Highlights of version 5.70

- **Analysis improvements:**
  - **New Multi-threaded Individual Optimization**
  - **New general-purpose ranking functions**
  - User-definable ranking columns (via AddRankColumn function)
  - Lots of internal speedups in backtesting/optimization engine
  - SPSO, Tribes optimization engines now available also in 64-bit

- **Time&Sales improvements**: user-definable filtering, user-definable colors, 2 user-selectable display modes
- **Database improvements**:
  - 64-bit version supports files larger than 2GB per symbol
  - in-memory cache can hold upto 100 000 symbols (up from 20K)
  - new 64-bit DDE and ODBC plugins
- **Charting improvements:**
  - Greatly improved performance QuickData technology implemented lowering CPU usage for charts
  - Edit->'Paste Special' allows to copy-paste entire chart pane with various options
  - Distance measuring when drawing trendlines (X,Y distance in the status bar)
  - X/Y constrains for drawing tools (press X and/or Y key to constrain movement in either X or Y direction when drawing)
  - ASCII importer adds support for millisecond timestamps
- **AFL improvements**:
  - new functions:
    - StaticVarGenerateRanks, StaticVarGetRankedSymbols - general-purpose user-definable ranking
    - Error - display user-definable error messages (also useful for plugin developers)
    - fdir - directory listing
    - CategoryCreate - programmatic creation of watch-lists
    - AddRankColumn - ranking columns in exploration
  - performance improved for Percentile() (order(s) of magnitude)
  - new fields supported in GetFnData
  - XShift support added to PlotShapes
  - speeded up transcendental mathfunctions (sqrt, sin,asin, cos, acos, tan, atan, ln, log10, etc)
  - improved SetSortColumns

CHANGES FOR VERSION 5.70.2 (as compared to 5.70.0), Official Release

- Vertical selector line for the very first quote in a chart was not displayed. Fixed.
- Exception could occur when passing openpos object to user-defined function. Fixed.
- Deleted databases are removed on start up from that "Recent databases" menu
- An exception at 678BAB24 when full name edit was incomplete and symbol was changed. Fixed.
- In 32-bit version of AmiBroker, StaticVarInfo could return negative value when static memory usage was greater than 2GB. Fixed
- Added extra checks in SetOption to avoid crash when user deletes symbols while formula is running (which is bad idea anyway)
- On attempt to print/copy image of an empty chart pane appropriate message is displayed
- State of command bars is loaded with 'silent' option to prevent "due to software update the toolbar has changed " messages that could appear when user launched AmiBroker from different folders or different versions (old and new)

CHANGES FOR VERSION 5.70.0 (as compared to 5.69.0)

- GDI: use of PS_ALTERNATE style in Windows 95/98 causes resource leak. Fixed.
- UI: Place Order dialog, it was problematic to enter fractional offsets for brackets. Fixed.
- " Recent database" menu randomly listed files instead of databases. Fixed. Note if you have custom toolbar with "Recent database" submenu added, you need to delete it and re-insert.

CHANGES FOR VERSION 5.69.0 (as compared to 5.68.1)

- Adding symbols via ASCII import caused change of selected symbols in charts. Fixed.
- OLE: AnalysisDoc.Run supports Individual Optimization now (Action == 5)
- Charts: when formula file can not be found the detailed information about the path referenced and current working directory is displayed instead of old "the formula can not be found - no chart to display"
- New Analysis: added Auto-repeat status icon in the right-hand corner of the Info bar (yellow 'play' arrow shows up when auto-repeat is active and scheduled for next run)
- AFL: Percentile() uses now new algorithm that has linear O(n) complexity instead of O(log(n)*n) so it is MUCH faster than the old one and close to Median() that uses quickselect (which is fastest)

  By the way, to find average of upper and lower median for EVEN number of elements you can use
  Percentile(array, range, 50)

  - it will execute approx 30% slower than Median, but Median only returns LOWER median for even 'range' parameter.

