- QUÁ NHIỀU

```

Bạn có thể thử sử dụng phương pháp thông minh (không toàn diện) thay thế, nhưng hãy nhớ rằng không gian tìm kiếm lớn như vậy rất khó để tìm kiếm và bạn khó có khả năng tìm thấy giá trị cực tiểu toàn cục.
```

Lỗi 710. Ngày Bắt Đầu Phạm Vi Không Được Lớn Hơn Ngày Kết Thúc

Bạn đã thử thực hiện Phân Tích bằng phạm vi "From-to", với Start/From là ngày lớn hơn End/To. Xin vui lòng sửa lại ngày sao cho Start bằng hoặc sớm hơn End.

Lỗi 711. Mục tiêu tối ưu hóa được chọn trong Settings -> Walk Forward không được tìm thấy trong danh sách kết quả

Bạn đã nhập mục tiêu tối ưu hóa trong phần Analysis, Settings, tab Walk Forward nhưng không thể tìm thấy nó trong danh sách kết quả. Thông thường, đây là do mục tiêu tối ưu hóa do người dùng xác định không được tạo ra bởi công thức backtest tùy chỉnh của bạn. Nếu bạn muốn sử dụng custom metrics như mục tiêu tối ưu hóa, bạn phải tính toán và thêm chúng vào báo cáo backtest của bạn như được giải thích trong Tutorial - Sử dụng custom metrics.

Nếu bạn xác định metric của riêng mình, hãy đảm bảo rằng bạn nhập chính xác cùng một tên (phân biệt chữ hoa chữ thường) vào trường "optimization target" trong tab Walk Forward. Nếu có sự không khớp, AmiBroker sẽ không tìm thấy custom metric của bạn và sẽ hiển thị thông báo lỗi này.

Lỗi 712: PointValue phải lớn hơn không. Ký hiệu: ...

Bạn đã cố gắng chạy backtesting đối với ký hiệu mà đã bị đặt sai PointValue. PointValue cần phải lớn hơn không do đó được sử dụng như là bộ nhân cho biến động giá. Khi PointValue được đặt bằng không thì tất cả lợi nhuận/thua lỗ sẽ không hề thay đổi mặc dù giá có biến động và điều này sẽ tạo ra kết quả backtest hoàn toàn vô nghĩa.

Tính Toán Thống Kê Cho Nhiều Chứng Khoán Với Hàm Addtocomposite

Đa số các hàm AFL hoạt động trên giá của một chứng khoán duy nhất. Có hai ngoại lệ cho quy tắc này do hai hàm RelStrength() và Foreign(). Những hàm này cho phép chúng tôi sử dụng giá của chứng khoán khác trong công thức AFL. Mặc dù những hàm này rất hữu dụng cho các việc như so sánh biểu đồ hiệu suất tương đối, nhưng nó không giúp ích nhiều cho những tác vụ yêu cầu giá của tất cả chứng khoán (hoặc một số lượng lớn) vì sẽ phải gõ hàng trăm lệnh Foreign() để làm vậy. Hơn nữa, cách tiếp cận này đòi hỏi phải liệt kê tất cả tên mã chứng khoán trong công thức, khiến công thức bị ràng buộc với thị trường cụ thể. Rõ ràng chúng tôi cần một cách tiếp cận hoàn toàn khác...

Hãy tưởng tượng nếu chúng ta có thể lưu kết quả của các phép tính thực hiện trên một chứng khoán vào một nơi nào đó, sau đó sử dụng những dữ liệu từng phần đó để tạo ra một chỉ số cho nhiều chứng khoán. Có thể có người sẽ nói rằng chúng ta có thể tạo đầu mục khám phá, sau đó xuất kết quả ra file CSV, sau đó tải nó vào Excel và thực hiện các phép tính tại đó. Điều này có thể hoạt động (trong một số trường hợp) nhưng bạn phải đồng ý rằng giải pháp này không tốt.

Đây là lúc mà hàm AddToComposite có thể giúp đỡ.

Ý tưởng chính đằng sau AddToComposite là chúng tôi chạy công thức (sử dụng tính năng Scan) qua một nhóm mã chứng khoán, thực hiện một số phép tính. Chúng tôi sẽ tính một số thống kê cho nhiều chứng khoán và lưu kết quả vào mã chứng khoán nhân tạo tạo ra bằng hàm AddToComposite.

2.3 Giải pháp.

Chìa khóa cho giải pháp là thuật toán sau:

1. Thực hiện một số phép tính AFL thông thường bằng cách sử dụng bất kỳ hàm nào hỗ trợ. Đưa kết quả của việc tính toán vào một trong các trường O, H, L, C, V, I của mã chứng khoán nhân tạo của mình (đặt tên là "~composite" chẳng hạn)
2. Khi quy trình trên được lặp lại trên một nhóm mã, mã chứng khoán tổng hợp của chúng ta sẽ chứa tổng kết quả của tính toán từng mã.

CÚ PHÁP	AddToComposite(array, "ticker", "field", flags = atcFlagDefaults)
TRẢ LẠI	KHÔNG

CHỨC NĂNG	Cho phép bạn tạo các chỉ báo tổng hợp một cách dễ dàng. Tham số: - array: Mảng các giá trị sẽ được thêm vào "field" trong biểu tượng tổng hợp. - ticker: Biểu tượng của tổng hợp. Nên sử dụng ~comp (dấu ngã ở đầu).

	- Các tổng hợp mới được thêm vào sẽ được gán vào nhóm 253 theo mặc định và có tính năng "chỉ sử dụng cơ sở dữ liệu cục bộ" được bật để hoạt động chính xác với các nguồn bên ngoài. Các mã trường có thể có:"C" - Giá đóng cửa, "O" - Giá mở cửa, "H" - Giá cao nhất, "L" - Giá thấp nhất, "V" - Khối lượng, "I" - Lãi mở, "X" - Cập nhật tất cả các trường OHLC cùng một lúc, "1" - Trường aux1, "2", Trường aux2 - flags: Chứa tổng giá trị của các tham số sau đây. **Các cờ (flags) cho chỉ báo tổng hợp:** • **atcFlagResetValues = 1**: Đặt lại giá trị tại đầu quét (được khuyến nghị). • **atcFlagCompositeGroup = 2**: Đưa biểu tượng tổng hợp vào nhóm 253 và LOẠI TRỪ tất cả các biểu tượng khác khỏi nhóm 253 (tránh việc thêm tổng hợp vào tổng hợp). • **atcFlagTimeStamp = 4**: Đưa ngày/giờ quét cuối cùng vào trường FullName. • **atcFlagEnableInBacktest = 8**: Cho phép chạy AddToComposite trong chế độ kiểm tra lại/tối ưu hóa. • **atcFlagEnableInExplore = 16**: Cho phép chạy AddToComposite trong chế độ khám phá. • **atcFlagResetValues = 32**: Đặt lại giá trị tại đầu quét (không cần thiết nếu bạn sử dụng atcFlagDeleteValues). • **atcFlagEnableInPortfolio = 64**: Cho phép chạy AddToComposite trong giai đoạn kiểm tra danh mục tùy chỉnh. • **atcFlagDefaults = 7**: Đây là sự kết hợp của các cờ atcFlagResetValues, atcFlagCompositeGroup và atcFlagTimeStamp.
VÍ DỤ	Hàm AddToComposite cũng phát hiện ngữ cảnh mà nó đang chạy (chỉ hoạt động ở chế độ quét, trừ khi có cờ atcFlagEnableInBacktest hay atcFlagEnableInExplore) và KHÔNG ảnh hưởng tới biểu đồ tổng hợp khi chạy ở chế độ Chỉ số hay Bình luận, do đó, bạn có thể kết hợp quét và chỉ số vào một công thức duy nhất. AddToComposite(MACD() > 0, "~BullMACD", "V"); graph0 = Foreign("~BullMACD", "V"); (Bây giờ bạn có thể sử dụng cùng một công thức trong quét và chỉ số)

Hàm AddToComposite mở ra nhiều ứng dụng thú vị. Dưới đây là một số ví dụ giúp bạn hiểu những gì bạn có thể làm với hàm AddToComposite:

Ví dụ 1:

Giả sử chúng ta muốn tạo chỉ số tùy chỉnh (trung bình giá của nhiều biểu đồ). Với hàm AddToComposite, bạn có thể thực hiện điều này một cách dễ dàng:

```
/* Các câu lệnh AddToComposite dành cho Analysis -> Scan */
/* thêm giá Đóng vào các trường OHLC của chỉ số của chúng ta */
AddToComposite(Close, "~MyIndex", "X" );
/* Thêm một vào trường intest mở (chúng tôi sử dụng trường này như một bộ đếm)*/
AddToComposite( 1, "~MyIndex", "I" );
buy = 0; // yêu cầu của chế độ quét
/* phần này dành cho Indicator */
graph0 = Foreign( "~MyIndex", "C" )/Foreign( "~MyIndex", "I" );
```

Bạn nên sử dụng công thức trên trong chế độ Analysis -> Scan (trên nhóm các ký hiệu theo lựa chọn của bạn). Điều này sẽ tạo ra biểu đồ "~MyIndex" nhân tạo mà sẽ chứa chỉ số của bạn.

Công thức trên chỉ đơn giản là cộng thêm giá Đóng vào các trường OHLC (trường "X" đại diện cho tất cả OHLC) của mã giao dịch nhân tạo của chúng ta ~MyIndex. Thêm vào đó, chúng ta cũng thêm "1" vào trường "I" (mở lượng giao dịch) - hiệu quả là đếm số lượng các mã giao dịch được kiểm tra. Sau này, chúng ta có thể dùng số lượng mã giao dịch này để chia tổng giá bởi số lượng mã giao dịch đã bao gồm (đây chính là dòng cuối cùng của công thức trên).

Ví dụ 2:

Trong ví dụ thứ hai, chúng ta sẽ hướng dẫn cách tính chỉ số chỉ ra số lượng mã giao dịch đáp ứng tiêu chuẩn nhất định. Trong ví dụ này, tiêu chuẩn đó sẽ là RSI nhỏ hơn 30 (điều kiện quá bán), nhưng nó có thể là bất cứ điều gì bạn muốn.

Vậy nên, dòng đầu tiên của công thức của chúng ta sẽ là:

```
values = rsi() < 30;
```

Điều này sẽ lưu trữ "true" trong mảng giá trị cho tất cả các điểm dữ liệu khi RSI nhỏ hơn 30. Sau đó, chúng ta thêm phần AddToComposite thông thường:

```
buy = 0; // không tạo ra tín hiệu

AddToComposite( values, "~MyComposite", "V" );
```

Nếu chúng tôi chạy công thức bằng chức năng "Scan" của cửa sổ Phân tích thì kết quả sẽ là một mã giao dịch nhân tạo "~MyComposite" chứa đầy báo giá. Trường Khối lượng của những báo giá này sẽ chứa số lượng các mã giao dịch đáp ứng tiêu chuẩn của chúng tôi (RSI<30) trong dân số của các mã giao dịch đã quét.

Bạn dễ dàng xem biểu đồ của chỉ số mới này bằng cách sử dụng công thức tùy chỉnh sau:

```
graph0 = foreign("~MyComposite", "V");
```

Giá trị cao của chỉ số này cho thấy hầu hết các mã giao dịch trong nhóm đang được phân tích đều ở trong tình trạng quá bán. Điều này thường xảy ra trước một sự tăng

giá mạnh mẽ của toàn bộ thị trường. Chúng tôi vừa tạo ra máy dò quá bán cho toàn thị trường!

Ví dụ 3:

Trong ví dụ thứ ba, tôi sẽ hướng dẫn bạn cách sử dụng cách thức tương tự để đếm số lượng các vị trí mở của hệ thống giao dịch của bạn. Điều này rất hữu ích nếu bạn muốn biết bạn cần tài khoản lớn như thế nào để giao dịch hệ thống của bạn theo tất cả các giao dịch. Công thức của chúng tôi sẽ rất giống với cái trước đó. Đầu tiên, chúng ta nên có công thức hệ thống giao dịch gốc của chúng ta:

```
 /* Your original formula here */

/* In this example this is simple macd/signal crossover system)

buy = cross( macd(), signal() );

sell = cross( signal(), macd() );

/* the following line uses Flip function to get "1" after the buy signal and reset it back to "0" after sell appears. */

in_trade = flip( buy, sell );

AddToComposite( in_trade, "~OpenPosCount", "V" );
```

Chúng ta sử dụng mã "~OpenPosCount" mô phỏng để lưu trữ kết quả. Lần nữa, chúng ta chỉ nên thực hiện Quét công thức và mã "~OpenPosCount" sẽ trở nên khả dụng.

Sử dụng cú pháp sau:

```
graph0 = foreign( "~OpenPosCount", "V");
```

sau khi chạy bộ thử nghiệm để xem biểu đồ về số lượng vị trí mở của hệ thống của bạn.

2.4 Ghi chú

Để biết thêm chi tiết về composite, hãy xem "Giới thiệu về AddToComposite" (PDF 122KB) của Herman van den Bergen.

Lưu ý rằng để cập nhật bất kỳ mã composite (ví dụ: sau khi thêm/sửa báo giá), bạn nên thực hiện "Scan" lại.

Ý tưởng này đã được trình bày lần đầu tiên trong số 12/2001 của bản tin mẹo AmiBroker. Chân thành cảm ơn ông Dimitris Tsokakis đã có những thảo luận rất xây dựng, cho phép tạo ra và cải tiến ý tưởng này.

Hàm Equity, Biểu đồ Equity Cá nhân và Danh mục

Giới thiệu

Đường equity (vốn) có lẽ là công cụ chẩn đoán tốt nhất cho nhà phát triển hệ thống giao dịch. Trong một biểu đồ, nó cho thấy tổng thể thành công hoặc thất bại của hệ thống đang được thử nghiệm, và tác động của nó đến vốn của bạn.

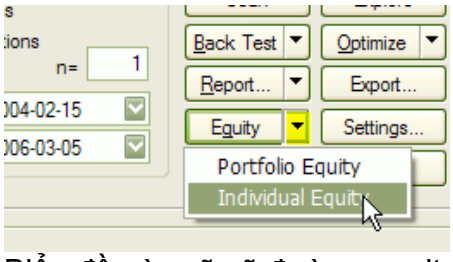
Các số liệu từ báo cáo về mức giảm (drawdown) rất hữu ích, nhưng với một biểu đồ, bạn có thể thấy cách thức hoạt động trước, trong và sau khi xảy ra mức giảm.

Đường do hàm equity tạo ra theo dõi số lượng vốn trong tài khoản của bạn. Ví dụ, nếu bạn kiểm tra lại một hệ thống trong suốt năm qua, bạn sẽ thấy rằng vào đầu năm đó, đường bắt đầu từ số vốn ban đầu của bạn, rồi tăng lên và giảm xuống, vì mỗi giao dịch có lãi hoặc lỗ sẽ làm cho vốn trong tài khoản tăng và giảm. Nó cho thấy số tiền có trong tài khoản của bạn trong suốt thời gian kiểm tra lại, giúp bạn hình dung hiệu suất của hệ thống.

Rõ ràng, bạn không muốn thấy đường đi xuống - điều đó có nghĩa là bạn đã mất tiền. Thông thường, bạn nên điều chỉnh các tham số thử nghiệm của hệ thống để có được đường đi mượt mà, thẳng đứng và đi lên. Điều này có nghĩa là hệ thống của bạn hoạt động ổn định theo thời gian và qua các điều kiện thị trường khác nhau. Một đường đi lên và xuống thường xuyên có nghĩa là hệ thống của bạn chỉ hoạt động tốt dưới những điều kiện nhất định và kém hiệu quả trong những điều kiện khác.

Biểu đồ Equity Cá nhân (single-security)

Để hiển thị biểu đồ Equity của một chứng khoán đơn lẻ, chỉ cần nhấp vào mũi tên thả xuống trên nút "Equity" và chọn "Individual Equity" từ menu trong cửa sổ Phân tích Tự động sau khi thực hiện một lần kiểm tra lại.



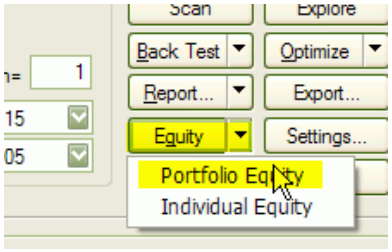
Biểu đồ này sẽ vẽ đường equity cho biểu tượng đang hoạt động và hệ thống vừa kiểm tra lại gần đây. Nếu bạn muốn xem đường Equity cho một biểu tượng khác, chỉ cần chuyển sang biểu tượng đó và đường Equity sẽ được tính toán lại tự động.

Bạn cũng có thể chọn một biểu tượng chưa được đưa vào bộ kiểm tra lại ban đầu và AmiBroker sẽ tính toán chính xác đường equity như thể một cuộc kiểm tra lại thực sự đã được thực hiện trên nó.

QUAN TRỌNG: Biểu đồ equity cá nhân chỉ phản ánh equity của một chứng khoán đơn lẻ và không cho thấy các hiệu ứng ở cấp độ danh mục như việc bỏ lỡ một số giao dịch do đạt giới hạn vị thế mở tối đa hoặc vốn được phân bổ cho các chứng khoán khác. Nó cũng không sử dụng một số chức năng nâng cao chỉ có ở bộ kiểm tra lại cấp độ danh mục.

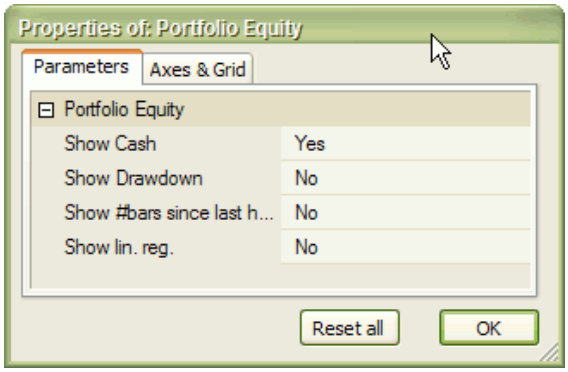
Biểu đồ Equity Cấp độ Danh mục

Để hiển thị biểu đồ equity cấp độ danh mục, chỉ cần nhấp đúp vào nút "Equity" trong cửa sổ Phân tích Tự động hoặc nhấp vào mũi tên thả xuống trên nút "Equity" và chọn "Portfolio Equity" từ menu sau khi thực hiện một lần kiểm tra lại.



Equity cấp độ danh mục đại diện cho equity của toàn bộ danh mục của bạn và phản ánh TẤT CẢ các hiệu ứng thực tế như bỏ lỡ giao dịch do thiếu vốn, đạt số lượng vị thế mở tối đa. Nó cũng phản ánh tất cả các yếu tố như điều chỉnh vị thế vào/ra, hiệu ứng HoldMinBars, phí thoát sớm và bất kỳ tính năng nào khác mà bạn có thể đã sử dụng trong công thức của mình.

Equity cấp độ danh mục cũng hiển thị mặc định số tiền còn lại trong danh mục. Sử dụng cửa sổ Tham số (nhấp chuột phải vào biểu đồ equity và chọn "Parameters" từ menu), bạn có thể bật hiển thị mức giảm (đường equity dưới nước), số lượng thanh kể từ mức cao equity cuối cùng và hồi quy tuyến tính của equity.



Hàm Equity

Hàm Equity() là một bộ kiểm tra lại đơn lẻ cho một chứng khoán. Nó có nhiều ứng dụng thú vị sẽ được mô tả ở đây. Hãy xem xét định nghĩa của hàm Equity:

cấu trú	vốn chủ sở hữu (Falgs = 0, Loại phạm vi = -1, Từ = 0, Đến = 0)
trả về	mảng

hàm	Trả về đường vốn chủ sở hữu dựa trên các quy tắc mua/bán/cắt lỗ/bù đắp, các mảng giá mua/bán/cắt lỗ/bù đắp, áp dụng tất cả các lệnh dừng và tất cả các thiết lập khác của trình kiểm tra lại. cờ - định nghĩa hành vi của hàm vốn chủ sở hữu. 0: (mặc định) vốn chủ sở hữu hoạt động như trong 3.98 - chỉ tính toán mảng vốn chủ sở hữu. 1: hoạt động như 0 nhưng thêm vào việc cập nhật các mảng mua/bán/cắt lỗ/bù đắp, do đó tất cả các tín hiệu thừa được loại bỏ chính xác như cách mà trình kiểm tra lại thực hiện. Ngoài ra, tất cả các lệnh thoát bằng cách dừng cũng được áp dụng, vì vậy giờ đây có thể trực quan hóa các lệnh dừng của ApplyStop(). 2: (nâng cao) hoạt động như 1 nhưng các tín hiệu cập nhật không được di chuyển trở lại vị trí ban đầu nếu độ trễ mua/bán/cắt lỗ/bù đắp được thiết lập trong tùy chọn là khác không. Lưu ý: giá trị cờ này đã được tài liệu hóa nhưng 99% trường hợp không nên sử dụng trong công thức của bạn. Các giá trị khác được dự trữ cho tương lai. RangType - định nghĩa phạm vi báo giá đang được sử dụng: -1: (mặc định) sử dụng phạm vi được thiết lập trong cửa sổ Phân tích Tự động. 0: tất cả các báo giá. 1: n báo giá gần nhất (n được xác định bởi tham số 'từ'). 2: n ngày gần nhất (n được xác định bởi tham số 'từ'). 3: ngày từ/đến. From: định nghĩa ngày bắt đầu (datenum) (khi RangeType == 3) hoặc tham số "n" (khi RangeType == 1 hoặc 2). To: định nghĩa ngày kết thúc (datenum) (khi rangType == 3), nếu không sẽ bị bỏ qua. datenum định nghĩa ngày theo cách tương tự như hàm DateNum() dưới định dạng YYYMMDD, trong đó YYY là (năm - 1900), MM là tháng, DD là ngày. Ngày 31 tháng 12 năm 1999 có detenum là 991231. Ngày 21 tháng 5 năm 2001 có datenum là 1010521. Tất cả các tham số này được đánh giá vào thời điểm gọi hàm vốn chủ sở hữu. Mảng vốn chủ sở hữu hoàn chỉnh được tạo ra một lần. Những thay đổi đối với quy tắc mua/bán/cắt lỗ/bù đắp được thực hiện sau khi gọi sẽ không có hiệu lực. Hàm vốn chủ sở hữu có thể được gọi nhiều lần trong một công thức. mua = quy tắc mua của bạn bán = quy tắc bán của bạn đồ thị 0 = equity()
Ví dụ	mua = quy tắc mua của bạn bán = quy tắc bán của bạn

	đồ thị 0 = equity()

Sử dụng hàm Equity, chúng ta có thể xây dựng chỉ báo vốn chủ sở hữu mà không cần phải chạy trình kiểm tra lại. Chỉ cần nhập công thức sau vào Trình chỉnh sửa công thức và nhấn Áp dụng:

mua = ... quy tắc mua của bạn ...

bán = ... quy tắc bán của bạn ...

đồ thị 0 = Equity();

Hàm Equity() sử dụng các quy tắc mua/bán/cắt lỗ/bù đắp được xác định TRƯỚC khi hàm này được gọi. Toàn bộ quy trình kiểm tra lại được thực hiện bên trong hàm Equity, tạo ra đường vốn chủ sở hữu.