- ODBC plugin: now available in 64-bit
- AFL: since 5.63 round() function used 32-bit floorf which resulted in some differnces as compared to 5.60. Previous code version is restored now.
- DDE plugin: 64-bit version of DDE plugin was producing incorrect timestamps. Fixed.
- Default seed for Random() and mtRandom() was based on time(), which could be the same for threads created in the very same moment. Now the seed in multiple threads is guaranteed to be unique. This prevents repeated pseudo-random sequences in multiple threads created in the very same moment.

- IB plugin: added support for CFDs (security type=CFD, exchange=SMART) and commodities (security type = CMDTY, exchange = SMART)

  Example symbology for CFD:
  IBUS500-SMART-CFD-USD

  Symbol-SMART-CFD-Currency

  Example symbology for commodities:
  XAUUSD-SMART-CMDTY

  Symbol-SMART-CMDTY-Currency

- UI: Now there are separate MRU (most recently used) menus for files and databases (File->Recent Databases, File->Recent Files_)
- Removed / changed some mnemonics (key accelerators) in menus to avoid duplicates
- The default setting for Layers/"Auto-Select Layer" is changed to OFF.
- Custom Backtester interface: added FullName read-only property to Trade object (gives full name of symbol)

  You can use it for example to add full name as a custom metric to the trade list

  ```
  SetCustomBacktestProc("");

  /* Now custom-backtest procedure follows */
  if( Status("action") == actionPortfolio )
  {
  bo = GetBacktesterObject();

  bo.Backtest(1); // run default backtest procedure

  // iterate through closed trades first
  for( trade = bo.GetFirstTrade(); trade; trade = bo.GetNextTrade() )
  {
  trade.AddCustomMetric("FullName", trade.FullName );
  }

  bo.ListTrades();

  }

  // your trading system here


  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));
  ```

- SPSO (standard particle swarm optimizer) plugin: now available in 64-bit
- Tribes plugin: now available in 64-bit
- DDE plugin: now available in 64-bit

CHANGES FOR VERSION 5.68.1 (as compared to 5.68.0)

- fixed crash in 64-bit version when usign StaticVarGenerateRanks on large arrays
- AFL: since 5.63 round() function used 32-bit floorf which resulted in some differnces as compared to 5.60. Previous code version is restored now.

CHANGES FOR VERSION 5.68.0 (as compared to 5.67.2)

- AFL: new function: Error("text") - allows to display user-defined error messages and stop execution of the formula. Good for 3rd party plugin implementors to display error messages

  Example:
  Error("Something went wrong");

  Note that error number in case of user-defined error message is always 99.

- AFL: new function: StaticVarGetRankedSymbols("outputprefix", "inputprefix", datetime)
- AFL: new function: StaticVarGenerateRanks function (flexible ranking function)

  NOTE: This function is NOT intended to replace bakctester's built-in ranking via PositionScore. Just the opposite: whenever you can, you should use PositionScore as it is way way faster and less memory-consuming way to perform backtests with ranking. To learn more about how to use PositionScore see "Portfolio level backtesting" section of the tutorial. StaticVarGenerateRanks is intended to be used for tasks OTHER than backtesting such as explorations or indicators that may require ranking functionality.

  WARNING: this function is computationally and memory intensive. It takes about 20ms per 15K bars and 7 symbols. Try to call it JUST ONCE per scan/exploration/backtest using if(Status("stocknum")==0) or better yet, use separate scan just once to pre-calculate ranks and use it later (like composite creation scan). If you fail to do so and call StaticVarGenerateRanks for every symbol performance would drop significantly as this function not only needs lots of time to compute but it also has to lock the access to shared memory used by static variables so other threads trying to access static variables would wait until this function completes.