Ghi chú:

1. Đường vốn chủ sở hữu phụ thuộc vào các tham số trong cài đặt Phân tích Tự động.
2. Vốn chủ sở hữu theo dõi Lợi nhuận Lãi suất khi bạn RA NGOÀI thị trường. Nếu bạn không muốn điều này, chỉ cần nhập 0 vào trường "Lãi suất hàng năm" trong cài đặt.
3. Vốn chủ sở hữu cũng theo dõi hoa hồng. Nếu hoa hồng không bằng 0, hoa hồng vào sẽ được tính dựa trên kích thước vị trí và hoa hồng thoát sẽ được tính cho mỗi điểm để mô phỏng số tiền bạn có nếu bạn đóng vị trí tại thanh đã cho.
4. AmiBroker sử dụng mảng SellPrice cho các giao dịch dài hạn và mảng CoverPrice cho các giao dịch ngắn hạn để tính giá trị vốn chủ sở hữu hiện tại.
5. Hàm Equity() chỉ áp dụng cho một chứng khoán duy nhất và không phản ánh các hiệu ứng ở mức danh mục đầu tư như bỏ qua giao dịch, không xử lý một số chức năng nâng cao chỉ có bởi trình kiểm tra lại danh mục. Để biết thêm thông tin, xem bảng đó.

Biểu tượng vốn chủ sở hữu danh mục đặc biệt

Sau khi chạy kiểm tra lại ở cấp độ danh mục, AmiBroker ghi lại các giá trị vốn chủ sở hữu danh mục vào biểu tượng đặc biệt "~~~EQUITY". Điều này cho phép bạn truy cập vốn chủ sở hữu danh mục của lần kiểm tra lại cuối cùng trong công thức của bạn. Để làm điều này, sử dụng hàm Foreign:

PortEquity = Foreign("~~~EQUITY", "C");

Đây chính là những gì mà, ví dụ, công thức biểu đồ vốn chủ sở hữu danh mục tích hợp đang thực hiện.

Tôi có thể tính toán thống kê hệ thống bằng cách sử dụng hàm Equity không?

Có, bạn có thể. Dưới đây là một vài ví dụ về các phép tính do Herman van den Bergen cung cấp:

```
E = Equity();

// Bạn đã giàu như thế nào :-) Plot(Highest(E,"Tôi nên bán",1,3);

// Giảm giá hiện tại của bạn: Plot(Highest(E) - E,"Giảm giá hiện tại",4,1);

// Lợi nhuận giao dịch: LongProfit = IIf(Sell,E - ValueWhen(Buy,E),0); ShortProfit =
IIf(Cover,ValueWhen(Short,E)-E,0); Plot(IIf(Sell,LongProfit,0),"LProfit",8,2+4);
Plot(IIf(Cover,ShortProfit,0),"SProfit",4,2+4);

// Lợi nhuận giao dịch hiện tại: Lastbar = Cum(1) == LastValue(Cum(1));
Plot(IIf(LastBar AND pos,E-ValueWhen(Buy,E),ValueWhen(Short,E) - E),"Lợi nhuận
hiện tại",9,2+4);

// Lợi nhuận hàng ngày: Plot(IIf(pos,E-Ref(E,-1),Ref(E,-1)-E),"Lợi nhuận hàng
ngày",7,2);
```

Làm thế nào để bạn vẽ một Đường vốn chủ sở hữu tổng hợp? Tôi không muốn toàn bộ cơ sở dữ liệu, nhưng muốn xem đường cong dựa trên danh sách theo dõi mà tôi sử dụng cho kiểm tra lại.

Chỉ cần sử dụng biểu đồ vốn chủ sở hữu ở cấp độ danh mục (xem ở trên). Nó đại diện cho vốn chủ sở hữu kiểm tra lại danh mục thực tế, vì vậy nếu bạn chạy kiểm tra lại trên danh sách theo dõi, nó sẽ đại diện cho những gì bạn cần. Bạn cũng có thể viết công thức của riêng mình thực hiện điều tương tự:

```
Plot(Foreign("~~~EQUITY", "C"), "PL Equity", colorRed);
```

Các hàm chấp nhận các khoảng thời gian biến đổi

Các hàm sau đây hỗ trợ các khoảng thời gian biến đổi (trong đó tham số khoảng thời gian có thể là mảng và thay đổi từ thanh này sang thanh khác):

- AMA
- AMA2
- DEMA
- HHV
- HHVBars
- IIR
- LinRegSlope
- LinearReg
- LinRegIntercept
- LLV
- LLVBars
- MA
- PercentRank
- Ref
- StdErr
- StDev
- Sum
- TEMA
- TSF
- WMA

Hàm định nghĩa bởi người dùng, quy trình. Phạm vi cục bộ/toàn cầu

Hàm định nghĩa bởi người dùng cho phép đóng gói mã người dùng thành các mô-đun dễ sử dụng mà có thể được sử dụng ở nhiều nơi mà không cần phải giống nhau nhiều lần.

Hàm phải có một định nghĩa. Định nghĩa hàm bao gồm thân hàm — mã thực thi khi hàm được gọi.

Định nghĩa hàm thiết lập tên và các thuộc tính (hoặc tham số) của hàm. Định nghĩa hàm phải đứng trước cuộc gọi hàm. Định nghĩa bắt đầu bằng từ khóa "function", sau đó là tên hàm, dấu ngoặc mở, sau đó là danh sách tham số tùy chọn và dấu ngoặc đóng. Tiếp theo là thân hàm được bao trong dấu ngoặc nhọn.

Cuộc gọi hàm chuyển quyền điều khiển thực thi từ hàm gọi sang hàm được gọi. Các tham số, nếu có, được truyền theo giá trị cho hàm được gọi. Việc thực thi một lệnh trả về trong hàm được gọi sẽ trả lại quyền điều khiển và có thể là một giá trị cho hàm gọi.

Nếu hàm không bao gồm bất kỳ lệnh trả về nào (không trả về gì), thì chúng ta gọi đó là một quy trình.

Dưới đây là một ví dụ về định nghĩa hàm:

```
// hàm sau đây là bộ lọc bậc 2

function IIR2(input, f0, f1, f2)

{

   result[0] = input[0];

   result[1] = input[1];


   for (i = 2; i < BarCount; i++)

   {

      result[i] = f0 * input[i] +

                  f1 * result[i - 1] + f2 * result[i - 2];

   }

   return result;

}


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

```
Plot(IIR2(Close, 0.2, 1.4, -0.6), "ví dụ hàm", colorRed);
```

Trong đoạn mã này, IIR2 là một hàm do người dùng định nghĩa. input, f0, f1, f2 là các tham số chính thức của hàm. Tại thời điểm gọi hàm, các giá trị của tham số được truyền vào các biến này. Các tham số chính thức hoạt động như các biến cục bộ. Sau đó, chúng ta có result và i là các biến cục bộ. Các biến cục bộ chỉ có thể nhìn thấy bên trong hàm. Nếu bất kỳ hàm nào khác sử dụng cùng tên biến, chúng sẽ không gây ảnh hưởng lẫn nhau.

Do AFL không yêu cầu khai báo biến, quyết định xem biến cụ thể có được coi là cục bộ hay toàn cầu phụ thuộc vào nơi nó ĐƯỢC SỬ DỤNG ĐẦU TIÊN.

- Nếu định danh xuất hiện lần đầu TRONG định nghĩa hàm — thì nó được coi là BIẾN CỤC BỘ.
- Nếu định danh xuất hiện lần đầu NGOÀI định nghĩa hàm — thì nó được coi là BIẾN TOÀN CẦU.

Hành vi mặc định này có thể được ghi đè bằng cách sử dụng từ khóa global và local (được giới thiệu trong phiên bản 4.36) - xem ví dụ 2.

Ví dụ (chú thích):

```
k = 4; // đây là BIẾN TOÀN CẦU

function f(x)

{

   z = 3; // đây là BIẾN CỤC BỘ

   return z * x * k; // 'k' ở đây tham chiếu đến biến toàn cầu k (được sử dụng lần đầu ở trên)

}

z = 5; // đây là BIẾN TOÀN CẦU có cùng tên với biến cục bộ trong hàm f


"The value of z before function call :" + WriteVal(z);

// Ngay cả khi chúng ta gọi hàm

// giá trị của biến toàn cầu z của chúng ta

// không bị ảnh hưởng bởi cuộc gọi hàm vì

// biến toàn cầu z và biến cục bộ z là tách biệt và

// các tham số được truyền theo giá trị (không phải theo tham chiếu)

"The result of f(z) = " + WriteVal(f(z));
```

```
"The value of z after function call is unchanged : " + WriteVal(z);
```

Ví dụ 2: Sử dụng từ khóa local và global để ghi đè quy tắc hiển thị mặc định:

```
VariableA = 5; // biến toàn cầu ngầm định
function Test()
{
    local VariableA; // biến cục bộ rõ ràng với cùng tên như biến toàn cầu
    global VariableB; // biến toàn cầu rõ ràng chưa được định nghĩa trước
    // có thể được sử dụng để trả về nhiều hơn một giá trị từ hàm
    VariableA = 99;
    VariableB = 333;
}
VariableB = 1; // biến toàn cầu
"Trước cuộc gọi hàm";
"VariableA = " + VariableA;
"VariableB = " + VariableB;
Test();
"Sau cuộc gọi hàm";
"VariableA = " + VariableA + " (không bị ảnh hưởng bởi cuộc gọi hàm )";
"VariableB = " + VariableB + " (bị ảnh hưởng bởi cuộc gọi hàm)";
```

Cuối hàm, chúng ta có thể thấy lệnh return được sử dụng để trả về kết quả cho người gọi. Lưu ý rằng hiện tại lệnh trả về phải được đặt ở cuối hàm.

Cũng có thể viết một quy trình (một hàm không trả về gì - void):

```
procedure SinePlotter(Freq, ColorIndex)
{
    pname = "Line" + WriteVal(ColorIndex, 1.0);
    array = sin(Cum(Freq * 0.01));
```

```
    Plot(array, pname, colorRed + ColorIndex, styleThick);
}
for (n = 1; n < 10; n++)
{
    SinePlotter(n / 2 + Cum(0.01), n);
}
```

Lưu ý rằng mặc dù có hai từ khóa riêng biệt là "function" và "procedure", AmiBroker hiện tại đối xử chúng giống nhau (cả hai đều chấp nhận giá trị trả về nhưng không yêu cầu chúng), nhưng trong tương lai, quy tắc có thể được thực thi để chỉ sử dụng lệnh return CÙNG với từ khóa "function". Vì vậy, nên sử dụng từ khóa "function" trong trường hợp hàm của bạn trả về bất kỳ giá trị nào và từ khóa "procedure" trong các trường hợp ngược lại.

Cũng lưu ý rằng việc đệ quy (một hàm gọi chính nó từ bên trong chính nó) HIỆN TẠI KHÔNG được hỗ trợ.

Thông tin thêm

Vui lòng đọc thêm bài viết "Hiểu cách AFL hoạt động" để tìm hiểu thêm.

Công cụ AFL

Phân tích kỹ thuật tự động

Giới thiệu

Kể từ phiên bản 2.5, AmiBroker có các công cụ phân tích kỹ thuật tự động. AmiBroker có thể kiểm tra các điều kiện mua/bán do người dùng xác định, cung cấp cho bạn cái nhìn về tình hình hiện tại trên thị trường. Nó cũng có thể thực hiện kiểm tra hệ thống (mô phỏng) để cho bạn biết hiệu suất của hệ thống giao dịch của bạn. Phiên bản 3.0 của AmiBroker đã giới thiệu ngôn ngữ công thức mới (AFL) cho phép bạn viết không chỉ kiểm tra hệ thống mà còn cả các chỉ báo tùy chỉnh và bình luận của cố vấn guru.

Để làm điều này, bạn cần xác định các quy tắc mua và bán, công thức chỉ báo hoặc bình luận bằng cách sử dụng ngôn ngữ Công thức AmiBroker (AFL), được mô tả dưới đây. Để biết thêm thông tin về cách sử dụng các công cụ phân tích, hãy tham khảo mô tả về cửa sổ Phân tích Tự động, Trình chỉnh sửa Công thức và Cửa sổ Bình luận trong Chương 2.

Ngôn ngữ Công thức AmiBroker (AFL)

AFL được sử dụng để xác định các quy tắc giao dịch và khám phá trong cửa sổ Phân tích Tự động, bình luận tùy chỉnh trong cửa sổ Bình luận Guru và công thức chỉ báo trong cửa sổ Trình chỉnh sửa Công thức.

Tham khảo chi tiết về ngôn ngữ AFL được cung cấp tại đây.

Ví dụ

Dưới đây bạn sẽ tìm thấy một số quy tắc mua và bán đơn giản. Chúng chỉ là các tương đương chính thức của một số quy tắc diễn giải chỉ báo phổ biến nhất. Bạn có thể coi chúng như là điểm khởi đầu để phát triển chiến lược giao dịch của riêng bạn, nhưng trước khi bạn quá hào hứng khi nghĩ rằng mình đã tìm ra "chén thánh" của các hệ thống giao dịch, hãy kiểm tra các điểm sau:

- Kiểm tra hệ thống trên các biểu tượng và khung thời gian khác nhau. Kết quả nên tương tự với dữ liệu gốc đã thử nghiệm.
- Kiểm tra hệ thống trên các loại thị trường khác nhau (ví dụ: thị trường tăng giá, thị trường giảm giá và thị trường đi ngang). Một hệ thống tốt nên hoạt động trong tất cả các loại thị trường, vì bạn sẽ không luôn biết khi nào một thị trường chuyển từ xu hướng sang giao dịch hoặc ngược lại.
- Chú ý kỹ đến số lượng giao dịch được tạo ra bởi một mô phỏng giao dịch. Nếu có một số lượng lớn giao dịch và lợi nhuận lớn, hãy chắc chắn rằng bạn đã chỉ định hoa hồng thực tế. Kết quả của bài kiểm tra có thể khác biệt nhiều khi tính toán hoa hồng.

```
buy = cross(macd(), 0);

sell = cross(0, macd());
```

```
buy = cross(ema(close, 9), ema(close, 15));

sell = cross(ema(close, 15), ema(close, 9));


buy = cross(rsi(), 30);

sell = cross(70, rsi());


buy = cross(ultimate(), 50);

sell = cross(50, ultimate());
```

Cửa sổ phân tích tự động

Cửa sổ phân tích tự động cho phép bạn kiểm tra các báo giá của mình dựa trên các quy tắc mua/bán đã định nghĩa. AmiBroker có thể tạo ra báo cáo cho bạn biết liệu có tín hiệu mua/bán xảy ra trên biểu tượng nhất định trong khoảng thời gian đã chỉ định hay không. Nó cũng có thể mô phỏng giao dịch, giúp bạn có cái nhìn về hiệu suất của hệ thống của bạn.

Ở phần trên cùng của cửa sổ, bạn có thể thấy một trường nhập văn bản. Trong trường này, bạn nên nhập các quy tắc mua và bán. Những quy tắc này là các câu lệnh gán được viết bằng ngôn ngữ riêng của AmiBroker. Bạn có thể tìm thấy mô tả về ngôn ngữ này trong hướng dẫn tham khảo AFL.

Để mọi thứ hoạt động, bạn cần viết hai câu lệnh gán (một cho quy tắc mua, một cho quy tắc bán), ví dụ:

```
buy = cross(macd(), 0);

sell = cross(0, macd());
```

Cửa sổ phân tích tự động cũng cho phép bạn tối ưu hóa hệ thống giao dịch của mình và thực hiện các khám phá sâu hơn. Xem thêm: mô tả chi tiết về các điều khiển của cửa sổ Phân tích Tự động.

Trình chỉnh sửa công thức

Trình chỉnh sửa công thức cho phép bạn viết các công thức để sử dụng làm chỉ báo hoặc trong cửa sổ Phân tích Tự động. Thêm thông tin về điều này có thể được tìm thấy ở đây.

Cửa sổ Bình luận Cố vấn Guru

Cửa sổ Bình luận cho phép bạn xem các mô tả bằng văn bản về tình hình kỹ thuật thực tế trên thị trường đã cho. Các bình luận được tạo ra bằng cách sử dụng các công thức được viết bằng ngôn ngữ công thức riêng của AmiBroker. Bạn có thể tìm

thấy mô tả về ngôn ngữ này trong Hướng dẫn Tham khảo Ngôn ngữ Công thức AmiBroker.

Hơn nữa, tính năng Bình luận cũng cung cấp cho bạn biểu diễn đồ họa của các tín hiệu mua và bán bằng cách đặt các dấu (mũi tên) trên biểu đồ giá.

LƯU Ý: Sử dụng các công thức bình luận phức tạp có thể khiến thời gian thực thi kéo dài.

Xem thêm: mô tả chi tiết về các điều khiển của cửa sổ Bình luận Cố vấn Guru.

Máy chủ kịch bản AFL

LƯU Ý QUAN TRỌNG: Kể từ phiên bản 4.40, với sự ra mắt của các câu lệnh vòng lặp và điều khiển luồng như if-else và while, tầm quan trọng của kịch bản đã giảm đi rất nhiều. Hiện tại, hầu hết các tác vụ cần kịch bản trong các phiên bản trước có thể được xử lý bằng AFL gốc. Hơn nữa, vòng lặp AFL nhanh hơn từ 3 đến 6 lần so với JScript/VBScript.

Cơ bản

Máy chủ kịch bản AFL là một giao diện giữa động cơ AFL và các động cơ JScript/VBScript (còn gọi là công nghệ Kịch bản Hoạt động) có sẵn như một phần của nền tảng Công cụ & Công nghệ Internet do Microsoft cung cấp. Nó cho phép bạn xây dựng các công thức có các phần trong mã AFL và các phần trong JScript/VBScript.

Yêu cầu

- AmiBroker 3.59 hoặc cao hơn
- Động cơ Microsoft JScript/VBScript đã được cài đặt

Các động cơ JScript/VBScript được cài đặt cùng với Windows.

Tài liệu JScript/VBScript có thể được tìm thấy trên trang kịch bản chính thức tại: msdn.microsoft.com/scripting/

Bật Máy chủ Kịch bản AFL

Nếu bạn muốn sử dụng kịch bản trong các công thức của mình, bạn phải gọi hàm EnableScript() ở đầu công thức của bạn. Hàm này nhận một tham số đầu vào - tên động cơ:

```
EnableScript("jscript");
```

hoặc

```
EnableScript("vbscript");
```

Từ đó, bạn sẽ có thể tích hợp các phần viết bằng ngôn ngữ kịch bản vào các công thức của mình. Đầu và cuối của kịch bản phải được đánh dấu bằng các ký tự <% và %>, như trong ví dụ dưới đây:

```
EnableScript("vbscript");

// Các câu lệnh AFL "bình thường"

buy = cross(macd(), 0);

sell = cross(0, macd());

<%
```

```
..... mã kịch bản của bạn ở đây.....

%>

// Các câu lệnh AFL "bình thường"

buy = ExRem(buy, sell);
```

Sử dụng biến

Hiện tại, cách duy nhất để trao đổi thông tin giữa phần AFL "bình thường" và phần kịch bản là sử dụng biến. Máy chủ kịch bản AFL cung cấp một đối tượng (được định nghĩa trước, không cần tạo/khởi tạo) gọi là AFL.

Đối tượng AFL có một thuộc tính (mặc định) được tham số hóa gọi là Var(varname) có thể được sử dụng để truy cập các biến AFL từ phía kịch bản:

```
// Ví dụ trong VBScript

<%


buyarrayfromscript = AFL.Var("buy")

' lấy mảng mua từ AFL vào biến định nghĩa trong kịch bản


AFL.Var("buy") = buyarrayfromscript

' thiết lập mảng mua trong AFL từ biến định nghĩa trong kịch bản

%>
```

Vì Var là thuộc tính mặc định, bạn có thể bỏ qua tên của nó và viết đơn giản là AFL(varname) như trong ví dụ dưới đây:

```
<%

buyarrayfromscript = AFL("buy")

' lấy mảng mua từ AFL vào biến định nghĩa trong kịch bản

AFL("buy") = buyarrayfromscript

' thiết lập mảng mua trong AFL từ biến định nghĩa trong kịch bản

%>
```

Trong AFL có ba kiểu dữ liệu khả thi: mảng (của số thực), số (số thực) và chuỗi. Các động cơ VBScript và JScript sử dụng kiểu dữ liệu variant có thể chứa bất kỳ loại biến nào, bao gồm ba loại được sử dụng bởi AFL. Như trong AFL, bạn không khai báo biến trong các ngôn ngữ kịch bản, kiểu sẽ được xác định bởi phép gán đầu tiên. Trong trường hợp của VBScript, bạn có thể lấy/thiết lập các biến AFL của bất kỳ kiểu hỗ trợ nào bằng cú pháp như trên. Nhưng trong JScript, do sự khác biệt cơ bản trong cách xử lý mảng trong JScript (các phần tử mảng trong JScript được triển khai như các thuộc tính động của một đối tượng mảng), bạn cần sử dụng mã sau để lấy giá trị của mảng AFL vào mảng JScript:

```
// Ví dụ trong JScript

<%

function GetArray(name) {

   return VBArray(AFL(name)).toArray();

}

myJScriptArray = GetArray("close");

%>
```

Hàm GetArray() ở trên giúp dễ dàng chuyển đổi mảng kiểu tự động thành mảng JScript. Ví dụ này cũng cho thấy cách định nghĩa và sử dụng các hàm trong JScript.

Việc gán các biến AFL từ các mảng phía kịch bản đơn giản hơn nhiều, máy chủ kịch bản AFL phát hiện các mảng JScript và có thể lấy nội dung của chúng trực tiếp:

```
// Ví dụ trong JScript

<%

AFL("buy") = myJScriptBuyArray;

%>
```

Tất cả các kiểu dữ liệu khác đều được xử lý tương tự trong JScript và VBScript:

```
// Ví dụ trong VBScript

ticker = name();

<%

tickerstring = AFL("ticker");

AFL("ticker") = "new name";

%>
```

hoặc trong JScript:

```
// Ví dụ trong JScript

ticker = name();

<%

tickerstring = AFL("ticker");

AFL("ticker") = "new name";

%>
```

Lặp qua các mảng

Một trong những tác vụ cơ bản mà mọi người có thể làm là lặp qua mảng. Trong VBScript, điều này có thể được thực hiện bằng câu lệnh For..To..Next, trong JScript bằng câu lệnh for(;;). Cả hai cấu trúc này đều cần biết kích thước mảng hoặc số phần tử trong mảng. Trong VBScript, bạn nên sử dụng hàm UBound(array) để lấy giới hạn trên của mảng, trong JScript, bạn chỉ cần sử dụng thuộc tính length của mảng. Các ví dụ sau cho thấy điều này. (Xin nhớ rằng trong cả VBScript và JScript, mảng đều bắt đầu từ 0.)

```
// Ví dụ trong VBScript

<%

myArray = AFL("close")


sum = 0

for i = 0 to UBound(myArray)

    sum = sum + myArray(i)

next

%>
```

hoặc trong JScript:

```
// Ví dụ trong JScript

<%

function GetArray(name) {

    return VBArray(AFL(name)).toArray();

}
```

```
myArray = GetArray("close");
sum = 0;
for (i = 0; i < myArray.length; i++) {
    sum += myArray[i];
}
%>
```

Ví dụ

a) Ví dụ chỉ báo - Trung bình động lũy thừa:

```
EnableScript("jscript");
<%
close = VBArray(AFL("close")).toArray();
output = new Array();
// khởi tạo phần tử đầu tiên
output[0] = close[0];

// thực hiện vòng lặp tính toán trung bình động lũy thừa
factor = 0.05;

for (i = 1; i < close.length; i++) {
    output[i] = factor * close[i] + (1 - factor) * output[i - 1];
}

AFL.Var("graph0") = close;
AFL.Var("graph1") = output;
%>
```

```
WriteVal(graph1);
```

b) Ví dụ về dừng mục tiêu lợi nhuận

Dưới đây là ví dụ về công thức thực hiện dừng mục tiêu lợi nhuận ở mức 10% cố định từ giá mua. Lưu ý rằng điều kiện mua được thỏa mãn khi giá đạt đỉnh mới, vì vậy nó xảy ra nhiều lần sau khi mua ban đầu. Do đó, ValueWhen(buy, close) không thể cung cấp giá mua ban đầu và quy tắc giao dịch như vậy không thể được triển khai trong AFL. Nhưng, với kịch bản, không có vấn đề gì...