  StaticVarGenarateRanks("outputprefix", "inputprefix", topranks, tiemode)

  "inputprefix" is a prefix that defines names of static variables that will be used as input for ranking.
  AmiBroker will search for all static variables that begin with that prefix and assume that remaining part of the variable name is a stock symbol. Say you want to rank stocks by ROC (rate of change). All you need to do is to store values into static variables.
  Let us say that we will use static variable names like "ValuesToSortAPPL", "ValuesToSortMSFT", and so on.

  To fill input static variables you can use this loop:

  ```
  for( i = 0; ( sym = StrExtract( symlist, i ) ) != ""; i++ )
  {
  SetForeign(sym );
  Value = ROC( C, 10 );
  RestorePriceArrays();
  StaticVarSet( "ValuesToSort" + sym, Value );
  }
  ```

Now you are ready to perform sorting/ranking. There are two modes, normal ranking mode and Top/Bottom Rank mode.
Normal ranking mode is performed when toprank argument is set to zero.

```
StaticVarGenerateRanks( "rank", "ValuesToSort", 0, 1224 );
```

In this case StaticVarGenerateRanks call would generate set of static variables starting with prefix defined by 2nd argument
each variable holding the rank of particular symbol, so in this case
RankValuesToSortMSFT will hold ranking of MSFT
RankValuesToSortAAPL will hold ranking of AAPL
Note that in AmiBroker rank count start from ZERO.

Third argument (topranks) is zero in normal ranking mode
Fourth argument (tiemode) defines how ties are ranked. Supported modes are 1234 and 1224. In 1224 mode ties are numbered with equal rank.

Example code for normal ranking mode (everything done is done in one pass, can be used in indicator):

```
symlist = "C,CAT,DD,GE,IBM,INTC,MSFT";

// delete static variables
StaticVarRemove("ValuesToSort*");

// fill input static arrays
for( i = 0; ( sym = StrExtract( symlist, i ) ) != ""; i++ )
{
SetForeign(sym );
Value = ROC( C, 10 );
RestorePriceArrays();
StaticVarSet( "ValuesToSort" + sym, Value );
}

// perform ranking
StaticVarGenerateRanks( "rank", "ValuesToSort", 0, 1224 ); // normal rank mode

// read ranking
for( i = 0; ( sym = StrExtract( symlist, i ) ) != ""; i++ )
{
Plot( StaticVarGet( "RankValuesToSort" + sym ), sym, colorCustom10 + i );
}
```

Top/bottom ranking mode (that generates top/bottom ranking tables that hold indexes to top ranking values. When topranks > 0 top ranked values are used, when topranks < 0 then bottom ranked values are used. The values are stored in variables that have format of: OutputprefixInputprefixN where N is a number 1, 2, 3 representing top/bottom ranks. Let us assume that OutputPrefix parameter is "Top" and Inputprefix parameter is ROC. In such case variable TopROC1 would hold the index of top rated value. TopROC2 would hold second top rated value, and so on.
StaticVarGenerateRanks function uses rank numbering that starts from ONE. In top ranking mode StaticVarGenerateRanks will also prepare static variable that contains comma separated list of variable names that can be used to find out which index refers to which symbol. So if TopROC1 holds 1 you would lookup first substring in TopROCSymbols variable to find out what variable (symbol) ranked at the top. Additionally StaticVarGetRankedSymbols gives easy-to-use method to retrieve comma separated list of ranked symbols for particular datetime.