```
EnableScript("VBScript");


hh = HHV(close, 250);


mua khi giá đạt đỉnh mới

buy = Close == HHV(close, 250);


đảm bảo rằng sell là một mảng,

sell = 0 sẽ thiết lập loại thành số

sell = buy;


<%

close = AFL("close");

buy = AFL("buy");

sell = AFL("sell");


biến này giữ giá mua cuối cùng

nếu nó bằng 0 có nghĩa là không có giao dịch nào mở

lastbuyprice = 0;

 lặp qua mảng
```

```
for (i = 0 to UBound(close)) {
    sell(i) = 0;

  Xóa tín hiệu Mua nếu giao dịch đã được khởi động
    if (lastbuyprice > 0) {
        buy(i) = 0;
    end if

  Nếu không có giao dịch mở và tín hiệu mua xảy ra, lấy giá mua
if (lastbuyprice = 0) AND (buy(i) = 1) then
    lastbuyprice = close(i)
end if
```

Nếu giao dịch đang mở và điều kiện bán hợp lệ, tạo tín hiệu bán và đóng giao dịch

```
if (lastbuyprice > 0) AND (close(i) > (1.1 * lastbuyprice)) then
    sell(i) = 1
    lastbuyprice = 0
end if
next
AFL("buy") = buy
AFL("sell") = sell
```

Thông tin thêm

Nhiều ví dụ về kịch bản có sẵn trong thư viện trực tuyến AFL tại: http://www.amibroker.com/library/list.php

Nếu bạn có bất kỳ câu hỏi, ý kiến hoặc gợi ý nào khác, vui lòng sử dụng trang hỗ trợ khách hàng tại http://www.amibroker.com/support.html. Xin lưu ý rằng lập trình kịch bản AFL là một chủ đề khá nâng cao, và bạn nên thử nghiệm một chút với AFL trước khi đi sâu vào lập trình kịch bản.

Hỗ trợ Component Object Model (COM) trong AFL

Giới thiệu

Component Object Model (COM) là công nghệ định nghĩa và triển khai các cơ chế cho phép các thành phần phần mềm, như ứng dụng, đối tượng dữ liệu, điều khiển và dịch vụ, tương tác với nhau như các đối tượng. Hỗ trợ COM trong AFL được giới thiệu trong phiên bản 3.75beta, cho phép tạo các phiên bản của các đối tượng COM và gọi các hàm (phương thức) được đối tượng đó cung cấp.

Đối tượng COM có thể được tạo ra bằng nhiều ngôn ngữ khác nhau, bao gồm C/C++, Visual Basic, Delphi, v.v. Điều này cho phép bạn viết các phần của chỉ báo, hệ thống, khám phá và bình luận của bạn bằng ngôn ngữ mà bạn chọn và chạy chúng với tốc độ mã biên dịch đầy đủ.

Các động cơ kịch bản được sử dụng bởi AFL scripting host (JScript và VBScript) cũng tự hiển thị như các đối tượng COM. Hỗ trợ COM trong AFL hiện cho phép bạn gọi các hàm được định nghĩa trong phần kịch bản trực tiếp từ AFL mà không cần sử dụng biến để truyền dữ liệu vào và lấy dữ liệu ra từ kịch bản.

Gọi các hàm được định nghĩa trong kịch bản

Cho đến phiên bản 3.75, cách duy nhất để trao đổi thông tin giữa AFL và kịch bản là sử dụng biến. Giả sử chúng ta cần một hàm tính toán bộ lọc IIR bậc hai:

y[n]=f0∗x[n]+f1∗y[n−1]+f2∗y[n−2]y[n] = f0 * x[n] + f1 * y[n-1] + f2 * y[n-2]y[n]=f0∗x[n]+f1∗y[n−1]+f2∗y[n−2]

Hàm này sẽ hoạt động tốt hơn so với bộ lọc trung bình di động thông thường.

Với "cách cũ", mã sẽ trông như sau:

```
EnableScript("jscript");

x = (High + Low) / 2;

f0 = 0.2;

f1 = 1.2;

f2 = -0.4;


<%

x = VBArray(AFL("x")).toArray();

f0 = AFL("f0");

f1 = AFL("f1");

f2 = AFL("f2");
```

```
y = new Array();
y[0] = x[0];
y[1] = x[1];

for (i = 2; i < x.length; i++) {
    y[i] = f0 * x[i] + f1 * y[i - 1] + f2 * y[i - 2];
}

AFL.Var("y") = y;
%>

Graph0 = Close;
Graph0Style = 64;
Graph1 = y;
```

Cách này có thể không hiệu quả nếu cần sử dụng hàm nhiều lần. Phương pháp tốt hơn là định nghĩa hàm có thể gọi từ nhiều nơi mà không cần lặp lại mã:

```
EnableScript("jscript");

<%
function IIR2(x, f0, f1, f2) {
    x = VBArray(x).toArray();
    y = new Array();
    y[0] = x[0];
    y[1] = x[1];

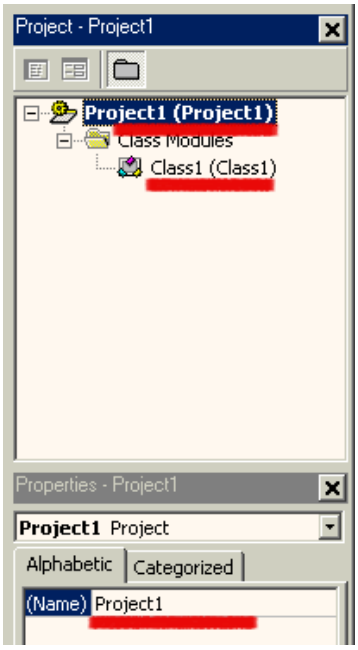
    for (i = 2; i < x.length; i++) {
```

```
        y[i] = f0 * x[i] + f1 * y[i - 1] + f2 * y[i - 2];
    }
    return y;
}
%>
```

Để gọi hàm này từ AFL, bạn có thể sử dụng hàm mới GetScriptObject() để truy cập vào động cơ kịch bản.

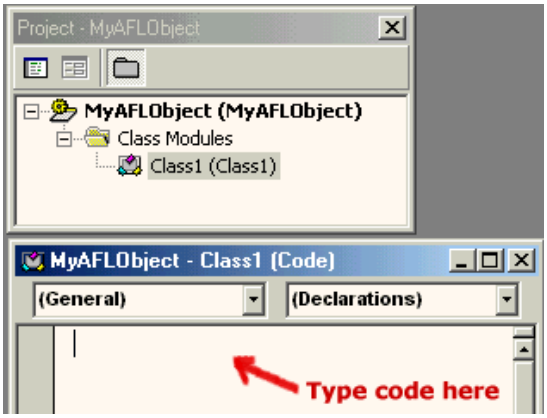
```
script = GetScriptObject();
Graph0 = script.IIR2((High + Low) / 2, 0.2, 1.2, -0.4);
Graph1 = script.IIR2((Open + Close) / 2, 0.2, 1.0, -0.2);
```

Sử dụng các đối tượng COM/ActiveX bên ngoài trong AFL



Tương tự, bạn có thể gọi các hàm trong các đối tượng COM bên ngoài trực tiếp từ công thức AFL. Dưới đây là cách viết ActiveX bên ngoài trong Visual Basic:

1. Chạy Visual Basic.
2. Trong hộp thoại "Dự án mới", chọn "ActiveX DLL" để tạo "Project1".
3. Đổi tên "Project1" thành tên có ý nghĩa hơn, ví dụ "MyAFLObject".
4. Nhấp đúp vào "Class1" trong cây dự án để mở cửa sổ mã.



Public Function IIR2(InputArray() As Variant, f0 As Variant, f1 As Variant, f2 As Variant) As Variant

```
    Dim Result()
    ReDim Result(UBound(InputArray)) ' Kích thước mảng Result theo InputArray
    Result(0) = InputArray(0)
    Result(1) = InputArray(1)

    For i = 2 To UBound(InputArray)
        Result(i) = f0 * InputArray(i) + f1 * Result(i - 1) + f2 * Result(i - 2)
    Next
    IIR2 = Result
End Function
```

Kết luận

Việc giới thiệu hỗ trợ COM trong AFL mang đến sức mạnh mới cho AFL và AmiBroker. Giờ đây, bạn có thể viết các chỉ báo, hệ thống giao dịch, khám phá và bình luận sử dụng các hàm tùy chỉnh dễ dàng tạo ra bằng ngôn ngữ kịch bản hoặc môi trường phát triển đầy đủ như Visual Basic, Borland Delphi, C++ Builder, Visual C++, và nhiều ngôn ngữ khác. Sử dụng môi trường phát triển tích hợp giúp quá trình gỡ lỗi, kiểm tra và phát triển nhanh chóng hơn. Mã biên dịch cũng chạy nhanh hơn nhiều so với mã kịch bản hoặc AFL đã diễn giải.

C/C++ lập trình viên có thể viết các plugin DLL không sử dụng công nghệ COM. Các plugin DLL có tính năng bổ sung như gọi lại các hàm tích hợp của AFL và hỗ trợ tô màu cú pháp tự động cho các hàm do plugin cung cấp. Chủ đề này được đề cập trong Bộ phát triển AmiBroker có sẵn từ khu vực thành viên của trang web AmiBroker.

Plugin trong AFL

Giới thiệu

Phần này mô tả các DLL plugin thông thường. Nếu bạn quan tâm đến các plugin ActiveX, hãy xem phần "Hỗ trợ COM trong AFL".

Giao diện Plugin

Giao diện Plugin của AmiBroker cho phép gọi một mô-đun bên ngoài (thư viện DLL) trực tiếp từ AFL. Thư viện này có thể cung cấp nhiều hàm có thể được sử dụng trong hệ thống giao dịch, chỉ báo, bình luận, quét và khám phá của bạn. DLL plugin có thể được tạo ra bằng bất kỳ trình biên dịch C/C++, Delphi và các ngôn ngữ khác hỗ trợ tạo DLL thông thường. Vì mã của plugin được biên dịch thành mã máy của bộ xử lý gốc, nó chạy nhanh hơn nhiều lần so với AFL. Bạn có thể sử dụng toàn bộ sức mạnh của ngôn ngữ C/C++ (hoặc ngôn ngữ khác) để xây dựng các hàm phức tạp một cách dễ dàng.

Ngoài việc cung cấp các hàm cho AFL, các DLL plugin còn có khả năng gọi lại các hàm tích hợp của AFL, truy xuất và thiết lập trực tiếp các biến AFL và hỗ trợ tô màu cú pháp tự động cho các hàm được cung cấp bởi plugin.

Mô tả chi tiết về giao diện plugin và mã mẫu của DLL plugin được bao gồm trong Bộ phát triển AmiBroker (ADK). ADK chỉ có sẵn cho người dùng đã đăng ký (có thể tải xuống từ khu vực thành viên).

Tải về các Plugin của bên thứ ba

Các plugin của bên thứ ba miễn phí có sẵn để tải xuống từ http://www.amibroker.net/3rdparty.php.

Sử dụng các Plugin bên thứ ba

Để sử dụng DLL plugin bên thứ ba, chỉ cần sao chép tệp DLL vào thư mục Plugins trong thư mục AmiBroker. Sau đó, chạy AmiBroker. Chọn menu Tools -> Plugins. Trong cửa sổ Plugins, bạn sẽ thấy danh sách tất cả các DLL plugin đã tải. Nếu AmiBroker đang chạy khi bạn sao chép DLL, bạn nên nhấp vào nút "Unload" và sau đó nhấp vào nút "Load". Điều này sẽ buộc quét lại thư mục Plugins và tải các DLL.

Khi DLL plugin được tải, các hàm mới do DLL này cung cấp sẽ trở nên khả dụng cho tất cả các công thức AFL của bạn. Để biết danh sách các hàm do plugin cung cấp, bạn nên tham khảo tài liệu của chính plugin đó.

Lưu ý quan trọng

AmiBroker không đảm bảo về các tính năng và hiệu suất của các plugin bên thứ ba không được chứng nhận. Cụ thể, một số plugin có thể gây ra sự không ổn định hoặc thậm chí là sự cố. Việc sử dụng toàn bộ các plugin bên thứ ba không được chứng nhận là hoàn toàn thuộc về rủi ro của bạn.

Những Lỗi Lập Trình Thông Thường trong AFL

Tài liệu này trình bày các lỗi và vấn đề thường gặp mà người dùng gặp phải khi viết công thức tùy chỉnh. Vui lòng đọc kỹ để tránh mắc phải những lỗi tương tự.

• = (gan) vs == (kiểm tra sự bằng nhau)

• Sử dụng dấu ngoặc

• Hàm IIf

• IIf dành cho mảng, WriteIf dành cho chuỗi

• Câu lệnh if-else cần giá trị boolean (hoặc biểu thức số duy nhất), không phải mảng

• Barcount vs BarIndex()

• TimeFrameExpand( ) cần thiết để khớp dữ liệu với khung thời gian gốc

---

= (gan) vs == (kiểm tra sự bằng nhau)

Có hai toán tử trông giống nhau nhưng hoàn toàn khác nhau trong AFL.

- **=** là toán tử gán biến
- **==** là toán tử kiểm tra sự bằng nhau

VÍ DỤ

Mã sai:

```
result = IIf( Variable = 10 , High, Low ); // SAI
```

Nếu bạn muốn kiểm tra xem biến có bằng 10 hay không, bạn PHẢI sử dụng “==”.

Mã đúng:

```
result = IIf( Variable == 10 , High, Low ); // ĐÚNG
```

Sử dụng dấu ngoặc

Dấu ngoặc có thể được sử dụng để kiểm soát thứ tự thực hiện (thứ tự mà các toán tử được tính toán). AmiBroker luôn thực hiện các phép toán trong dấu ngoặc bên trong đầu tiên. Để tìm hiểu thứ tự thực hiện khi không sử dụng dấu ngoặc, hãy truy cập: AmiBroker Language Guide.

VÍ DỤ:

Tôi muốn mua khi nào đó Close cao hơn trung bình di động 10 kỳ của nó hoặc Close cao hơn ít nhất 10% so với giá đóng cửa của ngày hôm qua, nhưng việc mua chỉ nên áp dụng khi Khối lượng Hiện tại cao hơn trung bình di động 10 kỳ của nó. Tuy nhiên, tôi nhận được tín hiệu Mua vào những ngày mà Khối lượng thấp hơn MA(Volume,10). Tại sao?

```
Buy = Close > MA(Close, 10) OR Close == 1.1 * Ref(Close, -1) AND Volume >
MA(Volume, 10);
```

Giải pháp là thêm dấu ngoặc, nếu không hệ thống sẽ mua khi điều kiện Close > MA(Close,10) được thỏa mãn (hoặc cả Close == 1.1*Ref(Close,-1) và Volume > MA(Volume,10) đều thỏa mãn).

```
Buy = (Close > MA(Close, 10) OR Close == 1.1 * Ref(Close, -1)) AND Volume >
MA(Volume, 10);
```

Hàm IIf

Hàm IIf( ) được sử dụng để tạo các gán điều kiện.

```
variable = IIf(EXPRESSION, TRUE_PART, FALSE_PART);
```

Câu lệnh "IIf" trên có nghĩa là: Đối với mỗi thanh, nếu EXPRESSION là đúng, gán TRUE_PART cho biến, nếu không (khi EXPRESSION sai) gán FALSE_PART.

VÍ DỤ

Mã sai:

```
IIf(Close > 10, result = 7, result = 9); // SAI
```

Mã đúng:

```
result = IIf(Close > 10, 7, 9); // ĐÚNG
```

IIf dành cho mảng, WriteIf dành cho chuỗi

Hàm IIf nên được sử dụng để xử lý mảng; nếu bạn cần hàm chuỗi điều kiện, hãy sử dụng WriteIf thay thế.

VÍ DỤ

Mã sai:

```
variable = IIf(Condition, "Text 1", "Text 2"); // SAI
```

Hàm IIf( ) trả về mảng, KHÔNG phải là CHUỖI, vì vậy không thể gán văn bản cho biến bằng cách sử dụng IIF. Sử dụng hàm WriteIf( ) thay vào đó:

Mã đúng:

```
variable = WriteIf(condition, "Text 1", "Text 2"); // ĐÚNG
```

Lưu ý rằng hàm WriteIf chỉ trả về một CHUỖI đơn, không phải mảng các chuỗi, vì vậy chỉ giá trị đã chọn được sử dụng để đánh giá.

Câu lệnh if-else cần giá trị boolean (hoặc biểu thức số duy nhất), không phải mảng

Từ khóa if thực hiện statement1 nếu biểu thức đúng (khác không); nếu else xuất hiện và biểu thức sai (bằng không), nó thực hiện statement2. Sau khi thực hiện statement1 hoặc statement2, điều khiển chuyển sang câu lệnh tiếp theo. Biểu thức phải là loại boolean (Đúng/Sai) (do đó không thể là MẢNG vì sẽ không có cách nào để quyết định có thực hiện statement1 hay không, nếu ví dụ mảng là:

```
[True, True, False, ..., False, True]

if(expression) statement1

else

statement2
```

VÍ DỤ

```
if(Close > Open) // SAI
    Color = colorGreen; // statement 1
else
    Color = colorRed; // statement 2
Plot(Close, "Colored Price", Color, styleCandle);
```

Ví dụ trên là sai, vì cả Open và Close đều là mảng và biểu thức như Close > Open cũng là MẢNG. Giải pháp phụ thuộc vào câu lệnh. Có thể thực hiện theo từng thanh, sử dụng vòng lặp FOR:

```
for(i = 0; i < BarCount; i++)
{
    if(Close[i] > Open[i]) // ĐÚNG
```

```
        Color[i] = colorGreen;
    else
        Color[i] = colorRed;
}
Plot(Close, "Colored Price", Color, styleCandle);
```

Cũng có thể sử dụng hàm IIf( ) trong trường hợp này:

```
Color = IIf(Close > Open, colorGreen, colorRed); // CŨNG ĐÚNG - làm việc trực tiếp trên các mảng
Plot(Close, "Colored Price", Color, styleCandle);
```

Barcount vs BarIndex()

Có sự khác biệt cơ bản giữa BarCount và BarIndex(). BarCount là một biến số học chỉ giữ một số (số lượng phần tử trong mảng). Ngược lại, BarIndex() là một hàm trả về MẢNG đại diện cho chỉ số liên tiếp của mỗi thanh.

VÍ DỤ

Mã sai:

```
for(i = 0; i < BarIndex(); i++) // SAI
{
    // công thức của bạn
}
```

Không được phép sử dụng MẢNG bên trong vòng lặp for, và BarIndex() trả về MẢNG. Đó là lý do tại sao cần thay đổi công thức.

Mã đúng:

```
for(i = 0; i < BarCount; i++) // ĐÚNG
{
```

```
    // công thức của bạn
}
```

TimeFrameExpand( ) cần thiết để khớp dữ liệu với khung thời gian gốc

Hàm TimeFrameSet( ) thay thế các mảng giá/khối lượng hiện tại: open, high, low, close, volume, openint, avg bằng các thanh nén thời gian của khoảng thời gian chỉ định khi bạn chuyển sang khung thời gian khác, tất cả các phép toán và chỉ số tích hợp hoạt động trên khung thời gian đã chọn. Để quay lại khoảng thời gian gốc, gọi hàm TimeFrameRestore( ). Hàm TimeFrameExpand( ) được sử dụng để giải nén các biến mảng được tạo ra trong khung thời gian khác. Việc giải nén là cần thiết để hiển thị và sử dụng đúng cách mảng được tạo ra trong khung thời gian khác.

VÍ DỤ

Mã sai:

```
TimeFrameSet(inWeekly);

MA14_Weekly = MA(Close, 14);

TimeFrameRestore();

Buy = Cross(Close, MA14_Weekly); // SAI - Close và MA15_Weekly sử dụng các tỷ lệ thời gian khác nhau
```

Công thức trên là sai, vì biến MA14_Weekly nên được GIẢI NÉN để khớp với khung thời gian gốc. Nội dung đúng nên là:

Mã đúng:

```
TimeFrameSet(inWeekly);

MA14_Weekly = MA(Close, 14);

TimeFrameRestore();

Buy = Cross(Close, TimeFrameExpand(MA14_Weekly, inWeekly)); // ĐÚNG, MA hàng tuần đã giải nén có thể khớp với giá đóng cửa hàng ngày
```

VÍ DỤ 2:

Mã sai:

```
TimeFrameSet(inWeekly);
```

```
MA14_Weekly = MA(Close, 14);

TimeFrameRestore();

Buy = Cross(Close, TimeFrameExpand(MA14_Weekly, inDaily)); // SAI
```

Luôn cần chỉ định trong hàm TimeFrameExpand( ) rằng khung thời gian nào mà biến đã được tính toán. Vì vậy, nếu MA14_Weekly được tính toán từ dữ liệu ngoài tuần, inWeekly sẽ là tham số đúng cho hàm TimeFrameExpand( ).

Mã đúng:

```
TimeFrameSet(inWeekly);

MA14_Weekly = MA(Close, 14);

TimeFrameRestore();

Buy = Cross(Close, TimeFrameExpand(MA14_Weekly, inWeekly)); // ĐÚNG
```

Hướng Dẫn Tham Khảo Giao Diện Kiểm Tra Danh Mục Đầu Tư

(Cập nhật ngày 20 tháng 2 năm 2010 để bao gồm các cải tiến và bổ sung trong AmiBroker 5.30.0)

Cơ Bản

Phiên bản AmiBroker 4.67.0 đã cung cấp giao diện hướng đối tượng mới cho kiểm tra danh mục đầu tư, cho phép kiểm soát giai đoạn thứ hai của kiểm tra. Điều này cho phép nhiều ứng dụng, bao gồm nhưng không giới hạn:

- Xác định kích thước vị thế dựa trên vốn của danh mục đầu tư
- Thực hiện các hệ thống luân phiên tiên tiến (bạn có thể truy cập các mảng xếp hạng và quyết định giao dịch nào nên thực hiện sau khi biết các mã chứng khoán nào có điểm số tốt nhất trên cơ sở thanh)
- Thêm các chỉ số tùy chỉnh vào thống kê kiểm tra và tối ưu hóa
- Thực hiện các công thức tùy chỉnh để kiểm soát trượt giá
- Quy mô vào/ra nâng cao dựa trên vốn của danh mục và các số liệu thống kê thời gian chạy khác
- Các hệ thống giao dịch tiên tiến sử dụng số liệu thống kê cấp danh mục được đánh giá trên cơ sở thanh để quyết định giao dịch nào nên thực hiện

Tài liệu này mô tả tất cả các đối tượng, phương thức và thuộc tính được giao diện danh mục cung cấp.

Yêu Cầu

Để sử dụng giao diện mới, người dùng cần có AmiBroker 4.67.0 hoặc cao hơn và cần có kỹ năng lập trình AFL, bao gồm cả việc hiểu các thuật ngữ: đối tượng, phương thức và thuộc tính.

Các Cách Tiếp Cận Khác Nhau cho Các Ứng Dụng Khác Nhau

Giao diện kiểm tra danh mục hỗ trợ nhiều cách tiếp cận để tùy chỉnh quá trình kiểm tra phù hợp với các ứng dụng khác nhau.

- **Cách tiếp cận cao cấp (dễ nhất)**:
  - Sử dụng phương thức Backtest() để chạy quy trình kiểm tra mặc định (như trong các phiên bản cũ) - cho phép thực hiện đơn giản các chỉ số tùy chỉnh.
- **Cách tiếp cận trung cấp**:
  - Sử dụng các phương thức PreProcess()/ProcessTradeSignal()/PostProcess() - cho phép sửa đổi tín hiệu, truy vấn các vị thế mở (tốt cho việc xác định kích thước vị thế nâng cao).
- **Cách tiếp cận cấp thấp (phức tạp nhất)**:
  - Sử dụng các phương thức PreProcess()/EnterTrade()/ExitTrade()/ScaleTrade()/UpdateStats()/HandleStops()/PostProcess() - cung cấp quyền kiểm soát toàn bộ quá trình kiểm tra cho các lập trình viên chuyên nghiệp.

Truy Cập Giao Diện

Để truy cập giao diện kiểm tra danh mục mới, bạn cần:

- Bật quy trình kiểm tra tùy chỉnh bằng cách gọi:

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

hoặc gọi

```
SetCustomBacktestProc("C:\\MyPath\\MyCustomBacktest.");
```

trong công thức của bạn, hoặc bằng cách bật nó trong cửa sổ Cài đặt Phân Tích Tự Động, tab "Danh Mục" và chỉ định tệp quy trình tùy chỉnh bên ngoài.

- Truy cập đối tượng kiểm tra bằng cách gọi phương thức GetBacktesterObject(). Lưu ý rằng phương thức GetBacktester chỉ nên được gọi khi Status("action") trả về actionPortfolio:

```
if(Status("action") == actionPortfolio)
{
   // lấy giao diện kiểm tra danh mục
   bo = GetBacktesterObject();

   // ...đây là công thức kiểm tra tùy chỉnh của bạn.
}
```

Khi sử dụng tệp quy trình tùy chỉnh bên ngoài, bạn không cần kiểm tra actionPortfolio, vì các quy trình kiểm tra bên ngoài được gọi độc quyền trong chế độ actionPortfolio.

Quy Tắc Gõ

- **bool** - in nghiêng đại diện cho kiểu tham số/biến/giá trị trả về (Trade, Signal, Stats - đại diện cho loại đối tượng được trả về)
- **AddSymbols** - in đậm đại diện cho tên hàm / phương thức / thuộc tính
- **SymbolList** - kiểu chữ gạch chân đại diện cho tham số chính thức
- **[optional]** - chỉ định tham số tùy chọn (không cần phải cung cấp)
- **variant** - đại diện cho kiểu biến có thể là chuỗi hoặc số

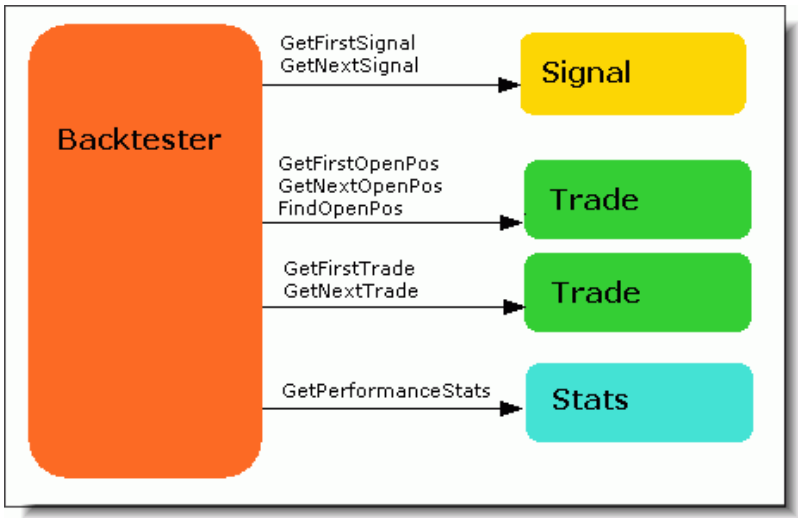
Mặc dù giao diện xử lý kiểu dữ liệu số nguyên như long, short, bool và hai kiểu số thực khác nhau: float và double, AFL tự động chuyển đổi tất cả các kiểu dữ liệu này thành float vì AFL coi tất cả các số là float (số thực 32-bit IEEE).

Các Đối Tượng

Giao diện cung cấp các đối tượng sau:

- **Đối tượng Backtester**
- **Đối tượng Signal**
- **Đối tượng Trade**
- **Đối tượng Stats**

Đối tượng duy nhất có thể truy cập trực tiếp từ AFL là đối tượng Backtester; tất cả các đối tượng khác có thể truy cập bằng cách gọi các phương thức của đối tượng Backtester như được mô tả trong hình dưới đây.



Đối Tượng Kiểm Tra Danh Mục Đầu Tư

Đối tượng Backtester cho phép kiểm soát quá trình kiểm tra (xử lý tín hiệu, vào/thoát/gia tăng vị thế) và truy cập danh sách tín hiệu, vị thế mở, danh sách giao dịch và đối tượng thống kê hiệu suất.

Phương Thức:

- **bool AddCustomMetric(string Title, variant Value, [optional] variant LongOnlyValue, [optional] variant ShortOnlyValue, [optional] variant DecPlaces = 2, [optional] variant CombineMethod = 2)**
  Phương thức này thêm chỉ số tùy chỉnh vào báo cáo kiểm tra, tóm tắt kiểm tra và danh sách kết quả tối ưu hóa. Title là tên của chỉ số hiển thị trong báo cáo, Value là giá trị của chỉ số. Các tham số tùy chọn LongOnlyValue, ShortOnlyValue cho phép cung cấp giá trị cho các cột chỉ số chỉ cho long/short trong báo cáo kiểm tra. Tham số DecPlaces kiểm soát số chữ số thập phân hiển thị giá trị. Tham số CombineMethod xác định cách kết hợp các chỉ số tùy chỉnh cho báo cáo tóm tắt out-of-sample.
  Các giá trị CombineMethod được hỗ trợ:
  1. giá trị bước đầu tiên - báo cáo tóm tắt sẽ hiển thị giá trị của chỉ số tùy chỉnh từ bước out-of-sample đầu tiên.
  2. giá trị bước cuối cùng (mặc định) - báo cáo tóm tắt sẽ hiển thị giá trị của chỉ số tùy chỉnh từ bước out-of-sample cuối cùng.

3. tổng - báo cáo tóm tắt sẽ hiển thị tổng các giá trị của chỉ số tùy chỉnh từ tất cả các bước out-of-sample.
4. trung bình - báo cáo tóm tắt sẽ hiển thị trung bình của các giá trị chỉ số tùy chỉnh từ tất cả các bước out-of-sample.
5. giá trị nhỏ nhất - báo cáo tóm tắt sẽ hiển thị giá trị nhỏ nhất của chỉ số tùy chỉnh từ tất cả các bước out-of-sample.
6. giá trị lớn nhất - báo cáo tóm tắt sẽ hiển thị giá trị lớn nhất của chỉ số tùy chỉnh từ tất cả các bước out-of-sample.

- Lưu ý rằng một số phương pháp tính toán chỉ số rất phức tạp, và việc tính trung bình có thể không dẫn đến biểu diễn chính xác về mặt toán học cho tất cả các bài kiểm tra out-of-sample. Các tóm tắt của tất cả các chỉ số tích hợp là chính xác ngay từ đầu. Điều này trái ngược với các chỉ số tùy chỉnh, vì chúng do người dùng định nghĩa và người dùng phải chọn phương pháp 'kết hợp'.
- **bool Backtest([optional] bool NoTradeList)**
  Phương thức cấp cao này thực hiện quy trình kiểm tra danh mục mặc định trong một lần gọi. Nó nên được sử dụng nếu bạn chỉ cần có được các chỉ số tùy chỉnh mà không muốn thay đổi cách thực hiện kiểm tra. Nếu tham số tùy chọn NoTradeList được đặt là True, danh sách giao dịch sẽ không được tạo tự động. Điều này hữu ích nếu bạn muốn thêm một số chỉ số cho từng giao dịch.
- **long EnterTrade(long Bar, string Symbol, bool bLong, float Price, float PosSize, [optional] variant PosScore, [optional] variant RoundLotSize, [optional] variant MarginDeposit, [optional] variant TickSize, [optional] variant PointValue)**
  Phương thức cấp thấp này cho phép vào giao dịch trên bất kỳ mã nào, kể cả các mã không có tín hiệu tương ứng. Nếu các giá trị cho các tham số tùy chọn không được cung cấp, AmiBroker sẽ sử dụng các giá trị được định nghĩa trong cửa sổ Thông tin Mã.
- **long ExitTrade(long Bar, string Symbol, float Price, [optional] variant ExitType)**
  Phương thức cấp thấp này thoát giao dịch trên bất kỳ mã nào. Phương thức này tìm kiếm trong danh sách giao dịch mở và nếu không có giao dịch mở cho mã đã cho, nó sẽ không làm gì. Tham số tùy chọn ExitType xác định lý do thoát.
- **Trade FindOpenPos(string Symbol)**
  Phương thức này tìm kiếm mã trong danh sách vị thế mở và trả về đối tượng giao dịch phù hợp nếu tìm thấy hoặc trả về đối tượng null nếu không tìm thấy.
- **Signal FindSignal(long Bar, string Symbol, long Type)** - mới trong v5.10
  Tìm tín hiệu đầu tiên phù hợp có fPrice != -1. Nếu 0 được sử dụng như kiểu, và tín hiệu vào và thoát nằm trên cùng một thanh, tín hiệu vào sẽ được trả về.
- **long GetSignalQty(long Bar, long Type)** - mới trong v5.30
  Phương thức này lấy số lượng tín hiệu xảy ra trên thanh đã cho.
- **Trade GetFirstOpenPos()**
  Phương thức này trả về đối tượng Trade đầu tiên từ danh sách vị thế mở.
- **Signal GetFirstSignal(long Bar)**
  Phương thức này trả về đối tượng tín hiệu giao dịch đầu tiên cho thanh đã cho.

- **Trade GetFirstTrade()**
  Phương thức này trả về đối tượng Trade đầu tiên từ danh sách giao dịch đã đóng.
- **Trade GetNextOpenPos()**
  Phương thức này trả về đối tượng Trade tiếp theo từ danh sách vị thế mở. Bạn nên gọi GetFirstOpenPos trước khi gọi phương thức này lần đầu tiên.
- **Signal GetNextSignal(long Bar)**
  Phương thức này trả về đối tượng Signal tiếp theo từ danh sách tín hiệu đã đóng cho thanh đã cho.
- **Trade GetNextTrade()**
  Phương thức này trả về đối tượng Trade tiếp theo từ danh sách giao dịch đã đóng.
- **long GetOpenPosQty()**
  Phương thức này trả về số lượng vị thế mở hiện tại.
- **Stats GetPerformanceStats(long Type)**
  Tính toán các thống kê và chỉ số tích hợp và trả về đối tượng Stats. Tham số Type xác định các giao dịch nào sẽ được tính.
- **HandleStops(long Bar)**
  Phương thức cấp thấp này xử lý các điểm dừng tự động (applystops). Phương thức này không nên được sử dụng trong các cách tiếp cận cấp cao và trung cấp.
- **ListTrades()**
  Phương thức này xuất các giao dịch vào danh sách kết quả của cửa sổ Phân Tích Tự Động. Thông thường, phương thức này không cần được gọi vì phương thức Backtest() đã tự động liệt kê các giao dịch.
- **PostProcess()**
  Phương thức này thực hiện xử lý cuối cùng cần thiết để hoàn thành kiểm tra chính xác. Nó giải phóng bộ nhớ giá, đóng tất cả các giao dịch mở và xuất danh sách giao dịch vào cửa sổ Phân Tích Tự Động.
- **PreProcess()**
  Phương thức này thực hiện xử lý ban đầu cần thiết để thực hiện kiểm tra chính xác. Nó khởi tạo bộ nhớ giá và thiết lập các biến ban đầu.
- **bool ProcessTradeSignals(long Bar)**
  Phương thức này xử lý tất cả các tín hiệu giao dịch cho thanh đã cho.
- **RawTextOutput(string Text)**
  Phương thức này xuất bất kỳ chuỗi nào vào danh sách kết quả Phân Tích Tự Động vào thời điểm gọi.
- **long ScaleTrade(long Bar, string Symbol, bool bIncrease, float Price, float PosSize, [optional] variant Deposit)**
  Phương thức cấp thấp này điều chỉnh quy mô giao dịch trên bất kỳ mã nào.
- **UpdateStats(long Bar, long TimeInsideBar)**
  Phương thức cấp thấp này cập nhật vốn, tiếp xúc, các chuyến đi giao dịch và các biến nội bộ khác cần thiết để tính toán chính xác các thống kê.
- **GetMonteCarloSim()**
  Lấy thể hiện của đối tượng MonteCarloSim để truy cập các phân phối MC.

Thuộc Tính:

- **double Cash**
  Số tiền có sẵn (tiền mặt) trong danh mục đầu tư của bạn.

Đối Tượng Kiểm Tra Danh Mục Đầu Tư

Thuộc Tính:

- **double Equity**
  - Giá trị vốn hiện tại của danh mục đầu tư (thuộc tính chỉ đọc).
- **array EquityArray**
  - Mảng vốn của danh mục đầu tư (thuộc tính chỉ đọc).
  - Lưu ý: Kể từ phiên bản 5.50, thuộc tính này trả về toàn bộ mảng vốn. Giá trị chỉ được điền trong quá trình kiểm tra, và chỉ sau khi kiểm tra hoàn tất, tất cả các giá trị mới hợp lệ. Nếu gọi giữa chừng, nó chỉ chứa dữ liệu "đến điểm đã cho". Hạn chế sử dụng để tránh tiêu tốn RAM/CPU. Bạn có thể sử dụng bo.EquityArray thay vì Foreign("~~~Equity", "C") trong mã kiểm tra tùy chỉnh.
- **double InitialEquity**
  - Số tiền có sẵn tại thời điểm bắt đầu kiểm tra.
- **double MarginLoan**
  - Số tiền vay (chỉ nếu bạn sử dụng tài khoản ký quỹ) (thuộc tính chỉ đọc).

Đối Tượng Tín Hiệu

Đối tượng Signal đại diện cho tín hiệu giao dịch (mua/bán/short/cover) hoặc phần tử mảng xếp hạng được tạo ra bởi AmiBroker trong giai đoạn đầu tiên của kiểm tra khi công thức của bạn được thực hiện trên mọi mã.

Phương Thức:

- **bool IsEntry()**
  - Trả về True nếu là tín hiệu vào, False ngược lại.
- **bool IsExit()**
  - Trả về True nếu là tín hiệu thoát, False ngược lại.
- **bool IsLong()**
  - Trả về True nếu là tín hiệu vào dài (mua) hoặc thoát dài (bán), False ngược lại.
- **bool IsScale()**
  - Trả về True nếu là tín hiệu điều chỉnh (scale-in/out), False ngược lại.

Thuộc Tính:

- **float MarginDeposit**
  - Tiền ký quỹ (cho hợp đồng tương lai).
- **float PointValue**
  - Giá trị điểm (cho hợp đồng tương lai, tiền tệ).
- **float PosScore**
  - Điểm số vị thế.
- **float PosSize**
  - Kích thước vị thế yêu cầu (số dương nghĩa là giá trị đô la, số âm nghĩa là phần trăm của vốn).
- **float Price**
  - Giá vào/thoát/gia tăng.

- **short int Reason**
  - Lý do thoát (1 - thoát bình thường, 2 - lỗ tối đa, 3 - lợi nhuận, 4 - trailing, 5 - N-bar, 6 - phá sản).
- **float RoundLotSize**
  - Kích thước lô tròn.
- **string Symbol**
  - Mã chứng khoán.
- **float TickSize**
  - Kích thước tick (thay đổi giá tối thiểu).
- **short int Type**
  - Loại tín hiệu (0 - xếp hạng, 1 - mua, 2 - bán, 3 - short, 4 - cover, 5 - scale-in, 6 - scale-out).

Đối Tượng Giao Dịch

Đối tượng Trade đại diện cho vị thế hiện tại (giao dịch mở) hoặc giao dịch đã đóng. AmiBroker duy trì 2 danh sách giao dịch: danh sách vị thế mở và danh sách giao dịch đã đóng.

Phương Thức:

- **long AddCustomMetric(string Title, variant Value, [optional] decplaces = 2)**
  - Thêm chỉ số tùy chỉnh cho danh sách giao dịch chỉ. Phải gọi trước khi gọi PostProcess.
- **float GetCommission([optional] bool InclExit)**
  - Nhận hoa hồng đã trả cho giao dịch (bao gồm/không bao gồm hoa hồng thoát).
- **double GetEntryValue()**
  - Nhận giá trị đô la vào của giao dịch.
- **float GetMAE()**
  - Nhận mức lỗ tối đa của giao dịch theo phần trăm.
- **float GetMFE()**
  - Nhận mức lợi nhuận tối đa của giao dịch theo phần trăm.
- **double GetPercentProfit()**
  - Nhận phần trăm lợi nhuận hiện tại của giao dịch.
- **double GetPositionValue()**
  - Nhận giá trị đô la hiện tại của vị thế.
- **float GetPrice(long Bar, string Field)**
  - Cung cấp truy cập nhanh đến mảng giá của vị thế mở.
- **double GetProfit()**
  - Nhận lợi nhuận hiện tại của giao dịch.

Thuộc Tính:

- **long BarsInTrade**
  - Số thanh đã tiêu tốn trong giao dịch.
- **float EntryDateTime**
  - Thời gian vào theo định dạng nội bộ của AmiBroker.

- **float EntryPrice**
  - Giá vào, nếu có điều chỉnh xảy ra.
- **float ExitDateTime**
  - Thời gian thoát theo định dạng nội bộ của AmiBroker.
- **float ExitPrice**
  - Giá thoát, nếu có điều chỉnh xảy ra.
- **bool IsLong**
  - Trả về True nếu giao dịch là dài, False ngược lại.
- **bool IsOpen**
  - Trả về True nếu giao dịch đang mở, False ngược lại.
- **float MarginDeposit**
  - Tiền ký quỹ ban đầu.
- **double MarginLoan**
  - Số tiền vay được sử dụng cho giao dịch.
- **string Symbol**
  - Mã chứng khoán.
- **float TickSize**
  - Kích thước tick (thay đổi giá tối thiểu).

Đối Tượng Giao Dịch (Trade Object)

Đặc điểm

- **Loại tín hiệu**
  - 0: Xếp hạng (chỉ cho hệ thống quay vòng)
  - 1: Mua
  - 2: Bán
  - 3: Short
  - 4: Cover
  - 5: Scale-in
  - 6: Scale-out

Phương thức

Lặp qua danh sách vị thế mở

```
// 'bo' biến lưu đối tượng Backtester đã được lấy trước đó

for (openpos = bo.GetFirstOpenPos(); openpos; openpos = bo.GetNextOpenPos()) {

    // openpos giờ đây giữ đối tượng Trade

}
```

Lặp qua danh sách giao dịch đã đóng

```
for (trade = bo.GetFirstTrade(); trade; trade = bo.GetNextTrade()) {

    // trade giờ đây giữ đối tượng Trade

}
```

-

Phương thức chính

- **long AddCustomMetric(string Title, variant Value, [optional] decplaces = 2)**
  - Thêm chỉ số tùy chỉnh cho danh sách giao dịch. Gọi trước khi PostProcess để đảm bảo đầu ra chính xác.
- **float GetCommission([optional] bool InclExit)**
  - Nhận hoa hồng đã trả cho giao dịch (bao gồm cả hoa hồng scale in/out).
- **double GetEntryValue()**
  - Nhận giá trị vào của giao dịch theo đô la.
- **float GetMAE()**
  - Nhận Mức lỗ tối đa (Maximum Adverse Excursion) của giao dịch.
- **float GetMFE()**
  - Nhận Mức lợi nhuận tối đa (Maximum Favorable Excursion) của giao dịch.
- **double GetPercentProfit()**
  - Nhận phần trăm lợi nhuận hiện tại của giao dịch.
- **double GetPositionValue()**
  - Nhận giá trị hiện tại của vị thế theo đô la.
- **float GetPrice(long Bar, string Field)**
  - Cung cấp truy cập nhanh đến mảng giá của vị thế mở.

Thuộc tính

- **long BarsInTrade**
  - Số thanh đã dành cho giao dịch (tính từ 0). Giá trị 0 chỉ có khi giao dịch vừa mở.
- **float EntryDateTime**
  - Thời gian vào theo định dạng nội bộ của AmiBroker.
- **float EntryPrice**
  - Giá vào, nếu có điều chỉnh xảy ra.
- **float ExitDateTime**
  - Thời gian thoát theo định dạng nội bộ của AmiBroker.
- **float ExitPrice**
  - Giá thoát, nếu có điều chỉnh xảy ra.
- **double Handle**
  - Giá trị handle nội bộ cho phép nhận diện và quản lý các giao dịch mở cùng một mã.
- **bool IsLong**
  - Trả về True nếu giao dịch là dài, False ngược lại.
- **bool IsOpen**
  - Trả về True nếu giao dịch đang mở, False ngược lại.
- **float MarginDeposit**
  - Tiền ký quỹ ban đầu.
- **double MarginLoan**
  - Số tiền vay cho giao dịch.
- **float PointValue**
  - Giá trị điểm (cho hợp đồng tương lai/tiền tệ).

- **float RoundLotSize**
  - Kích thước lô tròn.
- **float Score**
  - Điểm số vào.
- **float Shares**
  - Số lượng cổ phiếu/hợp đồng.
- **string Symbol**
  - Mã chứng khoán.
- **string FullName**
  - Tên đầy đủ của công cụ (thêm trong phiên bản 5.69).
- **float TickSize**
  - Kích thước tick (thay đổi giá tối thiểu).

Đối Tượng Thống Kê (Stats Object)

Mục đích

Đối tượng Stats cung cấp quyền truy cập vào các thống kê và chỉ số tích hợp sẵn của bộ kiểm tra lùi (backtester). Các chỉ số thường được tính toán khi hoàn tất việc kiểm tra lùi, nhưng cũng có thể tính toán trong quá trình này. Để tính toán và truy cập các chỉ số, bạn có thể sử dụng phương thức GetPerformanceStats của đối tượng Backtester.

Cách sử dụng