Example code for top ranking mode

```
symlist = "C,CAT,DD,GE,IBM,INTC,MSFT";

// delete static variables
StaticVarRemove("ValuesToSort*");

// fill input static arrays
for( i = 0; ( sym = StrExtract( symlist, i ) ) != ""; i++ )
{
SetForeign(sym );
Value = ROC( C, 10 );
RestorePriceArrays();
StaticVarSet( "ValuesToSort" + sym, Value );
}

// perform ranking
StaticVarGenerateRanks( "rank", "ValuesToSort", 0, 1224 ); // normal rank mode
StaticVarGenerateRanks( "top", "ValuesToSort", 3, 1224 ); // top-N mode
StaticVarGenerateRanks( "bot", "ValuesToSort", -3, 1224 ); // bottom-N mode
// read ranking
for( i = 0; ( sym = StrExtract( symlist, i ) ) != ""; i++ )
{
Plot( StaticVarGet( "RankValuesToSort" + sym ), sym, colorCustom10 + i );
}

sdt = SelectedValue( DateTime() );
Title = "{{NAME}} -{{DATE}} - {{VALUES}} TOP: " + StaticVarGetRankedSymbols( "top",
"ValuesToSort", sdt ) +
" BOT: " + StaticVarGetRankedSymbols( "bot", "ValuesToSort", sdt ) ;
```

- New Analysis: when two Analysis windows were open and running Scan simultaneously with AddToComposite composites could get reset by scan running in second window. Fixed.
- Added millisecond support to ASCII importer. Supported time format with milliseconds is HH:MM:SS.mmm (where mmm is milliseconds 000..999)
- Unsettled sale proceeds were settled at the end of the day instead of the beginning of the next day. Fixed.
- The message 'trade not entered because of insufficient funds' is modified to mention that it also it may not be entered because of incorrect position size specified by the user

CHANGES FOR VERSION 5.67.2 (as compared to 5.67.0)

- Using more than 100 Plot() statements in one formula could cause memory corruption sometimes resulting in access violation. Fixed.
- File->New->Blank chart and then clicking "View" caused exception in 5.67.1. Fixed
- Low-level gfx playback engine resets colors to black pen / white background, so it does not get influenced by chart colors like labels.
- GfxSetOverlayMode(2) sometimes produced incorrect output in 5.67.0. Fixed.

CHANGES FOR VERSION 5.67.0 (as compared to 5.66.0)

- New Analysis: Optimization continued even if there was error inside custom backtest procedure. Fixed.
- Charting: a new "Warning 902 can not plot a chart in logarithmic scale as a data set contains negative value(s)" message is displayed on attempt to use log scale with negative data and the chart is plotted in linear scale instead.
- UI: 'Logarithmic scale' item added to "View" menu.
  Also the user can add this to View toolbar using toolbar customization (click on arrow in the "View" toolbar, "Add or remove buttons" then "View" then "Logarithmic...")

- AFL: new function fdir("wildcard", flags = 1) - returns comma separated directory list (like "DIR" command)
  "wildcard" is a path with a wildcard pattern to match.
  For example "C:\\*.*" will give you all files in C: drive, but "C:\\*.txt" will give you only files with .txt extension.
  flags - controls what is returned the default is 1 - only files, 2 - only directories, 3 - both files and directories

  Example:

  ```
  printf("Only files:\n");
  _N( list = fdir( "c:\\*.*", 1 ) );

  for( i = 0; ( filename = StrExtract( List, i ) ) != ""; i++ )
  {
  printf( filename + "\r\n" );
  }

  printf("\n\nOnly directories:\n");
  _N( list = fdir( "c:\\*.*", 2 ) );

  for( i = 0; ( filename = StrExtract( List, i ) ) != ""; i++ )
  {
  printf( filename + "\r\n" );
  }

  printf("\n\nBoth files and directories:\n");
  _N( list = fdir( "c:\\*.*", 3 ) );

  for( i = 0; ( filename = StrExtract( List, i ) ) != ""; i++ )
  {
  printf( filename + "\r\n" );
  }
  ```

- GfxSetOverlayMode(mode = 1) did not work with "alternate background fill". Fixed.
- Analysis Settings Load/Save buttons were positioned wrong in RTL (right-to-left) systems. Fixed.
- AFL: CategoryCreate("name", category) - adds new category (currently supports only adding watchlists)
  newwl = CategoryCreate("name", categoryWatchList) - this will add (create new) watch list with "name" (if it does not exist) or will return existing "name" watch list

  newwl will hold new watch list index.