```
// 'bo' biến lưu đối tượng Backtester đã được lấy trước đó

stats = bo.GetPerformanceStats(0);
```

Phương thức

- **double GetValue(string MetricName)**
  - Trả về giá trị của một chỉ số. Các chỉ số có thể bao gồm:
  - **Vốn ban đầu và Vốn cuối**
    - "InitialCapital"
    - "EndingCapital"
  - **Lợi nhuận**
    - "NetProfit"
    - "NetProfitPercent"
  - **Khả năng tiếp xúc**
    - "ExposurePercent"
    - "NetRAR"
  - **Tỷ lệ lợi nhuận**
    - "CAR"
    - "RAR"
  - **Tổng quan giao dịch**

        - "AllQty"
        - "AllPercent"
        - "AllAvgProfitLoss"
        - "AllAvgProfitLossPercent"
        - "AllAvgBarsHeld"
    - **Giao dịch thắng**
        - "WinnersQty"
        - "WinnersPercent"
        - "WinnersTotalProfit"
        - "WinnersAvgProfit"
        - "WinnersAvgProfitPercent"
        - "WinnersAvgBarsHeld"
        - "WinnersMaxConsecutive"
        - "WinnersLargestWin"
        - "WinnersLargestWinBars"
    - **Giao dịch thua**
        - "LosersQty"
        - "LosersPercent"
        - "LosersTotalLoss"
        - "LosersAvgLoss"
        - "LosersAvgLossPercent"
        - "LosersAvgBarsHeld"
        - "LosersMaxConsecutive"
        - "LosersLargestLoss"
        - "LosersLargestLossBars"
    - **Rủi ro và lợi nhuận**
        - "MaxTradeDrawdown"
        - "MaxTradeDrawdownPercent"
        - "MaxSystemDrawdown"
        - "MaxSystemDrawdownPercent"
        - "RecoveryFactor"
        - "CAR/MDD"
        - "RAR/MDD"
        - "ProfitFactor"
        - "PayoffRatio"
        - "StandardError"
        - "RRR"
        - "UlcerIndex"
        - "UlcerPerformanceIndex"
    - **Tỷ lệ Sharpe**
        - "SharpeRatio"
        - "KRatio"

Đối Tượng Mô Phỏng Monte Carlo (MonteCarloSim Object)

Mục đích

Đối tượng Monte Carlo cho phép truy cập kết quả mô phỏng Monte Carlo và có một phương thức duy nhất.

Phương thức

GetValue(string field, float percentile)

- Nhận giá trị "field" ở mức phần trăm chỉ định. Các trường có sẵn bao gồm:
  - "FinalEquity"
  - "CAR"
  - "LowestEquity"
  - "MaxDrawdown"
  - "MaxPercDrawdown"

Tài nguyên bổ sung

Bạn có thể tìm thêm ví dụ và tài liệu chi tiết trong tài liệu Houston về giao diện kiểm tra lùi tùy chỉnh (dạng PDF 300 KB) và Cơ sở tri thức tại: AmiBroker Knowledge Base.

Cách thêm chỉ số định nghĩa bởi người dùng vào báo cáo kiểm tra/ tối ưu hóa

Một trong những bổ sung mới trong phiên bản BETA 4.67.x/4.68.x là giao diện lập trình backtester danh mục đầu tư, cung cấp quyền kiểm soát hoàn toàn giai đoạn thứ hai của kiểm tra danh mục đầu tư. Điều này cho phép nhiều ứng dụng, bao gồm nhưng không giới hạn ở:

• chỉ số định nghĩa bởi người dùng (xuất hiện dưới dạng cột bổ sung trong danh sách kết quả kiểm tra "tóm tắt", trong kết quả tối ưu hóa và dưới dạng hàng mới trong trang "thống kê" của báo cáo kiểm tra, cũng như các chỉ số theo giao dịch) • truy cập vào vốn cấp danh mục khi vòng lặp kiểm tra đang chạy - cho phép ví dụ như kích thước vị trí phức tạp dựa trên vốn của danh mục • truy cập đọc/ghi vào tiền mặt của danh mục - cho phép thêm quỹ vào danh mục • truy cập đọc/ghi theo từng thanh để xử lý các tín hiệu giao dịch được tạo ra trong giai đoạn kiểm tra đầu tiên, cho phép đọc mảng xếp hạng và điều chỉnh giá tín hiệu (ví dụ như công thức trượt giá tùy chỉnh), kích thước vị trí, v.v. • truy cập vào danh sách các vị trí đang mở, mỗi vị trí có thể được truy vấn cho nhiều thuộc tính khác nhau bao gồm lợi nhuận, MAE/MFE, số thanh trong giao dịch, v.v. • truy cập vào danh sách các giao dịch đã đóng, mỗi giao dịch đã đóng có thể được truy vấn cho nhiều thuộc tính khác nhau bao gồm lợi nhuận, MAE/MFE, số thanh trong giao dịch, v.v. • ba cấp độ lập trình khác nhau: • cấp cao - sử dụng phương thức Backtest() và chạy quy trình kiểm tra mặc định (như trong các phiên bản cũ) - tuyệt vời để thêm chỉ số tùy chỉnh • cấp giữa - sử dụng các phương thức PreProcess()/ProcessTradeSignal()/PostProcess() - cho phép điều chỉnh tín hiệu, truy vấn các vị trí mở (tốt cho kích thước vị trí nâng cao) • cấp thấp - sử dụng các phương thức PreProcess()/EnterTrade()/ExitTrade()/ScaleTrade()/UpdateStats()/HandleStops()/PostProcess() cung cấp quyền kiểm soát hoàn toàn quy trình kiểm tra cho các lập trình viên viết mã cứng

Tài liệu kỹ thuật của giao diện mới có sẵn ở đây, trong chương này chúng ta sẽ chỉ tập trung vào một số ví dụ thực tế.

Thêm chỉ số định nghĩa bởi người dùng

Ví dụ 1

Hãy bắt đầu với ứng dụng dễ nhất: trong ví dụ đầu tiên, tôi sẽ cho bạn thấy cách thêm chỉ số định nghĩa bởi người dùng vào báo cáo danh mục đầu tư và danh sách kết quả tối ưu hóa.

Trong bước đầu tiên, chúng ta sẽ thêm Expectancy vào báo cáo kiểm tra và tối ưu hóa. Có một số thảo luận về cách tính toán expectancy nhưng công thức dễ nhất cho nó là:

Expectancy ($) = %Người thắng * Lợi nhuận trung bình - %Người thua * Lỗ trung bình hoặc (cách khác để tính cùng một điều) Expectancy ($) = (Tổng lợi nhuận - Tổng lỗ) / Số giao dịch = Lợi nhuận ròng / Số giao dịch

Chúng ta bắt đầu với công thức đơn giản này. Với cách tiếp cận này, expectancy đơn giản cho chúng ta biết lợi nhuận kỳ vọng mỗi giao dịch bằng đô la. Công thức kiểm tra tùy chỉnh thực hiện chỉ số định nghĩa bởi người dùng này trông như sau:

```
/* Đầu tiên chúng ta cần kích hoạt quy trình kiểm tra tùy chỉnh và ** cho AmiBroker biết sử dụng công thức hiện tại */

SetCustomBacktestProc("");

/* Bây giờ quy trình kiểm tra tùy chỉnh theo sau */

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

bo.Backtest(); // chạy quy trình kiểm tra mặc định

st = bo.GetPerformanceStats(0); // lấy thống kê cho tất cả giao dịch

// Tính toán Expectancy (cách đơn giản) // %Win * Lợi nhuận trung bình - %Los * Lỗ trung bình // lưu ý rằng vì Lỗ trung bình đã là số âm // trong AmiBroker nên chúng ta đang cộng giá trị thay vì trừ đi // chúng ta cũng có thể sử dụng công thức đơn giản hơn Lợi nhuận ròng/Số giao dịch // nhưng để minh họa, chúng ta đang sử dụng công thức phức tạp hơn :-) expectancy = st.GetValue("WinnersAvgProfit")*st.GetValue("WinnersPercent")/100 + st.GetValue("LosersAvgLoss")*st.GetValue("LosersPercent")/100;

// Ở đây chúng ta thêm chỉ số tùy chỉnh vào báo cáo kiểm tra bo.AddCustomMetric( "Expectancy ($)", expectancy ); }

// hệ thống giao dịch của bạn ở đây fast = Optimize("fast", 12, 5, 20, 1 ); slow = Optimize("slow", 26, 10, 25, 1 ); Buy=Cross(MACD(fast,slow),Signal(fast,slow)); Sell=Cross(Signal(fast,slow),MACD(fast,slow));
```

Đầu tiên chúng ta cần cho AmiBroker biết sử dụng công thức kiểm tra tùy chỉnh thay vì công thức tích hợp sẵn. Chúng ta làm điều này bằng cách gọi SetCustomBacktestProc. Tham số đầu tiên xác định đường dẫn đến công thức kiểm tra tùy chỉnh (có thể được lưu trữ trong một tệp bên ngoài, độc lập với hệ thống giao dịch thực tế). Nếu chúng ta cung cấp chuỗi rỗng ở đó, chúng ta đang nói với AmiBroker sử dụng công thức hiện tại (cùng công thức mà được sử dụng cho hệ thống giao dịch).

Trong dòng tiếp theo, chúng ta có câu lệnh "if" để vào công thức kiểm tra tùy chỉnh nếu động cơ phân tích đang ở giai đoạn actionPortfolio (giai đoạn thứ hai của kiểm tra danh mục đầu tư). Điều này rất quan trọng vì công thức được thực thi trong cả giai đoạn quét (khi tín hiệu giao dịch được tạo ra) và trong giai đoạn kiểm tra danh mục đầu tư thực tế. Câu lệnh "if" cho phép chúng ta vào phần quy trình kiểm tra tùy chỉnh chỉ khi động cơ phân tích đang ở giai đoạn kiểm tra thực tế.

Trong dòng tiếp theo, chúng ta lấy quyền truy cập vào giao diện lập trình backtester bằng cách gọi hàm GetBacktesterObject. Hàm này trả về đối tượng Backtester, được sử dụng để truy cập tất cả chức năng của giao diện mới (các chi tiết về các đối tượng có sẵn xem tại: http://www.amibroker.com/docs/ab401.html).

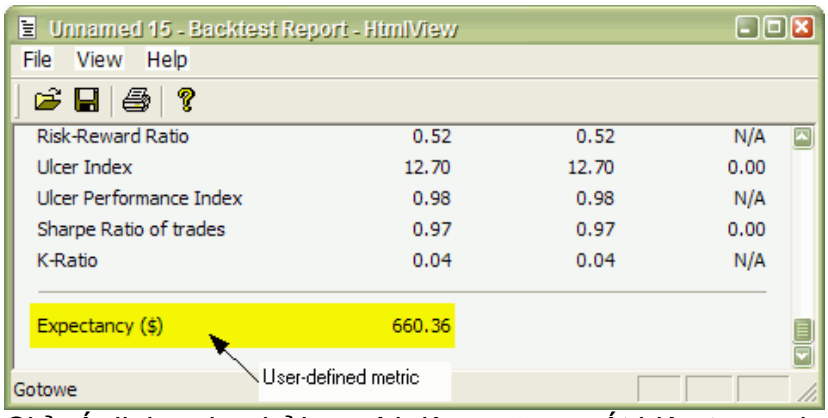
Sau đó, chúng ta lấy quyền truy cập vào các chỉ số tích hợp bằng cách gọi phương thức GetPerformanceStats của đối tượng backtester. Phương thức này trả về đối tượng Statistics, cho phép chúng ta truy cập bất kỳ chỉ số tích hợp nào bằng cách gọi phương thức GetValue.

Bước tiếp theo, chúng ta tính toán giá trị expectancy từ các chỉ số tích hợp được lấy bằng phương thức GetValue. Để xem danh sách các chỉ số được hỗ trợ bởi phương thức GetValue, vui lòng kiểm tra: http://www.amibroker.com/docs/ab401.html.

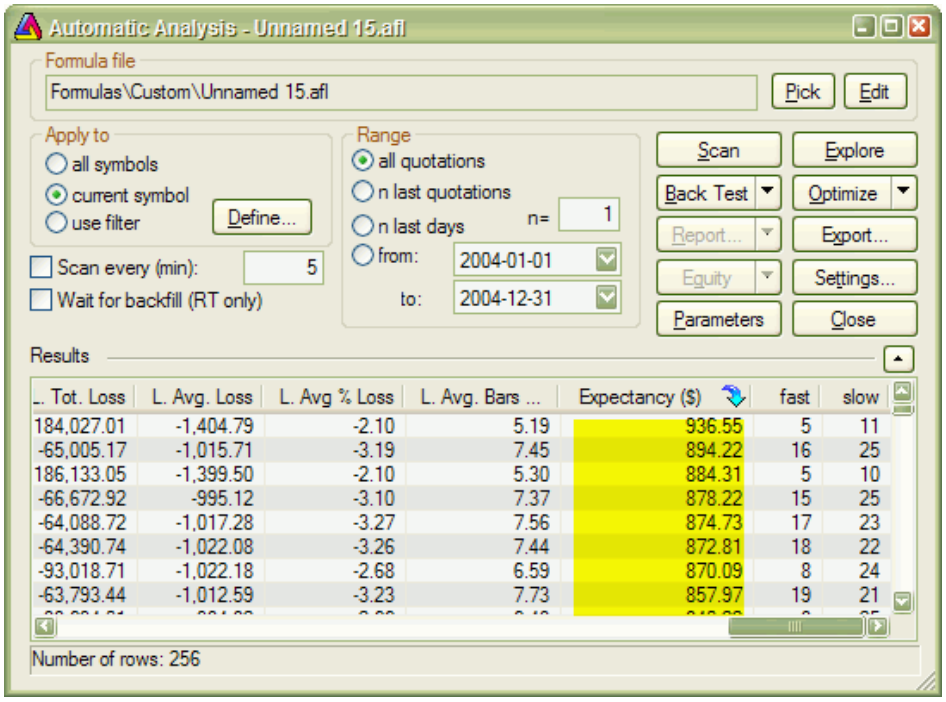
Trong bước cuối cùng, chúng ta đơn giản thêm chỉ số tùy chỉnh vào báo cáo bằng cách gọi hàm AddCustomMetric của đối tượng Backtester. Tham số đầu tiên là tên của chỉ số, tham số thứ hai là giá trị.

Sau câu lệnh "if" thực hiện quy trình kiểm tra tùy chỉnh của chúng ta, các quy tắc hệ thống giao dịch thông thường sẽ theo sau.

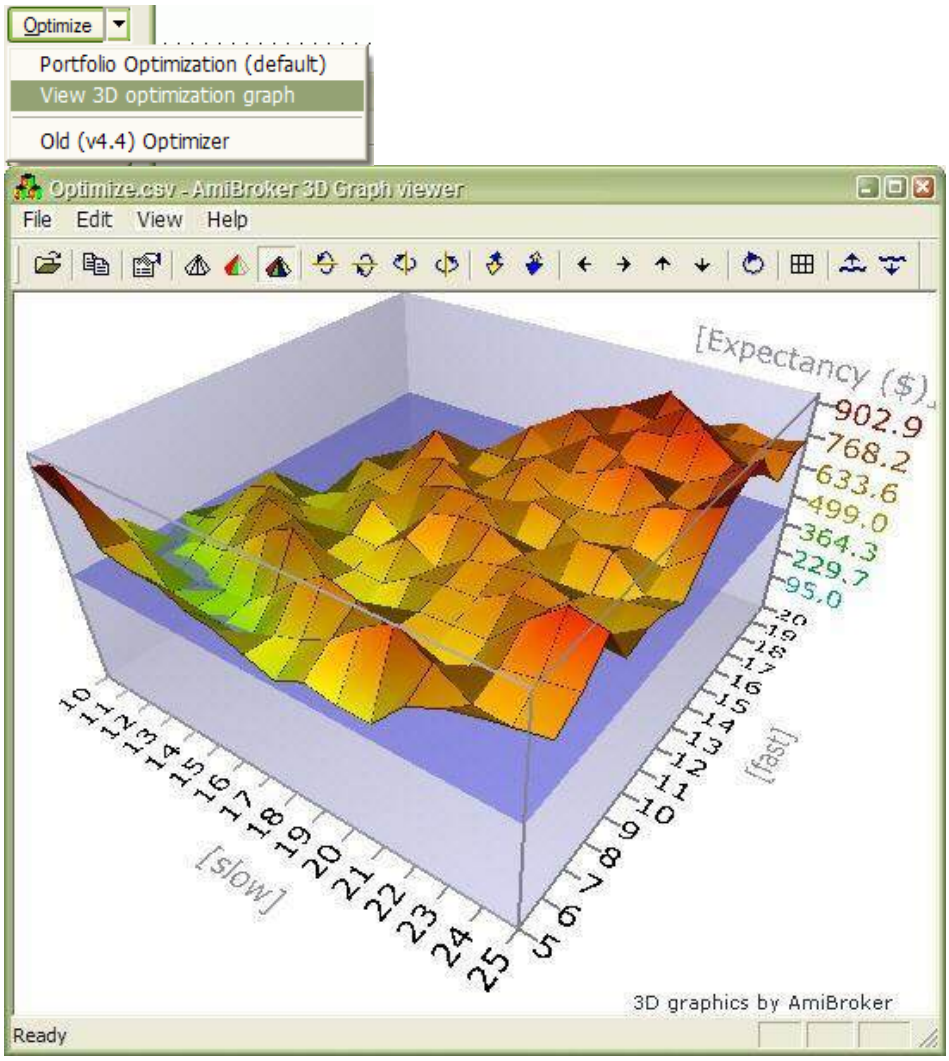
Giờ đây, khi bạn chạy Backtest và nhấp vào nút Báo cáo trong cửa sổ Phân tích Tự động, bạn sẽ thấy chỉ số tùy chỉnh của mình được thêm vào cuối trang thống kê.



Chỉ số định nghĩa bởi người dùng cũng xuất hiện trong danh sách kết quả tối ưu hóa:



Khi bạn nhấp vào cột chỉ số tùy chỉnh, các kết quả tối ưu hóa sẽ được sắp xếp theo chỉ số của bạn và bạn sẽ có thể hiển thị biểu đồ 3D của chỉ số định nghĩa bởi người dùng được vẽ so với các biến tối ưu hóa.



Ví dụ 2

Một số người chỉ ra rằng phương pháp đơn giản này để tính toán expectancy chỉ hoạt động tốt với kích thước vị trí cố định. Nếu không, với kích thước vị trí biến đổi và/hoặc gộp lãi, các giao dịch lớn sẽ có trọng số cao hơn các giao dịch nhỏ và điều này dẫn đến giá trị expectancy không chính xác. Để giải quyết vấn đề này, người ta có thể tính toán expectancy ví dụ như lợi nhuận kỳ vọng trên mỗi $100 đã đầu tư. Để tính toán thống kê như vậy, cần lặp qua các giao dịch, cộng dồn lợi nhuận trên mỗi đơn vị $100 và chia tổng này cho số giao dịch.

Công thức phù hợp như sau:

/* Đầu tiên chúng ta cần kích hoạt quy trình kiểm tra tùy chỉnh và ** cho AmiBroker biết sử dụng công thức hiện tại */ SetCustomBacktestProc("");

```
/* Bây giờ quy trình kiểm tra tùy chỉnh theo sau */
if( Status("action") == actionPortfolio ) { bo = GetBacktesterObject();
bo.Backtest(); // chạy quy trình kiểm tra mặc định

SumProfitPer100Inv = 0;
NumTrades = 0;

// lặp qua các giao dịch đã đóng trước
for( trade = bo.GetFirstTrade(); trade; trade = bo.GetNextTrade() )
{
   // ở đây chúng ta cộng dồn lợi nhuận trên mỗi $100 đã đầu tư
   SumProfitPer100Inv = SumProfitPer100Inv + trade.GetPercentProfit();
   NumTrades++;
}

// lặp qua các vị trí đang mở nếu có
for( trade = bo.GetFirstOpenPos(); trade; trade = bo.GetNextOpenPos() )
{
   SumProfitPer100Inv = SumProfitPer100Inv + trade.GetPercentProfit();
   NumTrades++;
}

expectancy2 = SumProfitPer100Inv / NumTrades;

bo.AddCustomMetric( "Expectancy (per $100 inv.)", expectancy2 );

}
```

// hệ thống giao dịch của bạn ở đây fast = Optimize("fast", 12, 5, 20, 1); slow = Optimize("slow", 26, 10, 25, 1); Buy = Cross(MACD(fast, slow), Signal(fast, slow)); Sell = Cross(Signal(fast, slow), MACD(fast, slow));

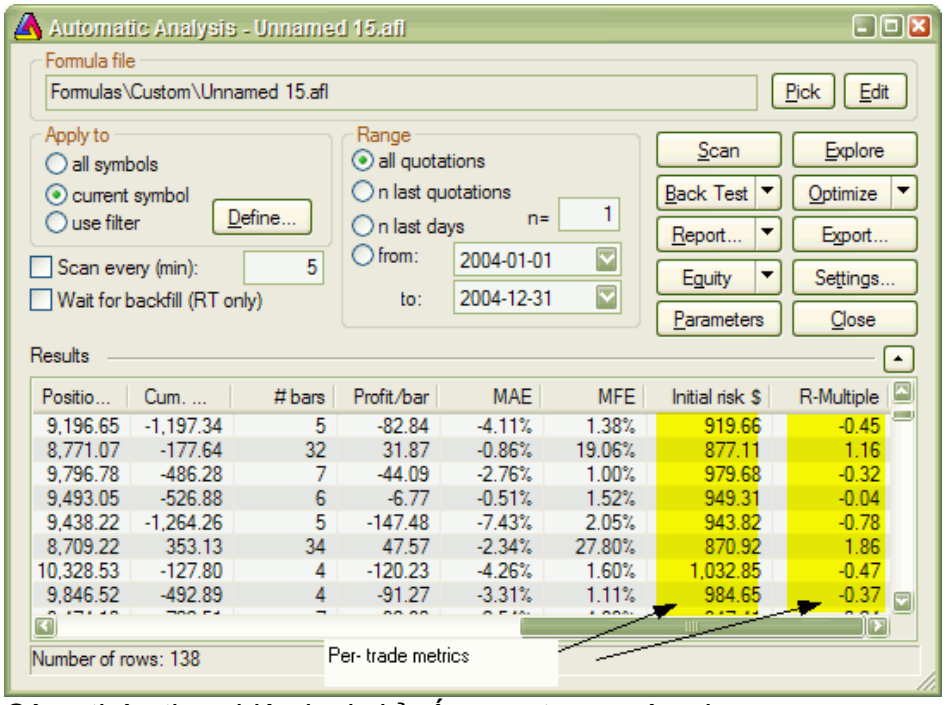
Sự khác biệt duy nhất giữa công thức này và công thức trước là chúng ta không sử dụng các chỉ số tích hợp để tính toán chỉ số expectancy của riêng mình. Thay vào đó, chúng ta cộng dồn tất cả lợi nhuận phần trăm của từng giao dịch (tương đương với lợi nhuận bằng đô la từ đầu tư $100) và cuối cùng chia tổng này cho số giao dịch. Việc cộng dồn được thực hiện trong vòng lặp "for". Cặp hàm GetFirstTrade/GetNextTrade của đối tượng backtester cho phép chúng ta bước qua danh sách các giao dịch đã đóng. Chúng ta sử dụng hai vòng lặp (vòng lặp thứ hai sử dụng GetFirstOpenPos/GetNextOpenPos) vì có thể có một số vị trí mở còn lại ở cuối kiểm tra. Nếu chúng ta chỉ muốn bao gồm các giao dịch đã đóng, thì chúng ta có thể loại bỏ vòng lặp "for" thứ hai.

Sau khi chạy đoạn mã này, chúng ta nhận thấy rằng expectancy được tính toán theo cách này, ngay cả khi điều chỉnh theo vốn ban đầu (bằng cách nhân với yếu tố InitialEquity/$100), vẫn nhỏ hơn expectancy được tính trong ví dụ đầu tiên. Điều này cho thấy rằng phương pháp "dễ" để tính toán expectancy (từ ví dụ 1) có thể dẫn đến kết quả quá lạc quan.

Ví dụ 3

Một số người theo dõi Van Tharp lại thích một "biến thể" hơi khác của chỉ số expectancy. Họ diễn đạt expectancy theo lợi nhuận kỳ vọng trên mỗi "đơn vị rủi ro". Lợi nhuận được diễn đạt theo các R-multiple, trong đó 1R được định nghĩa là số tiền rủi ro trên mỗi giao dịch. Số tiền rủi ro là số tiền tối đa bạn có thể mất, và thường được xác định bởi mức lỗ tối đa (hoặc dừng lỗ theo dõi). Theo Tharp, cách dễ nhất để tính toán expectancy là cộng dồn tất cả các R-multiple của bạn và trừ đi các R-multiple âm từ các R-multiple dương, sau đó chia cho số giao dịch. Điều này sẽ cho bạn chỉ số expectancy trên mỗi giao dịch.

Cách tiếp cận này rất giống với phương pháp được trình bày trong ví dụ 2, nhưng trong các tính toán, chúng ta không sử dụng giá trị của giao dịch mà thay vào đó là rủi ro trên mỗi giao dịch. Rủi ro phụ thuộc vào mức dừng mà chúng ta sử dụng trong hệ thống giao dịch của mình. Để đơn giản, trong ví dụ này, chúng ta đã sử dụng mức dừng lỗ tối đa là 10%. Trong ví dụ này, chúng ta cũng thêm các chỉ số theo giao dịch để minh họa rõ hơn về cách tính toán R-multiple. Các chỉ số theo giao dịch xuất hiện trong mỗi hàng của danh sách giao dịch trong kết quả kiểm tra.



Công thức thực hiện loại chỉ số expectancy này như sau:

/* Đầu tiên chúng ta cần kích hoạt quy trình kiểm tra tùy chỉnh và ** cho AmiBroker biết sử dụng công thức hiện tại */ SetCustomBacktestProc(""); MaxLossPercentStop = 10; // mức dừng lỗ tối đa 10%

/* Bây giờ quy trình kiểm tra tùy chỉnh theo sau */ if(Status("action") == actionPortfolio) { bo = GetBacktesterObject(); bo.Backtest(1); // chạy quy trình kiểm tra mặc định SumProfitPerRisk = 0; NumTrades = 0;

// lặp qua các giao dịch đã đóng trước

for(trade = bo.GetFirstTrade(); trade; trade = bo.GetNextTrade())

{

// Rủi ro được tính là giá trị tối đa mà chúng ta có thể mất cho mỗi giao dịch // Trong ví dụ này, chúng ta đang sử dụng mức dừng lỗ tối đa

// Điều này có nghĩa là chúng ta không thể mất nhiều hơn (MaxLoss%) của số tiền đã đầu tư // Do đó, rủi ro

Risk = (MaxLossPercentStop / 100) * trade.GetEntryValue(); RMultiple = trade.GetProfit() / Risk;

trade.AddCustomMetric("Rủi ro ban đầu $", Risk); trade.AddCustomMetric("R-Multiple", RMultiple);

SumProfitPerRisk = SumProfitPerRisk + RMultiple; NumTrades++; }