  Currently only creation of watch list is supported by this function.

- 64-bit: Fixed crash with backtest of more than 10 million bars
- 64-bit: Fixed 'squeeze' problem with charts at full zoom out when more than 10 million data bars were displayed
- OLE Stocks.Add and Stocks.Remove trigger resynchronization of selected indexes to avoid problems with "symbol locked" charts

CHANGES FOR VERSION 5.66.0 (as compared to 5.65.0)

- 32-bit and 64-bit versions now use different registry key to save list of DLLs so warning messages about non-certified 3rd party plugins are not displayed each time you switch between 32-bit and 64-bit app version anymore.
- Foreign("~~~EQUITY") within custom backtest proc in old AA returned zero (because of change in 5.65). Fixed.
- CBT: when low-level mode was used (no ProcessTradeSignals was called) ~~~Equity open interest field was not filled with number of open positions. Fixed.

- When user moved default Formula path somewhere else Report charts were not generated properly. Fixed.
- CBT: a new field "TransactionCosts" (equivalent to report's "total transaction costs") is now supported in the custom backtester Stats.GetValue method
- ASCII importer: added DELISTING_DATE to fields available in $FORMAT command.
  To import delisted symbols you may use file like this. Acceptable date formats are
  YYYY-MM-DD (ISO standard) and DD-MM-YYYY (automatically detected).
  Note that MM-DD-YYYY (US) is not detected because one can not automatically detect if 01-02-2003 means
  february 1st or january 2nd

  ```
  #
  $FORMAT NAME, DELISTING_DATE
  $NOQUOTES 1
  $OVERWRITE 1
  SYMBOL,"2012-07-0
  ```

- AFL: GetFnData returned invalid value for fields DividendPayDate and ExDividendDate when they were empty (not set) in the Information window. Fixed (now Null is returned)
- AFL: GetFnData now supports new field "DelistingDate"

  ```
  dd = GetFnData("DelistingDate");
  if( IsNull( dd ) )
  printf("Security is active");
  else
  printf( "Delisted at" + dd + DateTimeToStr( dd ) );
  ```
- Information window: added "Delisting date" field.
  Delisting Date is a last day when security was available for trading due to delisting. If the security is still active the field is empty.
- Information window: it was not possible to set/change Dividend Pay Date and Ex Dividend date from UI. Fixed.
- Preferences window: new tab "Candles & Bars" that controls detailed appearance of candlestick and traditional bar charts

  Use up/down colors - when marked bar charts use candlestick up/down color defined in the Colors tab
  End cap style - controls whenever bar charts are plotted so the endings of lines are rounded, square or flat.
  The difference between square and flat is that square end will plot with "square" pen and with wide pens it will plot width/2 pixels above high and below low because of pen thickness (looks good but high/lows are not precisely marked).
  " Flat" means that ends terminate at exact position (so high and lows are precise), but the open/close ticks may look weird when plotted at high/low ("staircase effect")

  Open/Close tick % thickness - controls the thickness of line used to plot open/close ticks. It is expressed in % of main bar thickness (default is 100%)
  Open/Close tick % length - controls the length of open/close ticks. It is expressed in % of candle width which is usually half the distance between bars (but not more than 30 pixels).

- The database save was triggered when opening documents using recent file list. Fixed.
- Parameter window: Right mouse button double click over parameter label resets selected parameter to the default value now
- AFL: PlotShapes() now has XShift parameter
- Custom metric optimization target (WF tab) is now matched case-insensitive (FC#2442)

CHANGES FOR VERSION 5.65.0 (as compared to 5.64.0)