```
expectancy3 = SumProfitPerRisk / NumTrades; bo.AddCustomMetric("Expectancy (theo rủi ro)", expectancy3); bo.ListTrades(); }

// Hệ thống giao dịch của bạn ở đây ApplyStop(stopTypeLoss, stopModePercent, MaxLossPercentStop); fast = Optimize("fast", 12, 5, 20, 1); slow = Optimize("slow", 26, 10, 25, 1); Buy = Cross(MACD(fast, slow), Signal(fast, slow)); Sell = Cross(Signal(fast, slow), MACD(fast, slow));
```

Đoạn mã này về cơ bản rất giống với ví dụ 2. Chỉ có một vài khác biệt. Đầu tiên, chúng ta gọi phương thức Backtest với tham số NoTradeList được đặt thành 1. Bằng cách này, chúng ta vô hiệu hóa danh sách giao dịch mặc định, vì vậy chúng ta có thể thêm các chỉ số tùy chỉnh theo giao dịch và liệt kê các giao dịch sau bằng cách gọi phương thức ListTrades. Sau đó, chúng ta lặp qua các giao dịch và tính toán rủi ro dựa trên giá trị nhập giao dịch và mức dừng lỗ tối đa được sử dụng. RMultiple sau đó được tính là lợi nhuận giao dịch chia cho số tiền rủi ro trên mỗi giao dịch. Cả rủi ro và R-multiple sau đó được thêm vào dưới dạng các chỉ số tùy chỉnh theo giao dịch (lưu ý rằng chúng ta đang gọi phương thức AddCustomMetric của đối tượng Trade ở đây). Sau đó, chúng ta thực hiện các tính toán còn lại. Cuối cùng, trong quy trình kiểm tra tùy chỉnh, chúng ta thêm chỉ số kiểm tra tùy chỉnh (lần này gọi phương thức AddCustomMetric của đối tượng Backtester), và sau đó chúng ta kích hoạt danh sách các giao dịch bằng phương thức ListTrades. Để đơn giản, chúng ta bỏ qua bất kỳ vị trí mở nào có thể còn lại vào cuối giai đoạn phân tích. Sự thay đổi duy nhất đối với hệ thống giao dịch là việc thêm dừng lỗ tối đa (dòng ApplyStop).

Kết luận

Giao diện lập trình backtester danh mục mới cung cấp khả năng thêm các thống kê do người dùng định nghĩa dưới mọi hình thức, cho phép người dùng nâng cao phân tích kết quả kiểm tra về một cấp độ hoàn toàn mới.

Sử dụng các hàm đồ họa mức thấp

Giao diện đồ họa AFL hoàn toàn mới với các hàm đồ họa mức thấp cho phép linh hoạt hoàn toàn trong việc tạo bất kỳ loại hiển thị tùy chỉnh nào.

Giao diện này mô phỏng gần như chính xác API GDI của Windows, với cùng tên cho hầu hết các hàm để dễ sử dụng cho các lập trình viên đã quen thuộc với GDI. Sự khác biệt duy nhất là:

1. So với GDI của Windows, tất cả các hàm đều được tiền tố bằng 'Gfx'.
2. Việc tạo/ chọn bút, cọ, phông chữ được đơn giản hóa để dễ sử dụng hơn và bạn không cần phải lo lắng về việc xóa các đối tượng GDI.
3. Có ba chế độ chồng lớp để bạn có thể kết hợp đồ họa mức thấp với các câu lệnh Plot() thông thường:
    - Chế độ = 0 (mặc định) - chồng đồ họa mức thấp lên trên biểu đồ,
    - Chế độ = 1 - chồng biểu đồ lên trên đồ họa mức thấp,
    - Chế độ = 2 - chỉ vẽ đồ họa mức thấp (không có biểu đồ/ lưới/ tiêu đề thông thường).

Tất cả các hàm sử dụng PIXELS làm tọa độ (khi sử dụng trên màn hình). Đối với bản in và metafiles, pixel được ánh xạ tới các đơn vị logic để khớp với độ phân giải cao hơn của máy in. Sử dụng Status("pxwidth") và Status("pxheight") để tìm kích thước pixel của bề mặt vẽ.

Các hàm gfx mức thấp có sẵn (nhấp vào các liên kết để xem giải thích chi tiết):

- GfxMoveTo(x, y)
- GfxLineTo(x, y)
- GfxSetPixel(x, y, color)
- GfxTextOut("text", x, y)
- GfxSelectPen(color, width = 1, penstyle = penSolid)
- GfxSelectSolidBrush(color)
- GfxSelectFont("facename", pointsize, weight = fontNormal, italic = False, underline = False, orientation = 0)
- GfxRectangle(x1, y1, x2, y2)
- GfxRoundRect(x1, y1, x2, y2, x3, y3)
- GfxPie(x1, y1, x2, y2, x3, y3, x4, y4)
- GfxEllipse(x1, y1, x2, y2)
- GfxCircle(x, y, radius)
- GfxChord(x1, y1, x2, y2, x3, y3, x4, y4)
- GfxArc(x1, y1, x2, y2, x3, y3, x4, y4)
- GfxPolygon(x1, y1, x2, y2, ...)
- GfxPolyline(x1, y1, x2, y2, ...)
- GfxSetTextColor(color)
- GfxSetTextAlign(align)
- GfxSetBkColor(color)
- GfxSetBkMode(bkmode)
- GfxGradientRect(x1, y1, x2, y2, fromcolor, tocolor)
- GfxDrawText("text", left, top, right, bottom, format = 0)
- GfxSetOverlayMode(mode = 0)

CÁC HÀM MỚI TRONG 5.80

- GfxSetCoordsMode(mode) - cho phép chọn giữa chế độ pixel và chế độ bar/price.
- GfxSetZOrder(layer) - cho phép đặt đồ họa Gfx trên lớp Z-axis (độ sâu) do người dùng chỉ định.
- GfxGetTextWidth("text") - trả về chiều rộng pixel của chuỗi đã chỉ định. LƯU Ý: điều này chậm vì nó phải tạo DC tạm thời và phông chữ để đo chiều dài văn bản. Nó mất 40us (micro giây), tức là khoảng 40 lần so với các hàm Gfx khác.

CÁC CHẾ ĐỘ TỌA ĐỘ

Bắt đầu từ phiên bản 5.80, AmiBroker hỗ trợ sử dụng hai chế độ tọa độ khác nhau trong đồ họa mức thấp: chế độ pixel và chế độ bar/price.

Hàm GfxSetCoordsMode cho phép chuyển đổi hệ tọa độ cho các hàm gfx mức thấp từ pixel trên màn hình (chế độ = 0) - mặc định, sang chế độ bar/price (chế độ = 1) trong đó X được diễn đạt bằng chỉ số bar và Y được diễn đạt bằng giá. Chế độ mới này cho phép việc chồng lớp dễ dàng hơn trên các biểu đồ hiện có mà không cần chuyển đổi giữa các pixel bar/price và không cần làm mới bổ sung thường thấy trong các phiên bản cũ khi thay đổi thang đo Y.

Hàm này có thể được gọi để chuyển đổi qua lại giữa chế độ pixel -> chế độ bar/price và ngược lại nhiều lần, cho phép kết hợp các chế độ khác nhau trong cùng một biểu đồ.

Khi chế độ tọa độ 1 được chọn (bar/price), các tọa độ có thể là số phân số. Ví dụ, nếu x là 2.5, điều này có nghĩa là ở giữa bar 2 và 3.

Ví dụ:

```
// Ví dụ cho thấy cách sử dụng GfxSetCoordsMode( 1 )

// kết quả là

// a) mã hóa dễ dàng hơn cho các biểu đồ chồng lớp trên biểu đồ tích hợp (không cần chuyển đổi từ bar/price sang pixel)

// b) sự khớp hoàn hảo giữa Plot() tích hợp và vị trí Gfx

Plot( C, "Giá", colorDefault, styleLine );

GfxSetOverlayMode( 1 );

GfxSetCoordsMode( 1 ); // chế độ bar/price (thay vì pixel)



GfxSelectSolidBrush( colorRed );

GfxSelectPen( colorRed );
```

```
boxheight = 0.01 * ( HighestVisibleValue( C ) - LowestVisibleValue( C ) );
bi = BarIndex();
start = FirstVisibleValue( bi );
end = LastVisibleValue( bi );

for ( i = start; i <= end; i++ )
{
    Cl = Close[ i ];
    Op = Open[ i ];

    Color = IIf( Cl > Op, colorGreen, colorRed );
    GfxSelectPen( Color );
    GfxSelectSolidBrush( Color );

    bodyup = Max( Op, Cl );
    bodydn = Min( Op, Cl );

    GfxEllipse( i - 0.4, bodyup, i + 0.4, bodydn );

    GfxMoveTo( i, H[ i ] );
    GfxLineTo( i, bodyup );
    GfxMoveTo( i, bodydn );
    GfxLineTo( i, L[ i ] );
}
```

HỖ TRỢ THỨ TỰ Z

Bắt đầu từ phiên bản 5.80, AmiBroker hỗ trợ việc đặt đồ họa mức thấp trong các lớp thứ tự Z khác nhau. Một hàm mới GfxSetZOrder được cung cấp để cho phép điều này như trong ví dụ sau:

```
Plot( C, "Giá", colorDefault );

GraphGridZOrder = 1;


GfxSetZOrder(0);

GfxSelectSolidBrush( colorGreen );

GfxCircle( 100, 100, 100 );


GfxSetZOrder(-1);

GfxSelectSolidBrush( colorRed );

GfxCircle( 150, 150, 100 );


GfxSetZOrder(-2);

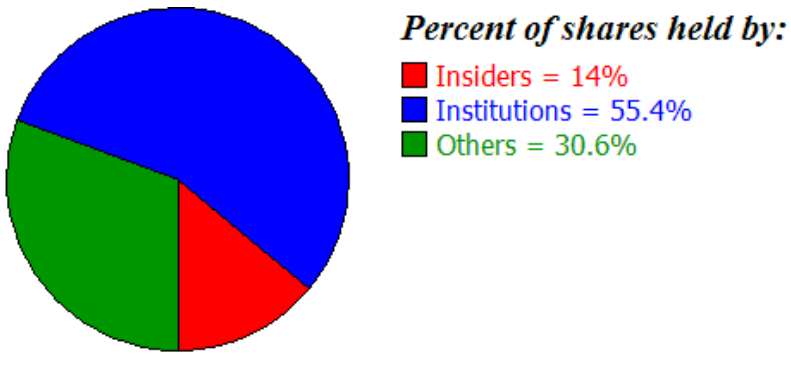
GfxSelectSolidBrush( colorBlue );

GfxCircle( 180, 180, 100 );
```

Ví dụ sử dụng:

Ví dụ 1. Biểu đồ hình tròn thể hiện tỷ lệ sở hữu của các loại cổ đông khác nhau.

Dưới đây là cách nó trông như thế nào:



Công thức:

```
// OverlayMode = 2 có nghĩa là chỉ có
// đồ họa mức thấp được vẽ
// sẽ không có lưới, không có tiêu đề, không có biểu đồ
// và chỉ có những gì chúng ta mã hóa bằng các cuộc gọi Gfx*
GfxSetOverlayMode(2);

HInsiders = GetFnData("InsiderHoldPercent");
HInst = GetFnData("InstitutionHoldPercent");

function DrawPiePercent( x, y, radius, startpct, endpct )
{
    PI = 3.1415926;
    sa = 2 * PI * startpct / 100;
    ea = 2 * PI * endpct / 100;
    xsa = x + radius * sin( sa );
    ysa = y + radius * cos( sa );
    xea = x + radius * sin( ea );
    yea = y + radius * cos( ea );

    GfxPie( x - radius, y - radius, x + radius, y + radius, xsa, ysa, xea, yea );
}

radius = 0.45 * Status("pxheight"); // lấy chiều cao pixel của biểu đồ và sử dụng 45%
cho bán kính biểu đồ hình tròn
textoffset = 2.4 * radius;
GfxSelectSolidBrush( colorRed );
DrawPiePercent( 1.1*radius, 1.1*radius, radius, 0, HInsiders );
GfxRectangle( textoffset , 42, textoffset +15, 57 );
```

```
GfxSelectSolidBrush( colorBlue );
DrawPiePercent( 1.1*radius, 1.1*radius, radius, HInsiders, HInst + HInsiders );
GfxRectangle( textoffset , 62, textoffset +15, 77 );
GfxSelectSolidBrush( colorGreen );
DrawPiePercent( 1.1*radius, 1.1*radius, radius, HInst + HInsiders, 100 );
GfxRectangle( textoffset , 82, textoffset +15, 97 );

GfxSelectFont("Times New Roman", 16, 700, True );
GfxTextOut("Tỷ lệ cổ phiếu do:", textoffset , 10 );
GfxSelectFont("Tahoma", 12 );
GfxSetTextColor( colorRed );
GfxTextOut( "Cổ đông = " + HInsiders + "%", textoffset + 20, 40 );
GfxSetTextColor( colorBlue );
GfxTextOut( "Tổ chức = " + HInst + "%", textoffset + 20, 60 );
GfxSetTextColor( colorGreen );
GfxTextOut( "Khác = " + ( 100 - (HInst+HInsiders) ) + "%", textoffset + 20, 80 );
GfxSelectFont("Tahoma", 8 );
```

Ví dụ 2. Mẫu đầu ra định dạng (giống như bảng) sử dụng các hàm gfx mức thấp.

```
// mẫu đầu ra văn bản định dạng qua các hàm gfx mức thấp

CellHeight = 20;
CellWidth = 100;
GfxSelectFont( "Tahoma", CellHeight/2 );

function PrintInCell( string, row, Col )
{
```

```
    GfxDrawText( string, Col * CellWidth, row * CellHeight, (Col + 1 ) * CellWidth, (row
+ 1 ) * CellHeight, 0 );
}

PrintInCell( "Mở", 0, 0 );
PrintInCell( "Cao", 0, 1 );
PrintInCell( "Thấp", 0, 2 );
PrintInCell( "Đóng", 0, 3 );
PrintInCell( "Khối lượng", 0, 4 );

GfxSelectPen( colorBlue );
for( i = 1; i < 10 && i < BarCount; i++ )
{
    PrintInCell( StrFormat("%g",    O[    i    ]    ),    i,    0    );
    PrintInCell( StrFormat("%g",    H[    i    ]    ),    i,    1    );
    PrintInCell( StrFormat("%g",    L[    i    ]    ),    i,    2    );
    PrintInCell( StrFormat("%g",    C[    i    ]    ),    i,    3    );
    PrintInCell( StrFormat("%g",    V[    i    ]    ),    i,    4    );
    GfxMoveTo( 0, i * CellHeight );
    GfxLineTo( 5 * CellWidth, i * CellHeight );
}
GfxMoveTo( 0, i * CellHeight );
GfxLineTo( 5 * CellWidth, i * CellHeight );

for( Col = 1; Col < 6; Col++ )
{
    GfxMoveTo( Col * CellWidth, 0);
```

```
    GfxLineTo( Col * CellWidth, 10 * CellHeight );
}

Title="";
```

Ví dụ 3. Demo đồ họa mức thấp với phần hình tròn, đa giác, bánh xe màu, văn bản động và chồng biểu đồ.

```
// chế độ chồng lớp = 1 có nghĩa là
// đồ họa mức thấp nên nằm ở nền
GfxSetOverlayMode(1);

Plot(C, "Đóng", colorBlack, styleCandle );
PI = 3.1415926;
k = (GetPerformanceCounter()/100)%256;
for( i = 0; i < 256; i++ )
{
    x = 2 * PI * i / 256;

    GfxMoveTo( 100+k, 100 );
    GfxSelectPen( ColorHSB( ( i + k ) % 256, 255, 255 ), 4 );
    GfxLineTo( 100 +k+ 100 * sin( x ), 100 + 100 * cos( x ) );
}

GfxSelectFont("Tahoma", 20, 700 );
GfxSetBkMode(1);
GfxSetTextColor(colorBrown);
GfxTextOut("Kiểm tra khả năng đồ họa", 20, 128-k/2 );
```

```
GfxSelectPen( colorRed );

GfxSelectSolidBrush( colorBlue );

GfxChord(100,0,200,100,150,0,200,50);


//GfxPie(100,0,200,100,150,0,200,50);

GfxSelectPen( colorGreen, 2 );

GfxSelectSolidBrush( colorYellow );

GfxPolygon(250,200,200,200,250,0,200,50);


RequestTimedRefresh(1);
```

Ví dụ 4. Định vị đồ họa mức thấp - cho thấy cách căn chỉnh các biểu đồ tích hợp Plot() với đồ họa mức thấp. Lưu ý rằng nếu thang đo thay đổi (chiều cao pixel thay đổi) do dữ liệu mới hoặc mức thu phóng khác, cần phải làm mới bổ sung để đọc thang đo mới và điều chỉnh vị trí một cách chính xác.

plaintext

Sao chép mã

```
Plot(C, "Giá", colorBlack, styleLine );

GfxSetOverlayMode(0);


Miny = Status("axisminy");

Maxy = Status("axismaxy");


lvb = Status("lastvisiblebar");

fvb = Status("firstvisiblebar");


pxwidth = Status("pxwidth");

pxheight = Status("pxheight");
```

```
TotalBars = lvb - fvb;


axisarea = 56; // có thể cần điều chỉnh nếu bạn sử dụng phông chữ không phải mặc định cho trục

GfxSelectSolidBrush( colorRed );


GfxSelectPen( colorRed );

for( i = 0; i < TotalBars AND i < ( BarCount - fvb ); i++ )

{

    x = 5 + i * (pxwidth - axisarea - 10) / ( TotalBars + 1 );


    y = 5 + ( C[ i + fvb ] - Miny ) * ( pxheight - 10 )/ ( Maxy - Miny );


    GfxRectangle( x - 1, pxheight - y - 1, x + 2, pxheight - y + 2);