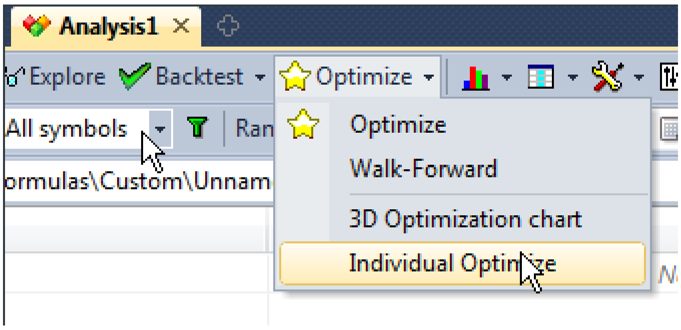
- In-memory quotation cache can now be larger than 20K symbols. Now the maximum is 100K symbols (user definable in Tools->Preferences, "Data" tab). Caveat use extreme values ONLY if you have plenty of memory AND you are using 64-bit version
- New Analysis: Walk-forward result list colums had "alpha" type which resulted in alphabetic sort when user clicked on column. Fixed (numeric type is used now).
- QuickData did not work properly for not time-based intervals so it is now automatically turned off for charts using such intervals.
- Fixed crash @0048C19A/48C1C4 (click on result list after backtest to show arrows sometimes produced this).
- Trade profit calculation speeded up which results in upto 15% faster optimization with lots of trades
- New Analysis: In 5.64.x "stocknum" incorrectly started counting from 1 instead of 0 as in old versions. Fixed.
- Foreign/SetForeign/RelStrength of the same symbol as active does nothing (i.e. refers to already existing thread local data, instead of querying database)
- New Analysis: when SetSortColumns was used in 5.63 and above it could trigger redundant multiple sorts. Now it is fixed (sort only once)
- New Analysis: results are now sorted after Individual Optimization the same way as after portfolio Optimization

CHANGES FOR VERSION 5.64.0 (as compared to 5.63.0)

- **New Analysis: Multi-threaded Individual Optimization** (experimental)**.** Upto NumberOfCores faster. Note: no CBT and only exhaustive mode is supported as of now

  The new optimizer is called "Individual Optimize" in the New analysis window and you can access it here:

  Note that old "Optimize" button runs OLD (i.e. 5.60) optimizer - for comparison purposes.

  "Individual Optimize" will use all available processor cores to perform single-symbol optimization.
  In "Current symbol" mode it will perform optimization on one symbol.
  In "All symbols" and "Filter" modes it will process all symbols sequentially, i.e.
  first complete optimization for first symbol, then optimization on second symbol, etc.

  Limitations:
  1. Custom backtester is not supported (yet)
  2. Smart optimization engines are NOT supported - only EXHAUSTIVE optimization works.

  For explanation of these limitations see http://www.amibroker.com/guide/h_multithreading.html

  Eventually I may get rid of limitation (1) - when I change the CBT NOT to use OLE anymore.
  But (2) is probably here to stay for long.

- Built-in constant are set up once and shared among all threads to save setup time (gives 10% speed up in execution of simple formulas)
- Built-in constants such as colorBlack are now read-only and any attempt to overwrite such constant value results in error 55.
- New Analysis, list view in scans, explorations and individual optimizations is refreshed less often to gain some speed.
- In case of individual backtest / individual optimization the backtester allocates much smaller price cache (just for 2 symbols) - takes less time.
- In 5.63 Bar Replay did not with "QuickData" enabled (it required View->Refresh All to start working). Fixed.
- Some actions (like import of past data, split, etc) did not trigger automatic chart refresh when QuickData was enabled. Fixed.