}
```

Thông tin kỹ thuật

• Hướng dẫn khắc phục sự cố
• Tập tin sử dụng bởi AmiBroker
• Hệ thống phục hồi sự cố và báo cáo lỗi tự động
• Mẹo tối ưu hiệu suất
• Các tính năng mới trong AmiBroker?

Hướng dẫn khắc phục sự cố

Nhảy nhanh: AmiQuote, các plugin dữ liệu (eSignal, myTrack, IQFeed, QuoteTracker, Quotes Plus, TC2000)

DANH MỤC: SỰ CỐ NGHIÊM TRỌNG HOẶC ĐÌNH TRỆ

Nơi xảy ra	Vấn đề	Nguyên nhân	Giải pháp
AmiBroker, **Chỉ sau khi cài đặt**	AmiBroker treo tại khởi động hoặc xuất hiện lỗi thường xuyên	Thành phần hệ thống cũ/không tương thích	Tải xuống và cài đặt phiên bản FULL mới nhất của AmiBroker từ http://www.amibroker.com/download.html
AmiBroker, **Tại khởi động** *(trước đây vẫn hoạt động bình thường)*	AmiBroker gặp sự cố hoặc treo	Hỏng tệp dữ liệu nào đó	Xin vui lòng thử các bước sau: • Đổi tên thư mục cơ sở dữ liệu mặc định và thử khởi động AmiBroker. Nếu nó hoạt động, điều đó có nghĩa là một số tệp bên trong cơ sở dữ liệu này đã bị hỏng. Bạn có thể gửi tệp DEFAULT.AWL từ thư mục "Layouts" để chúng tôi kiểm tra. • Xóa hoặc đổi tên tất cả các tệp DEFAULT.AWL mà bạn tìm thấy trên ổ đĩa và thử khởi động AmiBroker. (Các tệp .AWL được tạo ra trong thư mục "Layouts" của cơ sở dữ liệu chính AmiBroker và các thư mục "Layout" của mỗi thư mục cơ sở dữ liệu.) • Đổi tên các tệp broker.charts và broker.bcharts (trong thư mục AmiBroker) và thử khởi động AmiBroker. • Đổi tên tệp default.layout (trong thư mục AmiBroker) và thử khởi động AmiBroker.
AmiBroker,	AmiBroker gặp sự cố hoặc treo	Lỗi trong công thức AFL hoặc trong AmiBroker	Gửi mã gây ra sự cố cho bộ phận hỗ trợ của AmiBroker (bugs@amibroker.com) và xóa / sửa đổi / bình luận ra để tiếp tục làm việc cho đến khi chúng tôi tìm ra nguyên nhân và giải pháp cho vấn đề.

Khi chạy một số mã AFL			

DANH MỤC: KHÁC

Nơi xảy ra	Vấn đề	Nguyên nhân	Giải pháp
AmiBroker, **Chỉ sau khi cài đặt**	Hướng dẫn sử dụng (Help) không thể truy cập	Hệ thống HTMLHelp chưa được cài đặt	Cài đặt bản cập nhật cho hệ thống HTML Help có sẵn từ Microsoft tại đây
	Kịch bản không hoạt động	Windows Scripting Host chưa được cài đặt	Kiểm tra xem phần mềm diệt virus của bạn có đang chặn scripting hay không.
AmiBroker, **Tại khởi động** *(trước đây vẫn hoạt động bình thường)*	Một số tính năng không hoạt động	Thiếu thành phần của AmiBroker	Vui lòng kiểm tra xem bạn có vô tình xóa bất kỳ tệp quan trọng nào của AmiBroker không. Nếu không chắc chắn, bạn có thể chạy lại thiết lập trên phiên bản cài đặt cũ (xin đừng gỡ cài đặt nếu bạn muốn giữ lại các cài đặt của mình)
AmiQuote - **Tải xuống báo giá**	**Tải xuống từ Quote.com thất bại với Lỗi:** 'Kết nối với máy chủ đã bị đặt lại'	Mật khẩu Livecharts không chính xác hoặc không được nhập đúng như mô tả ở đây.	Vui lòng truy cập trang CÀI ĐẶT AmiQuote, BỎ ĐÁNH DẤU "Sử dụng tài khoản livecharts" và chọn "MÁY CHỦ #1" hoặc làm theo chính xác hướng dẫn ở đây.
	Tải xuống từ Yahoo bắt đầu tốt nhưng sau đó dừng lại *(đặc biệt trên các kết nối DSL,*	Yahoo đang chặn bạn. Sau cuộc tấn công sâu rộng từ worm trên Internet, Yahoo coi việc tải xuống nhanh, lặp đi	Nếu có thể, hãy kết nối lại với địa chỉ IP khác (nếu không, bạn sẽ cần chờ khoảng nửa giờ hoặc lâu hơn cho đến khi Yahoo mở khóa cho bạn). Truy cập vào phần "Cài đặt" của AmiQuote và đặt "Thời gian trì hoãn ban đầu giữa các yêu cầu" là 1000

	ADSL, cáp modem)	lặp lại như một cuộc tấn công "từ chối dịch vụ"	và "số lượng tải xuống đồng thời tối đa" là 1.
	Tải xuống từ Yahoo thất bại	Có thể quá sớm để lấy dữ liệu lịch sử từ Yahoo hoặc gặp sự cố tạm thời với Yahoo.	Đầu tiên, hãy kiểm tra trang lịch sử của Yahoo: http://table.finance.yahoo.com/k?s=utx&g=d (thay thế "utx" bằng mã chứng khoán liên quan). Nếu trang hiển thị các báo giá cũ, thì đó là vấn đề của Yahoo, không phải của AmiQuote. Thực tế, AmiQuote sử dụng liên kết "Tải xuống định dạng bảng tính" trên trang đã đề cập: http://table.finance.yahoo.com/k?s=utx&g=d. Nếu dữ liệu lịch sử không có sẵn, bạn luôn có thể sử dụng chế độ "Hiện tại" của AmiQuote để lấy dữ liệu ngày hôm nay (ngay cả trong giờ giao dịch). Nếu bạn không biết mã chứng khoán cho chỉ số, cổ phiếu hoặc quỹ tương hỗ, hãy sử dụng tính năng tra cứu mã: a) tại Yahoo (cho các chế độ lịch sử và hiện tại): http://finance.yahoo.com/l. b) tại Lycos (cho chế độ trong ngày): http://finance.lycos.com/home/misc/symbol_search.asp.
	Tải xuống thành công nhưng không có báo giá nào được thêm	Vấn đề cấu hình cục bộ	Làm theo các bước sau để khắc phục sự cố: • Truy cập vào C:\Program Files\AmiBroker\AmiQuote\Download và kiểm tra xem có các tệp .AQH hay không. • Mở chúng bằng Notepad và xem

			dữ liệu có sẵn hay không. • Kiểm tra xem bạn có tệp aqh.format trong thư mục Formats của AmiBroker không. • Kiểm tra nội dung của nó và nếu cần, hãy gửi cho tôi để kiểm tra. • Thử nhập dữ liệu thủ công theo hướng dẫn trong hướng dẫn sử dụng. • Nếu bạn đã khôi phục thư mục dữ liệu từ CD ROM, hãy kiểm tra cờ "chỉ đọc" của các tệp dữ liệu trong cơ sở dữ liệu AmiBroker. Chúng không được để ở chế độ chỉ đọc. Nếu có, bạn cần đánh dấu tất cả các tệp trong thư mục dữ liệu và tất cả các thư mục con và bỏ đánh dấu cờ chỉ đọc
	Tải xuống từ bất kỳ nguồn nào thất bại	Vấn đề kết nối	Vui lòng kiểm tra kết nối Internet của bạn. Nếu bạn đang sử dụng tường lửa, hãy đảm bảo cho phép AmiQuote (QUOTE.EXE) truy cập Internet trên các cổng 80 và 443.
AmiBroker /eSignal plugin	Dữ liệu RT bị đột biến, ticks không chính xác, và các vấn đề dữ liệu khác.	Vấn đề kết nối hoặc lý do khác	Nhấp chuột phải vào khu vực trạng thái plugin (ô màu xanh "OK") và chọn "Fixup data for symbol", sau đó chờ một chút. Điều này sẽ khiến toàn bộ lịch sử intraday cho mã chứng khoán đó được tải lại.
AmiBroker /myTrack /IQFeed /QuoteTracker plugin	Dữ liệu RT bị đột biến, ticks không chính xác, và các vấn đề dữ liệu khác.	Vấn đề kết nối hoặc lý do khác	Nhấp chuột phải vào ô màu xanh "OK" và chọn "Force backfill", sau đó chờ một chút. Điều này sẽ khiến toàn bộ lịch sử intraday cho mã chứng khoán đó được tải lại.
AmiBroker /QuoteTracker plugin	Dữ liệu không được cập nhật.	Không đủ 'nhấp chuột quảng cáo' trong QuoteTracker.	Bạn cần nhấp vào quảng cáo trong QuoteTracker hoặc đăng ký. Để biết thêm chi tiết, xem tại: http://www.amibroker.com/qthelp.html.
AmiBroker	Mất kết nối.	Vấn đề kết nối.	Nhấp chuột phải vào khu vực trạng thái plugin (sẽ hiển thị màu đỏ

/bất kỳ plugin RT nào			"WAIT" hoặc "SHUT") và chọn "Disconnect" rồi "Connect". Nếu điều này không giúp, hãy khởi động lại AmiBroker.
AmiBroker /QuotesPlus plugin	Thông báo lỗi "Cannot initialize C-TREE".	Vấn đề với cơ sở dữ liệu Quotes Plus.	Kiểm tra xem chương trình biểu đồ của Quotes Plus có hoạt động không. Nếu không, điều này có nghĩa là có vấn đề trong module cơ sở dữ liệu QP2 và bạn cần liên hệ với bộ phận hỗ trợ kỹ thuật của Quotes Plus để được giúp đỡ.
AmiBroker /TC2000 /TCNet plugin, thỉnh thoảng...	**Thông báo lỗi** "Biến hoặc đối tượng không được thiết lập"	**Vấn đề thời gian** trong TC2000 các thành phần của riêng bạn.	Giải pháp tạm thời duy nhất cho vấn đề này là di chuyển tất cả dữ liệu từ đĩa CD-ROM TC2000 vào ổ cứng của bạn (bằng danh sách ổ cứng).

Câu hỏi khác nhau

Câu hỏi	Trả lời
Tôi có phải sử dụng Internet Explorer không? Tôi có thể sử dụng Netscape Navigator, Opera hoặc trình duyệt khác thay thế không?	Bạn có thể sử dụng bất kỳ trình duyệt nào. Trong một số phiên bản máy chủ của Windows, nó phải được chọn để cài đặt. Internet Explorer hiện chỉ là một phần của hệ điều hành và AmiBroker sử dụng rất nhiều các phần đó. Nhưng bạn không cần phải sử dụng Internet Explorer.
Tệp trợ giúp (hướng dẫn) ở đâu?	Hướng dẫn người dùng có sẵn từ menu Trợ giúp -> Các chủ đề trợ giúp trong AmiBroker (cũng có thể truy cập bằng phím F1).

Tệp và thư mục được sử dụng bởi AmiBroker

Thư mục chính của AmiBroker - thư mục nơi bạn đã cài đặt AmiBroker. Bạn có thể tìm thấy tệp thực thi chính của AmiBroker (BROKER.EXE) ở đó.

Tệp chương trình

- **Broker.exe** - tệp ứng dụng chính
- **CoolTool.dll, MiscTool.dll, Brokey.dll** - các tệp ứng dụng bổ sung cần thiết cho broker.exe
- **HTMLView.exe** - trình xem báo cáo, được sử dụng để hiển thị báo cáo kiểm tra lại
- **emailer.exe, emailerssl.exe** - các tiện ích dòng lệnh nhỏ gửi cảnh báo do người dùng định nghĩa qua email
- **DBCAPI.DLL** - (chỉ xuất hiện nếu phiên bản RT được cài đặt) - thư viện hỗ trợ eSignal DBCAPI
- **Plugins** - thư mục con chứa tất cả các DLL của plug-in.
- **unins000.exe và unins000.dat** - chương trình gỡ cài đặt.
- **Broker.chm** - tệp HELP đã biên dịch

Tệp dữ liệu

- **Data** - thư mục cơ sở dữ liệu mặc định, chứa broker.master, broker.workspace, các thư mục con dữ liệu từ 0-9, A-Z, và ký hiệu '_', cùng với thư mục Layouts. Đọc phần này để tìm hiểu thêm về các khái niệm cơ sở dữ liệu của AmiBroker.
- **broker.master** - tệp dữ liệu nhị phân chứa bảng tất cả các ký hiệu có sẵn trong cơ sở dữ liệu. Được sử dụng để tải nhanh các ký hiệu. Bảng ký hiệu bao gồm thông tin về phân loại ký hiệu thành các danh mục (thị trường, nhóm, lĩnh vực, ngành, danh sách theo dõi, chỉ số và yêu thích). Nếu bạn xóa tệp này, AmiBroker sẽ tái tạo lại nó vì thông tin này cũng có trong các tệp dữ liệu ký hiệu riêng lẻ.
- **broker.workspace** - tệp dữ liệu nhị phân chứa thông tin về cài đặt cơ sở dữ liệu (khoảng thời gian, nguồn dữ liệu được sử dụng, v.v.), tên danh mục, dữ liệu tăng/giảm toàn cầu, v.v. Nếu bạn xóa tệp này, bạn sẽ mất cài đặt cơ sở dữ liệu và tên danh mục sẽ được đặt lại về mặc định.
- **default.awl, .awl* - tệp bố cục không gian làm việc của AmiBroker. Tệp văn bản chứa thông tin về các bố cục và trang biểu đồ. Tệp default.awl được lưu trong thư mục "Layouts" bên trong thư mục cơ sở dữ liệu chứa bố cục LOCAL mặc định cho cơ sở dữ liệu này. Tệp default.awl được lưu trong thư mục "Layouts" của thư mục chính AmiBroker chứa bố cục GLOBAL mặc định (sử dụng khi không có bố cục địa phương nào). Nếu bạn xóa tệp này, bố cục mặc định sẽ được tạo lại dựa trên tệp default.layout được cung cấp cùng với AmiBroker.
- **broker.newcharts** - tệp nhị phân chứa tham chiếu giữa các bố cục biểu đồ và các tệp công thức được sử dụng. Nếu bạn xóa nó, bạn sẽ thấy thông báo "tệp công thức rỗng hoặc không thể tìm thấy" trong các biểu đồ của mình vì tham chiếu giữa ID biểu đồ và công thức bị mất. Tuy nhiên, bạn sẽ không mất công thức của mình vì công thức là tách biệt. Bạn sẽ có thể chèn lại chúng vào các biểu đồ của mình.
- **broker.layers** - tệp văn bản chứa thông tin về các lớp biểu đồ. Nếu bạn xóa tệp này, các lớp sẽ được đặt lại về mặc định từ nhà máy.

- **broker.groups, broker.markets, broker.sectors, broker.industries** - các tệp văn bản chứa tên mặc định cho các nhóm, thị trường, lĩnh vực và ngành. Chỉ được sử dụng tại thời điểm tạo cơ sở dữ liệu. Sau đó, thông tin này được lưu per-database trong tệp broker.workspace. Nếu bạn xóa chúng, AmiBroker sẽ mặc định vào tên nhóm n, thị trường n, lĩnh vực n, ngành n, trong đó n là 1...256.
- **broker.prefs** - tệp nhị phân chứa cài đặt tùy chọn của người dùng (có sẵn từ Tools->Preferences). Nếu bạn xóa tệp này, AmiBroker sẽ đặt lại về cài đặt mặc định từ nhà máy.
- **broker.params -** tệp văn bản chứa thông tin bền vững về các tham số chỉ báo do người dùng định nghĩa.

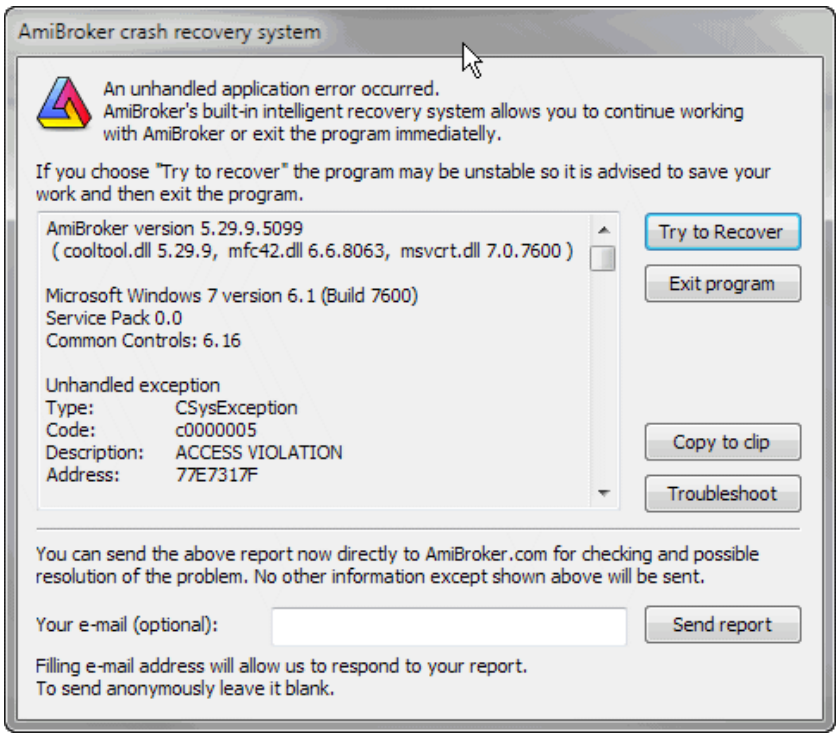
Hệ thống phục hồi sự cố và báo cáo lỗi tự động

AmiBroker có một hệ thống phát hiện và báo cáo lỗi được gọi là "Hệ thống phục hồi sự cố". Tên cho thấy AmiBroker hiện có thể phục hồi sau những tình huống bất ngờ như vậy và thực sự có thể!.

Làm thế nào để thực hiện được điều này? Vâng... cần một số thủ thuật để gói cơ chế xử lý ngoại lệ được Windows sử dụng :-)

Thông thường, khi ứng dụng Windows thực hiện một số truy cập bộ nhớ bất hợp pháp, hoạt động bất hợp pháp (ví dụ chia cho số không) hoặc lệnh bất hợp pháp, hệ thống sẽ bật hộp thông báo ngõ cụt với nội dung "Chương trình này đã thực hiện một hoạt động bất hợp pháp và sẽ bị tắt". Bây giờ bạn không còn lựa chọn nào khác - ứng dụng sẽ bị chấm dứt khi bạn nhấp vào nút OK.

Hệ thống phục hồi sự cố của AmiBroker được giới thiệu trong phiên bản beta v3.47 sẽ chặn ngoại lệ do Windows tạo ra và thay vì hộp thông báo ngõ cụt tiêu chuẩn, nó sẽ hiển thị hộp thoại sau:



Như bạn thấy, có một cửa sổ hiển thị thông tin hệ thống quan trọng và có năm nút Thử phục hồi, Thoát chương trình, Sao chép vào clip, Khắc phục sự cố, Gửi báo cáo. Nhấp vào nút Thoát chương trình hoạt động giống hệt như nhấp vào nút "OK" trong hộp thông báo ngõ cụt chuẩn của Windows. Nhưng hai nút đầu tiên cung cấp

cho bạn những khả năng hoàn toàn mới. Nếu bạn nhấp vào nút Thử khôi phục, AmiBroker sẽ cố gắng khôi phục sau lỗi và tiếp tục chạy. Trong hầu hết các trường hợp, bạn sẽ có thể lưu công việc và các sửa đổi đã thực hiện để không mất bất kỳ thứ gì. Trên thực tế, bạn sẽ có thể làm việc bình thường. Tuy nhiên, có một số trường hợp khôi phục không thành công và AmiBroker có thể không ổn định, vì vậy bạn chỉ nên lưu dữ liệu và thoát. Cũng có thể xảy ra trường hợp cửa sổ này sẽ bật lên một vài lần - sau đó bạn chỉ cần nhấp vào Thử khôi phục nhiều lần. Sao chép vào clip - sao chép chi tiết báo cáo lỗi có thông tin hệ thống vào bảng tạm để bạn có thể dán thông tin này vào chương trình email và gửi cho chúng tôi.

Chức năng khôi phục khá hay nhưng mục đích chính của hệ thống này là tìm và khắc phục sự cố trong phiên bản tương lai và đây là lý do tại sao chức năng quan trọng nhất được cung cấp - Gửi báo cáo. Nếu cửa sổ phục hồi sự cố

hiện ra trên màn hình, vui lòng nhấp vào nút Gửi báo cáo trước khi cố gắng tiếp tục làm việc. Thao tác này sẽ tự động gửi thông tin chi tiết được hiển thị trong cửa sổ phục hồi sự cố cho chúng tôi. Vui lòng thêm email của bạn (trong trường Email của bạn) để chúng tôi có thể phản hồi báo cáo của bạn. Nếu bạn không cung cấp email, báo cáo sẽ được gửi ẩn danh, nhưng chúng tôi sẽ không thể phản hồi và cung cấp cho bạn bất kỳ hướng dẫn nào.

AmiBroker phiên bản 5.29.9.5099

(cooltool.dll 5.29.9, mfc42.dll 6.6.8063, msvcrt.dll 7.0.7600)

Microsoft Windows 7 phiên bản 6.1 (Bản dựng 7600)

Gói dịch vụ 0.0

Các điều khiển chung: 6.16

Ngoại lệ chưa xử lý Loại: CSysException Mã: c0000005

Mô tả: VI PHẠM QUYỀN TRUY CẬP Địa chỉ: 77E7317F

Graph0=C;

Sum(C,-C)

- ^

Lỗi 47.

Ngoại lệ xảy ra trong quá trình thực thi công thức AFL tại địa chỉ: 77E7317F, mã: C0000005

Thông tin ngoại lệ chi tiết:

Broker.exe đã gây ra EXCEPTION_ACCESS_VIOLATION trong mô-đun ntdll.dll tại 0023:77E7317F, RtlImageNtHeader()+0411 byte

Call Stack:

0023:77E7317F ntdll.dll, RtlImageNtHeader()+0411 byte 0023:77E73407 ntdll.dll, RtlImageNtHeader()+1059 byte 0023:77E732F2 ntdll.dll, RtlImageNtHeader()+0782 byte(s)

CPU Registers:

EAX=071BD978 EBX=071BCDA8 ECX=00000000 EDX=00000000 ESI=071BD970 EDI=071B0000 EBP=00000000 ESP=0018F490 EIP=76D18FBA FLG=00010246 CS=0023 DS=002B SS=002B ES=002B FS=0053 GS=002B

Trạng thái bộ phân tích cú pháp AFL:

Giai đoạn xử lý: EXCEPTION Công thức ID: 1995 (Không có tên 190)

Hành động 1 (CHỈ BÁO)

Thông tin bổ sung:

Số lượng cổ phiếu đã tải: 35 Cổ phiếu hiện đang chọn: MCD

Số lượng báo giá (cổ phiếu hiện tại): 751

Không gian làm việc:

Nguồn dữ liệu = (mặc định), Chế độ cục bộ dữ liệu = 1, NumBars = 250

Tùy chọn:

Nguồn dữ liệu = (cục bộ), Chế độ cục bộ dữ liệu = 1, NumBars = 1000

Lịch sử lệnh:

2783 - Cài đặt tùy chọn--Tùy chọn

2824 - Hiển thị trình chỉnh sửa công thức AFL--Trình chỉnh sửa công thức

Thống kê trình quản lý bộ đệm:

Số lượng phần tử danh sách: 1 Số lượng phần tử bản đồ: 1 Kích thước bảng băm: 5987

Trạng thái bộ nhớ:

MemoryLoad: 25 %

TotalPhys: 4194303K AvailPhys: 4194303K TotalPageFile: 4194303K AvailPageFile: 4194303K TotalVirtual: 4194176K AvailVirtual: 4029976K

Tin nhắn Windows cuối cùng:

HWnd: 0x13075e Tin nhắn: 0x0110 wParam: 0x0025092c

lParam: 0x00000000

Như bạn thấy, AmiBroker tự động tạo ra những chi tiết quan trọng nhất cho báo cáo lỗi, bao gồm cả một số lịch sử lựa chọn menu (Lịch sử lệnh), nhưng điều cần thiết nhất lúc này là cung cấp mô tả về các bước cần thực hiện để tái tạo lỗi. Nếu bạn có thể gửi cho chúng tôi một email với mô tả các bước cần thiết để tái hiện vấn đề (những gì bạn đã làm trước khi lỗi xảy ra, các điều kiện đặc biệt cần được đáp ứng để tái tạo lỗi, có thể là công thức AFL mà bạn đã thử và bất kỳ điều gì mà bạn cho là quan trọng), điều này sẽ rất hữu ích (mặc dù AmiBroker đã bao gồm một vài dòng công thức gây lỗi tự động).

Điều này là rất quan trọng vì thông tin tự động tạo ra là rất tốt, nhưng không thể bao quát hết tất cả các chi tiết.

Nhấn vào nút Khắc phục sự cố sẽ đưa bạn đến trang Khắc phục sự cố tại http://www.amibroker.com/troubleshoot.html, nơi có mô tả về những vấn đề phổ biến nhất và cách giải quyết chúng.

Một vài lưu ý cuối cùng: Tôi đã đầu tư rất nhiều công sức vào việc làm cho hệ thống này đáng tin cậy, tuy nhiên bạn nên biết rằng không phải tất cả các lỗi ngoại lệ và/hoặc lỗi hệ thống đều có thể được xử lý bởi hệ thống này và có thể xảy ra trường hợp AmiBroker không thể phục hồi từ một lỗi nghiêm trọng. Cũng có khả năng hệ thống này sẽ không thể chặn tất cả các ngoại lệ cấp thấp. Trong trường hợp đó, hãy tự chuẩn bị báo cáo, cung cấp cho tôi càng nhiều chi tiết càng tốt.

Xin hãy nhớ rằng mục tiêu cuối cùng là làm cho AmiBroker trở nên vững chắc và không có lỗi. Đây là điều mà tôi đang làm việc liên tục.

Mẹo điều chỉnh hiệu suất

AmiBroker là một trong những chương trình phân tích kỹ thuật nhanh nhất (nếu không muốn nói là nhanh nhất), tuy nhiên, người dùng áp dụng các cài đặt không chính xác, mã hóa công thức kém và các lựa chọn không tối ưu khác có thể làm chậm đáng kể chương trình. Trong chương ngắn này, chúng tôi sẽ đưa ra một số gợi ý về cách làm cho chương trình hoạt động như thiết kế ban đầu của nó.

Có ba lĩnh vực điều chỉnh hiệu suất:

1. Hệ điều hành/cấp máy

2. Cấp cài đặt AmiBroker

3. Mã hóa công thức

Ở cấp hệ điều hành, bạn nên thực hiện:

• Tránh cài đặt các chương trình của bên thứ 3 không cần thiết

Cụ thể là tránh tất cả các loại "memory turbo", "windows optimizers", "windows cleaners" và các công cụ tương tự vì chúng chỉ làm tăng thêm tải cho CPU và tài nguyên hệ thống và không mang lại lợi ích gì hoặc hầu như không mang lại lợi ích gì. Tệ hơn nữa, chúng có thể ảnh hưởng đến hoạt động bình thường của Windows và gây ra sự cố về khả năng tương thích

• TẮT kiểm tra vi-rút cho các tệp DỮ LIỆU

Nhiều phần mềm diệt vi-rút bật cái gọi là bảo vệ "trực tiếp" trên tất cả các tệp trên tất cả các đĩa. Điều này sẽ khiến hiệu suất hệ thống bị giảm xuống. Tại sao? Đơn giản là vì phần mềm diệt vi-rút sẽ chặn mọi truy cập tệp và chạy kiểm tra mở rộng trên mỗi truy cập dữ liệu. Điều này có thể làm chậm đáng kể tốc độ truy cập tệp đến mức Windows khởi động chậm hơn bình thường gấp 3 lần.

Xem bài viết sau http://www.thepcspy.com/read/what_really_slows_windows_down/5 cho thấy một số phép đo thực tế về tác động của phần mềm diệt vi-rút.

Mặc dù tôi không nói là tắt hoàn toàn phần mềm diệt vi-rút, nhưng chúng tôi khuyên bạn nên TẮT

kiểm tra phần mềm diệt vi-rút trên TỆP DỮ LIỆU, cụ thể là trên tất cả các tệp dữ liệu bên trong thư mục AmiBroker. Nói chung, có hai loại tệp: tệp thực thi (.exe, .dll, .ocx) và tệp dữ liệu (.txt, .html,

.wav, .jpg, .gif, .chm, v.v.). Việc kiểm tra tệp dữ liệu để tìm vi-rút thực sự không có ý nghĩa gì hoặc không có ý nghĩa gì cả vì chúng không chứa bất kỳ mã thực thi nào do đó không thực sự có thể là nguồn lây nhiễm. Miễn là tệp không bao gồm bất kỳ phần thực thi nào - thì tệp đó an toàn. Tuy nhiên, lưu ý rằng một số tệp dữ liệu (như Excel .xls, có thể chứa macro và macro _có_ thể_ thực thi và có thể bị nhiễm). Nhưng đối với AmiBroker, bất kỳ thứ gì bên trong thư mục AmiBroker không có phần mở rộng .exe, .dll hoặc .js đều không thể thực thi và an toàn.

• Chọn phần mềm diệt vi-rút một cách khôn ngoan

Như đã chỉ ra ở trên, một số sản phẩm diệt vi-rút có thể thực sự ngốn tài nguyên và bạn nên kiểm tra các bài kiểm tra hiệu suất (xem liên kết ở trên) trước khi quyết định sử dụng phần mềm diệt vi-rút nào.

Ở cấp độ cài đặt AmiBroker

• Tránh "Số thanh" lớn không cần thiết trong Tệp->Cài đặt cơ sở dữ liệu

Đây là lỗi mà mọi người thường mắc phải nhất. Ví dụ, tôi đã thấy 50000 được nhập vào "Số thanh" được nhập cho cơ sở dữ liệu cuối ngày. Điều này không hợp lý vì 50000 thanh hàng ngày là 192 năm!!! Không có nguồn dữ liệu nào cung cấp lịch sử như vậy và việc nhập số lượng đó là lãng phí bộ nhớ.

Bạn cần nhớ rằng bất cứ điều gì bạn nhập vào đây đều buộc AmiBroker phải thực sự phân bổ bộ nhớ cho nhiều thanh như vậy. Mỗi thanh dữ liệu là 40 byte. Vì vậy, nếu bạn nhập 100000 thanh, bạn sẽ buộc AmiBroker thực sự phân bổ 4MB bộ nhớ *cho mỗi ký hiệu*. Nghe có vẻ không quá nhiều nhưng AmiBroker lưu trữ bộ nhớ đệm dữ liệu được sử dụng gần đây và nếu bạn đã xác định bộ nhớ đệm lớn 500 ký hiệu, bạn sẽ có thể đạt được 2GB bộ nhớ theo cách đó. Vì vậy, hãy luôn xem hộp thoại Tệp->Cài đặt cơ sở dữ liệu hiển thị bên cạnh trường "Số thanh". Nó sẽ hiển thị số ngày tương đương với số thanh đã nhập và khoảng thời gian cơ sở đã chọn. Đảm bảo rằng bạn chỉ nhập các giá trị hợp lý và lưu ý rằng bất cứ điều gì bạn nhập vào đây đều có hậu quả về hiệu suất. Hậu quả không chỉ dừng lại ở mức tiêu thụ bộ

nhớ mà còn ở tốc độ. Xử lý nhiều thanh hơn có nghĩa là nhiều thời gian CPU hơn, đặc biệt là khi xem xét thực tế là trong khi lượng dữ liệu nhỏ có thể được lưu trong bộ nhớ đệm CPU trên chip, thì lượng dữ liệu lớn thì không. Bây giờ, xét đến việc bộ nhớ đệm trên chip thường nhanh hơn bộ nhớ thông thường 10 (mười) lần, bạn sẽ nhận ra ngay rằng việc chỉ định quá nhiều thanh ở đây sẽ làm giảm hiệu suất. Một lần nữa: một thanh là 40 byte. Để có hiệu suất tốt nhất, hãy đảm bảo rằng bạn không vượt quá kích thước bộ nhớ đệm trên chip

của CPU, vì vậy nếu CPU của bạn có bộ nhớ đệm 4MB, để có hiệu suất tốt nhất, bạn nên không sử dụng quá 100000 thanh trong Cài đặt tệp->Cơ sở dữ liệu.

• Giảm kích thước bộ nhớ đệm trong bộ nhớ, nếu bạn đang sử dụng cơ sở dữ liệu rất lớn (>2GB kích thước đĩa)

Kích thước "bộ đệm trong bộ nhớ" được xác định trong Tools->Preferences, tab "Data" kiểm soát số lượng ký hiệu trong bộ đệm và lượng bộ nhớ tối đa mà bộ đệm sử dụng. Mặc dù bộ đệm lớn hơn thường tăng tốc độ xử lý vì dữ liệu không cần phải được đọc từ đĩa và ghi lại mọi lúc, nhưng bạn có thể hết bộ nhớ trong một số trường hợp nhất định khi bộ đệm của bạn quá lớn, đặc biệt là trên hệ điều hành 32 bit giới hạn bộ nhớ ảo khả dụng cho mỗi quy trình ở mức 2GB. Để tránh hết bộ nhớ, hãy vào Tools->Preferences, "Data" và giảm kích thước bộ đệm trong bộ nhớ. Để đặt thành tối thiểu, hãy nhập "11" (mười một) vào "bộ đệm trong bộ nhớ (tối đa ký hiệu)".

QUAN TRỌNG: KHÔNG nên giảm bộ đệm khi cơ sở dữ liệu của bạn tương đối nhỏ (vài trăm MB) vì bộ đệm lớn hơn sẽ tăng tốc độ truy cập. Bộ đệm lớn là điều tốt miễn là bạn không hết bộ nhớ.

Về cấp độ mã hóa công thức

Mã hóa công thức kém là lý do chính khiến máy chậm lại. Những người đến từ "ngôn ngữ" khác thường không nhận ra hết tiềm năng của xử lý mảng AFL và mã hóa mọi thứ theo kiểu "cũ" (tức là với vòng lặp). Vòng lặp có thể chậm hơn 10..50 lần so với mã dựa trên mảng tương đương. Vì vậy, để có tốc độ tốt nhất, bạn nên tránh vòng lặp hoàn toàn và thay thế chúng bằng xử lý mảng hoặc ít nhất là làm cho mã lặp càng ngắn càng tốt.

Hãy xem xét mã sau:

```
SetBarsRequired( sbrAll ); GetPerformanceCounter(1);

for( i = 0; i<BarCount; i++)
```

```
{
med[ i ] = (H[ i ] + L[ i ])/2;
}

"Thời gian vòng lặp: "+GetPerformanceCounter(1); med = ( H + L )/2;
"Thời gian mảng: "+GetPerformanceCounter(1);
```

Khi chạy mã này trên 350000 thanh, phiên bản vòng lặp mất 100ms (0,1 giây) để thực thi phiên bản vòng lặp và

chỉ mất 2ms (0,002ms) để thực thi phiên bản mảng (do đó, mã mảng nhanh hơn 50 lần so với vòng lặp). Sự khác biệt không quá đáng kể với ít thanh hơn, nhưng ngay cả với 300 thanh, vòng lặp cần 0,1ms và mảng cần 0,01ms nên nhanh hơn 10 lần.

Vì vậy, có một số hướng dẫn cho mã hóa AFL:

• Sử dụng xử lý mảng thay vì vòng lặp bất cứ khi nào có thể. Để tìm hiểu thêm về xử lý mảng, vui lòng đọc hướng dẫn "Hiểu về AFL"

• Các vòng lặp bên trong chỉ sử dụng số vô hướng hoặc tham chiếu đến từng phần tử mảng bằng toán tử chỉ số []

• Các hoạt động mảng và tất cả mã bất biến vòng lặp nên được di chuyển ra ngoài vòng lặp bất cứ khi nào có thể để tránh việc đánh giá lại cùng một hàm nhiều lần

Hãy xem xét mã này:

```
for( i = 0; i < BarCount; i++ )
{
x = MA( C, 10 ); // SAI! mã bất biến vòng lặp được gọi đi gọi lại
! Nó phải nằm NGOÀI vòng lặp y[ i ] = C[ i ]/x[ i ];
}
```

Vấn đề với mã trên là nó lặp lại cùng một phép tính trung bình động dựa trên mảng với cùng một đầu vào nhiều lần (BarCount-times). Mã như thế này là bất biến vòng lặp (không phụ thuộc vào bộ đếm vòng lặp và không có bất kỳ đầu vào nào được tính toán tùy thuộc vào bộ đếm vòng lặp, do đó, nó có thể được tính MỘT LẦN và do đó nên được di chuyển ra ngoài (trước) vòng lặp, như được hiển thị bên dưới:

```
x = MA( C, 10 ); // ĐÚNG! mã bất biến vòng lặp nằm NGOÀI vòng lặp


for( i = 0; i < BarCount; i++ )

{

y[ i ] = C[ i ]/x[ i ];

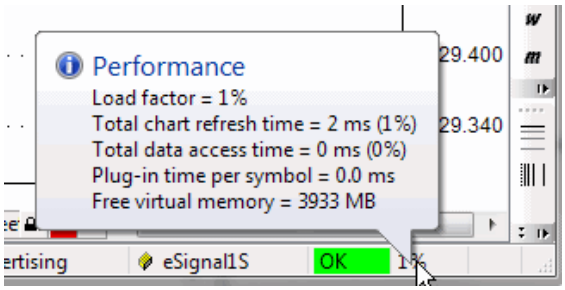
}
```

• Sử dụng Trình chỉnh sửa công thức AFL, Công cụ->Kiểm tra mã & Hồ sơ để tìm ra hàm nào được gọi bao nhiêu lần và hàm nào mất nhiều thời gian nhất. Bắt đầu điều chỉnh mã của bạn bằng các hàm được gọi thường xuyên nhất. Kiểm tra xem chúng có phải là bất biến vòng lặp không và nếu có, hãy di chuyển chúng RA NGOÀI vòng lặp.

• Nếu bạn cần lưu trữ dữ liệu mảng trong một công thức và đọc lại trong công thức khác và đã sử dụng AddToComposite/Foreign trước đó, hãy cân nhắc sử dụng biến tĩnh mảng (StaticVarGet/StaticVarSet) thay vào đó vì các biến tĩnh hoạt động nhanh hơn. Lưu ý rằng tuổi thọ của biến tĩnh bị giới hạn trong thời gian chạy chương trình (chúng không bền bỉ như các ticker AddToComposite), trừ khi bạn đặt tham số Persistent thành True trong StaticVarSet.

Giám sát hiệu suất

Để giúp bạn giám sát hiệu suất chương trình theo thời gian thực, AmiBroker cung cấp hai công cụ. Đầu tiên là cửa sổ giám sát hiệu suất, thứ hai là chỉ báo hiệu suất nằm ở phía bên phải nhất của thanh trạng thái AmiBroker.



Chỉ báo hiệu suất trên thanh trạng thái hiển thị:

• đối với cơ sở dữ liệu thời gian thực: chỉ báo hệ số tải phần trăm

• đối với cơ sở dữ liệu trong ngày và cuối ngày ngoại tuyến: bộ nhớ ảo trống

Hệ số tải là giá trị phần trăm cho biết 'độ nhanh' tương đối của chương trình. Hệ số tải được tính là (tổng thời gian làm mới biểu đồ tính bằng mili giây)/2 + (tổng thời gian truy cập dữ liệu tính bằng mili giây)/2 + (bộ nhớ ảo trống dưới 20% tổng bộ nhớ khả dụng). Vì vậy, nó sẽ đạt 100% nếu bất kỳ tình huống nào được liệt kê dưới đây xảy ra:

• tổng thời gian làm mới biểu đồ cao hơn 200ms

• tổng thời gian truy cập dữ liệu cao hơn 200ms

• bộ nhớ ảo trống giảm xuống dưới 10% tổng bộ nhớ (hoặc kết hợp các yếu tố trên)

Tổng thời gian làm mới biểu đồ là tổng thời gian cần thiết để vẽ lại hoàn toàn tất cả các biểu đồ hiển thị trên màn hình, bao gồm thời gian thực hiện AFL cho mỗi khung biểu đồ và đầu ra GDI (đồ họa) trên màn hình. (Không bao gồm truy cập dữ liệu)

Tổng thời gian truy cập dữ liệu là tổng thời gian cần thiết để truy cập dữ liệu mới thông qua plugin cho tất cả các ký hiệu được hiển thị cộng với thời gian cần thiết để áp dụng lọc thời gian và nén thời gian từ khoảng thời gian cơ sở sang khoảng thời gian được hiển thị.

Thời gian cắm thêm cho mỗi ký hiệu là thời gian dành cho lệnh gọi GetQuotes của plugin cho mỗi *một* ký hiệu. Nếu bạn hiển thị 10 biểu đồ ký hiệu cùng một lúc, AmiBroker sẽ gọi GetQuotes 10 lần, do đó thời gian này được nhân với số ký hiệu được hiển thị (tổng thời gian của plugin *được* bao gồm trong tổng thời gian truy cập dữ liệu - được liệt kê ở trên)

Nếu thời gian của Plug-in cho mỗi ký hiệu vượt quá 10ms, điều đó có nghĩa là plugin chậm (hoặc không sử dụng ADK 2.0 mới), nếu đúng như vậy, bạn nên liên hệ với nhà cung cấp plugin để cập nhật plugin sử dụng ADK 2.0.

Khuyến nghị là giữ hệ số tải này dưới 100%. Khi hệ số tải là 100%, AmiBroker có thể tiếp tục cập nhật tất cả các biểu đồ theo thời gian thực (thường xuyên hơn 5 lần mỗi giây) và duy trì giao diện người dùng phản hồi và mượt mà. Với hệ số tải là 200%, AMiBroker vẫn có thể tiếp tục cập nhật tất cả các biểu đồ với tần suất là 2,5 lần mỗi giây, nhưng thời gian phản ứng của giao diện người dùng có thể bị suy giảm một chút. Khuyến nghị giữ tải ở mức 100% hoặc thấp hơn.

200% là giá trị tối đa cho phép hoạt động "bình thường" hơn hoặc ít hơn.

Khi hệ số tải tăng trên 100%, chú giải công cụ cảnh báo sẽ bật lên một lần để thông báo lý do hiệu suất kém Khi hệ số tải tăng trên 300%, chú giải công cụ trên sẽ xuất hiện lại sau mỗi phút.

AmiBroker sẽ tiếp tục hoạt động với tải thậm chí trên 1000%, tuy nhiên hiệu suất sẽ kém (cập nhật một lần sau mỗi 5 giây hoặc hơn).

Hiệu suất đa luồng

Để biết thêm thông tin về cách sử dụng tối ưu đa luồng và các nhận xét chung khác về hiệu suất, hãy xem Sử dụng hiệu quả đa luồng

Làm thế nào để mua AmiBroker?

AmiBroker là phần mềm dùng thử. Điều này có nghĩa là BẠN NÊN đánh giá phiên bản dùng thử của chương trình trong thời gian 30 ngày trước khi mua.

Nếu bạn thích chương trình và muốn sử dụng trong thời gian đánh giá hơn 30 ngày - bạn phải mua giấy phép để sử dụng. Chúng tôi cho rằng bạn đã cài đặt AmiBroker trước khi đặt hàng và kiểm tra xem nó có phù hợp với nhu cầu của bạn không.

Phần mềm AmiBroker hiện có 2 phiên bản: Tiêu chuẩn và Chuyên nghiệp (RT). Để tìm hiểu về sự khác biệt giữa hai phiên bản này, hãy nhấp vào đây.

GIÁ CẢ

Phí cấp phép như sau đối với GIẤY PHÉP MỘT NGƯỜI DÙNG MỚI

Sản phẩm	Giá
AmiBroker Standard Edition	$299
AmiBroker Professional Edition	$399
AmiQuote (bổ sung tải xuống báo giá)	$99
AFL Code Wizard (bổ sung xây dựng công thức)	$99
AmiBroker Ultimate Pack (bao gồm AmiBroker Professional + AmiQuote + AFL Code Wizard)	$499

Phí cấp phép cho CÁC BẢN NÂNG CẤP (Có hiệu lực chỉ cho người dùng đã có giấy phép của các phiên bản trước sau 2 năm miễn phí bảo trì)

Sản phẩm	Giá
AmiBroker Standard Edition	$159
AmiBroker Professional Edition	$199

Tất cả các giao dịch có thể được thực hiện một cách an toàn qua cửa hàng trực tuyến của chúng tôi: https://www.amibroker.com/order.html

LỢI ÍCH KHI MUA AMIBROKER

Khi mua AmiBroker, bạn sẽ nhận được những lợi ích sau:

- **Giấy phép chính**: Kích hoạt tất cả các tính năng của chương trình (lưu cơ sở dữ liệu, không còn yêu cầu phiền phức).
- **Cập nhật miễn phí trong 2 năm**: Từ ngày mua, với ít nhất hai bản cập nhật lớn chính thức.
- **Truy cập khu vực chỉ dành cho thành viên**:
  - AmiBroker Developer Kit (dành cho các nhà phát triển plugin DLL).
  - Bản tin hàng tuần AmiBroker Tips mới nhất.
  - Các mẹo giao dịch hàng tháng từ Stocks & Commodities® cho AmiBroker.
  - Phiên bản mới nhất, riêng tư của AmiBroker.
  - Công thức AFL bổ sung cho các chỉ báo, bình luận, hệ thống giao dịch.
- **Ảnh hưởng đến tương lai của AmiBroker**: Đề xuất tính năng mới của bạn có khả năng cao hơn được thực hiện.
- **Giảm giá 50% cho năm tiếp theo của gói nâng cấp và bảo trì**.
- **Hỗ trợ kỹ thuật 24 tháng qua email**.
- **Các phần thưởng khác**.

GIAO HÀNG

Sau khi thanh toán phí đăng ký, bạn sẽ nhận được tệp khóa cá nhân qua email. Không hỗ trợ phương thức giao hàng khác. Khi mua, vui lòng cung cấp địa chỉ email của bạn.

CÁCH ĐẶT HÀNG AMIBROKER?

ĐẶT HÀNG TRỰC TUYẾN

Nếu bạn muốn mua AmiBroker, bạn có thể thực hiện giao dịch trực tuyến một cách an toàn qua các liên kết dưới đây. Phương thức thanh toán bao gồm tất cả các thẻ tín dụng lớn cũng như séc và chuyển khoản.

Để đặt hàng trực tuyến, vui lòng truy cập: https://www.amibroker.com/order.html

Dịch vụ đặt hàng trực tuyến của AmiBroker được cung cấp bởi FastSpring và ShareIt Digital River, các đại lý đăng ký shareware và xử lý thẻ tín dụng uy tín. Máy chủ của họ sử dụng mã hóa mạnh mẽ của trình duyệt bạn, cùng với xác thực VeriSign/Thawte, để mã hóa thông tin cá nhân và chi tiết thẻ tín dụng của bạn, đảm bảo không bị chặn bởi hacker hoặc bên thứ ba.

Tất cả dữ liệu thẻ tín dụng được truyền tải qua giao thức HTTP an toàn (mã hóa) theo tiêu chuẩn mã hóa SSL 128-bit hiện hành. Việc mua sắm trực tuyến an toàn đã trở thành mối quan tâm lớn. Năm nay, người tiêu dùng trực tuyến sẽ chi hơn 5.7 tỷ USD theo International Data Corp. Mối quan tâm chính của người tiêu dùng là thông tin thẻ tín dụng của họ có thể bị rò rỉ. FastSpring và ShareIt sử dụng công nghệ máy chủ an toàn, mã hóa thông tin đặt hàng của bạn, giữ cho nó riêng tư và bảo vệ. Công nghệ này được sử dụng bởi tất cả các trang mua sắm thương mại lớn.

Để biết thêm thông tin về các vấn đề an ninh, vui lòng tham khảo tài liệu của trình duyệt bạn. Ngoài ra, xin lưu ý rằng tất cả thông tin được gửi qua cửa hàng trực tuyến đều hoàn toàn bảo mật - chúng tôi sẽ không bán hoặc cung cấp địa chỉ email hay chi tiết của bạn cho bên thứ ba!

THÔNG TIN THÊM

Mua hàng trực tuyến là cách nhanh nhất để nhận mã đăng ký cá nhân của bạn. Sau khi hoàn tất đăng ký, bạn sẽ nhận được thông tin cá nhân trong vòng 24 giờ.

Lưu ý quan trọng: Vui lòng cung cấp địa chỉ email chính xác và đầy đủ. Nếu bạn nhập sai địa chỉ email (hoặc địa chỉ không hoạt động), bạn sẽ không thể đăng ký phần mềm của mình.

Các câu hỏi liên quan đến đăng ký có thể được giải đáp qua trang hỗ trợ: http://www.amibroker.com/support.htm

Cảm ơn

Nhiều người đã đóng góp quan trọng vào sự phát triển và thử nghiệm AmiBroker. Xin cảm ơn tất cả các bạn. Đặc biệt cảm ơn các cá nhân sau: Herman van den Bergen, Bill Barack, Bruce Robinson, Howard B. Bandy, Rick Perkins, Ken Close, Dimitris Tsokakis, Marek Ch³opek, Ed Winters, Patrick Hargus, Mark Leon, Dan Clark, Rick Parsons, Charlie Hooper, Steve Wiser, Jim Ellis, David Holzgrefe, Carlton McEachern, Geoff Mulhall, Richard Cloonan, Peter Gialames, Stephane Carrasset, Dale Wingo, Fred Tonetti, Chuck Rademacher, Gary A. Serkhoshian, Rick Perkins, Tom Supera, Michael Robb, Mark Allen, Geo Singleman, Anthony Faragasso, Jayson Casavant, Al Holzwarth, Sidney Kaiser, William Peters, Ara Kaloustian và tất cả những người dùng AmiBroker khác đã cung cấp phản hồi, ý kiến, ý tưởng, gợi ý, kết quả thử nghiệm và sự hỗ trợ quý báu.

Cảm ơn đặc biệt đến eSignal, myTrack, IQFeed vì sự hỗ trợ và hợp tác rộng rãi. Cảm ơn Jerry Medved (QuoteTracker) đã hợp tác. Cảm ơn Mark Jurik của Jurik Research đã cung cấp công cụ của mình. Cảm ơn Gary Lyben của Quotes Plus đã giúp đỡ trong việc kết nối với cơ sở dữ liệu Quotes Plus. Xin chân thành cảm ơn tất cả những người đóng góp cho thư viện công thức AFL vì đã chia sẻ công việc của họ. Cảm ơn Sharenet (Robin và Steve) và tất cả người dùng Nam Phi vì sự hỗ trợ liên tục. Cảm ơn Jordan Russell và Martijn Laan vì InnoSetup/ISX. Cảm ơn Neil Hodgson vì điều khiển Scintilla. Cảm ơn Donald Dalley vì sự hỗ trợ lớn mà ông đã cung cấp cho phiên bản Amiga của AmiBroker.

Cuối cùng, tôi xin gửi lời cảm ơn và yêu thương sâu sắc đến vợ tôi, Elizabeth, và những nụ hôn dành cho các con chúng tôi, Julia, Jacob và Nico, vì đã mang lại niềm vui cho cuộc sống của tôi mỗi ngày. Và cảm ơn bạn, người dùng thân mến, vì đã mua AmiBroker! Với sự hỗ trợ của bạn, chúng tôi có thể biến những giấc mơ thành hiện thực.

AmiBroker trên mạng

Để cập nhật tin tức, bản vá và thông tin mới nhất, vui lòng truy cập trang web của AmiBroker: http://www.amibroker.com.

Thăm phần hỗ trợ trên trang web AmiBroker tại:
http://www.amibroker.com/support.html
Cơ sở tri thức: http://www.amibroker.com/kb/
DevLog: http://www.amibroker.com/devlog/
Diễn đàn cộng đồng chính thức của AmiBroker: https://forum.amibroker.com
Thư viện AFL trực tuyến: http://www.amibroker.com/library/
Tài liệu tham khảo AFL trực tuyến: http://www.amibroker.com/guide/afl/
Khu vực bên thứ ba (plugin, tài liệu): http://www.amibroker.net/3rdparty.php