CHANGES FOR VERSION 5.63.0 (as compared to 5.62.0)

- QuickData for charts implemented. QuickData is a new feature that greatly decreases time required to prepare compressed/filtered data for real-time charts.
  QuickData is turned ON by default. It can be turned off in Tools->Preferences, "AFL" tab. Do NOT turn it off however, unless you are having problems/issues.
  QuickData minimizes number of required operations by analysing what has changed in the input data and re-doing compression only for changed portion. This results in significant real-time performance improvement especially on 1-minute databases having >200K bars and many charts open. With this option turned on and 6 charts open load factor can drop from above 100% to less than 1%. CPU usage drops by more than 7x (from 7% to less than 1%)
- Time&Sales now supports filtering out trades and bid/ask quotes below user-defined size
- Time&Sales allows to customize colors (separate color setting for trades occuring above/at ask, inside, at / below bid, etc). Right-click and use context menu to bring up "Colors..." dialog
- Time&Sales now supports "Wide list" mode in addition to existing "Narrow" mode. "Wide mode" has separate columns for bid / ask data. To switch - right click on the T&S window and choose "Wide" or "Narrow" from context menu
- OpenInterest field of ~~~EQUITY symbol now holds number of open positions on bar per bar basis
- GetFnData() supports now "Currency" and "WebID" fields
- AFL: some transcendental math functions (sqrt, sin,asin, cos, acos, tan, atan, ln, log10, etc) are 10-30% faster now.
- AmiQuote 3.05: fix to "It is illegal to call out while inside message filter" error that sometimes occurred when OLE server was busy during long imports
- ASCII importer: if import takes more than 5 seconds, Windows does not mark AmiBroker as "not responding" anymore
- Progress indicator (for ASCII import) in the status bar is larger and works better for large files
- AFL: SetSortColumns() can now be called more than once to support multiple rankings via AddRankColumn
- 64-bit version: Added 'highDPIaware' and "asInvoker' settings to application manifest in x64 version, to prevent file system virtualization and scaling on high-DPI screens
- Added X/Y constaints for drawing tools. When you hold down "Y" key, start and end Y co-ordinates are set to equal value (with trend line it gives horizontal line). When you hold down "X" key you get perfectly vertical line (X start/end set to equal value)
- When loading and saving quotation files, AmiBroker does not allocate extra memory buffer. This conserves RAM when handling large files.
- 64-bit version is able to load/save files larger than 2GB per symbol (more than 53 million data bars per symbol). Note that you would need to use huge amounts of RAM as using 50 million data bars requires 6GB of RAM or more for any decent performance.
  WARNING: Files larger than 2GB can not be used with 32-bit version. You must use 64-bit version in order to be able to access files >2GB.
  Also note that loading / saving file that has 2GB (2000MB) size on hard disk that has 100MB/second read/write speed takes over 20 seconds. Do not assume that the program hanged up only because Windows marks it as "not responding" when it is busy writing 2GB+ files. Windows would say the application is 'not responding' if it is busy reading/writing for more than 1 second.
  To work with such huge files you should invest in some very fast SSD to cut down those times, with currently fastest SSD use SATA3 6GBps and this would give you theoretical

maximum of 750MB per second. Still loading/saving 2GB from such high-performance drive would take 3+ seconds.

- When user re-arranged tabs in the "Log" window, "Clear" and other right-menu commands used wrong tab. Fixed.
- AFL: new function AddRankColumn - adds ranking column(s) according to current sort set by SetSortColumns

  ```
  Filter = 1;
  AddColumn( Close, "Close" );
  AddColumn( Volume, "BI" );
  AddSummaryRows( 31+32, 1.5 );

  AddRankColumn(); // without prior sorting AddRankColumn just adds line number
  SetSortColumns(-4);
  AddRankColumn(); // rank according to 4th column (descending)
  SetSortColumns(-3);
  AddRankColumn(); // rank according to 3rd column (ascending)
  ```

CHANGES FOR VERSION 5.62.0 (as compared to 5.61.0)

- AddSummaryRows now supports also flag = 32. This flag adds standard deviation row

  ```
  Filter=1;
  AddColumn(V, "Volume" );
  AddSummaryRows( 63, 1.2 );
  // add Total, Average, Min, Max, and Count and StdDev rows (1+2+4+8+16+32)=63 - with two decimal places summary rows are added at the top of the list
  ```
- AddSummaryRows treated NULLs as zeros in averages. Now it is fixed and NULLs are not included in calculations.
- Drawing tooltip "bars" count starts from 0 (when line is pure vertical) to match status bar X distance display. Status bar coords display is hidden once drawing is done to prevent user confusion.
- Snap to price now allows snapping to Closing and Opening price, to activate snap to close hold down "C" key, to activate snap to open hold down "O" key while drawing line / moving the mouse
- When "Play" button is pressed Bar replay dialog checks if Start Date is earlier than End date and displays error message if that is not the case.
- Ticker box is updated with new full name if it is changed from symbol information
- Per-symbol UserData fields are accessible now via GetFnData("UserData0")..GetFnData("UserData99"). This feature is for implementors of custom data plugins to allow them to expose custom data

  ```
  for( i = 0; i < 100; i++ )
  printf( "UserData%g %g\n", i, GetFnData("UserData"+i) );
  ```
- New Analysis: In 5.61.0 Walk Forward kept old "Current symbol" until Scan/Explore/Backtest was run. Fixed
- First line in "Parallel trendlines" tool is snapped to price (if the option is turned on) as normal trendline.
- eSignal: now plugin supports EOD history for more than 40 years back
- Charting: Copy-Paste Special allows to copy entire chart pane with various options

  a) Entire chart pane (fully independent) - it creates a copy of chart pane, assigns new chart ID, creates duplicate of the formula file so parameters are independent and formula is private (not shared with source pane)

  b) Entire chart pane (independent parameters, but shared formula), the same as (a) but does NOT create duplicate of the formula so it shares the very same formula file with the original pane

  c) Entire chart pane (hard-wired, shared parameters, formula and drawings, same chartID) - a new pasted pane is directly hard-wired with pane being copied, so it shares the same CHART ID and every change made to it is made to the original (source) too.

- Backtest report now includes 'Total transaction costs' (sum of all commissions paid)
- Aux1/Aux2 fields now allow user-definable aggregation/compression mode (File->Database Settings->Intraday Settings). Available choices are last (default), first, highest, lowest, sum

CHANGES FOR VERSION 5.61.0 (as compared to 5.60.3)

- When using any drawing tool X and Y distance is shown in the status bar (FC #2380)
- When position can not be entered due to insufficient funds the Backtester in the "detailed mode" gives additional info such as: requested entry price, requested position size (and dollar value) and requested round lots (FC#2377)
- Time&Sales column layout persists between runs now (FC #2341)
- Running any #import command (including "Update US symbol list and categories") is preceeded by prompt for confirmation now
- Report Explorer now supports copying selected items to clipboard (Edit->Copy), Ctrl+C (FC#2288)
- Quote Editor: added "Go to selected" button that quickly scrolls to and selects the data bar marked on chart (FC#2342)
- New Analysis: parameters were not reset to default values when formula was loaded. Fixed. (#2353)
- New Analysis: "Add artificial future bar" option does not affect Exploration and Scan anymore (FC#2353)
- Max. chart rendering time of 1000 ms when multithreaded charts are turned ON can be overriden now by registry setting. See FC#2330 for details.
- Increased width of symbol combo boxes in Symbol->Merge dialog to prevent truncation of ticker names
- Floating window caption is now synchronized even if it is not active (FC#2376)
- eSignal: new version allows to select whenever EOD bars should report Settlement price or regular close
- Change PlotShapes does NOT use "shape0", "shape1" variables anymore to prevent clashing with user-defined variables (#104931)
- ASCII importer now supports $WEBID command and WEBID field in the $FORMAT command to allow importing web ID (FC#2362)
- ASCII importer now supports $ISINDEX, $ISFAVORITE commands, and ISINDEX and ISFAVORITE fields in the $FORMAT command (FC#2310)
- Apply To: Filter dialog takes less time to show up