BOSSA.PL
DOM MAKLERSKI BOŚ S.A.

Plugin bossaAB

Podręcznik Użytkownika

Wersja 3.3.2

Copyright® Remigiusz Gajda

Data aktualizacji: **26.09.2017**

Dokumentacja Pluginu bossaAB do programu Amibroker©.

Spis treści: **str.**

1. Wprowadzenie

Plugin rozszerza funkcjonalność Amibrokera pozwalając na wysyłanie zleceń bezpośrednio za pomocą komend AFL.
Plugin korzysta z API NOL3 dla Domu Maklerskiego BOŚ.
Korzystać z niego mogą klienci DM BOŚ, którzy akceptują postanowienia licencji na plugin.

2. Wymagania

System operacyjny Windows 7 SP1/8.1/10.
Do zainstalowania plugina mogą być wymagane uprawnienia administratora.
Amibroker© wersja min. 5.30.
Karta sieciowa z uruchomioną usługą TCP/IP.

3. Instalacja

Plugin „**bossaAB.dll**" należy skopiować do podkatalogu „**Plugins**" katalogu głównego programu Amibroker.

Do katalogu głównego Amibrokera należy skopiować pliki „**nolNadzorca.exe**" oraz **„ListaDerywatów.txt"**
Po skopiowaniu plików i uruchomieniu Amibrokera pojawi się komunikat o niecertyfikowanym pluginie. Należy pozwolić na używanie plugina.

4. Właściwości

Plugin w założeniu ma za zadanie maksymalnie ułatwić składanie zleceń z poziomu skryptów Amibrokera. Cechują go między innymi:

- automatyczne pobieranie nazwy waloru z wykresu Amibrokera. Zamiast automatycznego wyboru dostępne jest wprowadzanie nazwy przez użytkownika.
- automatyczne wybieranie rachunku (akcyjny lub pochodne) dla danego waloru (na podstawie nazwy waloru). Zamiast automatycznego wyboru dostępne jest wprowadzanie numeru rachunku przez użytkownika.
- zapisywanie treści złożonych zleceń w pliku **„orders_list.txt"** i wczytywanie ich po ponownym uruchomieniu Amibrokera. Zapisy w pliku nie przenoszą się na następną sesję.
- blokowanie wysyłania zleceń dla danego waloru, które mają sygnaturę czasową starszą od ostatniej transakcji (sygnatura czasowa pobierana z jest z wykresu Amibrokera) – dotyczy interwałów Intraday i Ticks.
- wyłączanie składania zleceń jeżeli aplikacja interfejsu (wizualna część plugina) utraci łączność z funkcjami plugina.
- możliwość włączenia dodatkowych zabezpieczeń tradingu: kontrola interwału, kontrola identyfikatora wykresu dla danego waloru, dopuszczenie tylko operacji naprzemiennych (po kupnie sprzedaż i odwrotnie).
- kontrola ilości wykonanych zleceń/Anula/modyfikacji (ograniczenie do 2000 na sesję).
- kontrola częstości wysyłania zleceń (8 na 10 sekund, może być ustawiane).
- definiowalne alerty dźwiękowe dla większości zdarzeń na zleceniach.
- Sygnalizacja stanu połączenia z NOL, oraz stanu NOL (On/Off Line)

5. Ograniczenia

- Plugin nie wspiera trybu multithread Amibrokera (wersje multithread).
- Plugin nie odtwarza stanu zleceń z poprzedniej sesji
- Plugin resetuje niektóre zmienne wewnetrzne przy zmienie daty w czasie działania plugina: czyści stan zleceń z poprzedniego dnia, resetuje także znaczniki czasu dla każdego z walorów

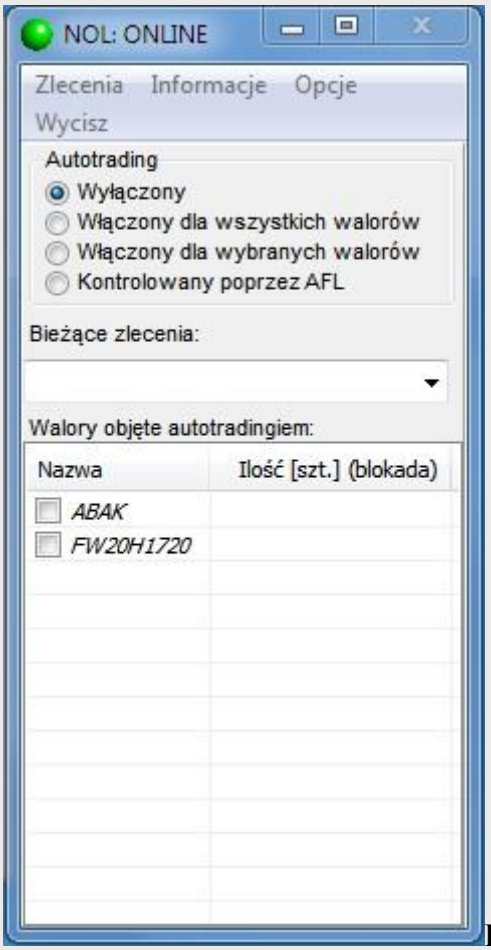
Plugin dopuszcza maksimum 300 różnych walorów na danym rachunku. Jeśli poprzez plugin zostanie złożone zlecenie na walor spoza tego zakresu, zostanie zwrócona informacja (okno logów) **„Walidacja. Nie można znaleźć security dla:** [nazwa waloru]**" –** samo zlecenie zostanie natomiast złożone! Dla walorów spoza zapamiętanej listy nie działają procedury zabezpieczające.

W trybie testowym niektóre z zabezpieczeń nie działają, są to: zabezpieczenie przed piramidowaniem pozycji (można złożyć kilka zleceń w tym samym kierunku np. kupna), zabezpieczenie czasu i daty złożenia zlecenia (można złożyć zlecenia z sygnaturą czasową wcześniejszą od ostatniego zlecenia). Wynika to stąd, że w trybie testowym nie są składane żadne realne zlecenia i odpowiednie parametry zleceń nie są aktualizowane.

6. Interfejs plugina

Okienko główne

Po uruchomieniu Amibrokera, jeśli na wykresie dowolnego waloru zostanie wykonany skrypt inicjalizujący omawiany plugin, w lewym górnym rogu ekranu pojawi się następujące okienko interfejsu (**rys. 1**):



Rys. 1

Powyższe okienko wyświetlane jest zawsze na wierzchu ekranu.

- Wybranie odpowiedniego pola **„Autotrading”** włącza/wyłącza funkcje automatycznego wysyłania zleceń:
  - **Wyłączony** – wyłącza autotrading dla wszystkich walorów
  - **Włączony dla wszystkich walorów** – włącza autotrading na wszystkich walorach w Amibrokerze, dla których zainicjalizowano plugin
  - **Włączony dla wybranych walorów** – włącza autotrading dla walorów zaznaczonych na liście poniżej (o ile w Amibrokerze został dla nich zainicjalizowany plugin)
  - **Kontrola poprzez AFL** – pozwala na włączanie/wyłączanie autotradingu dla każdego waloru z osobna z poziomu skryptu AFL

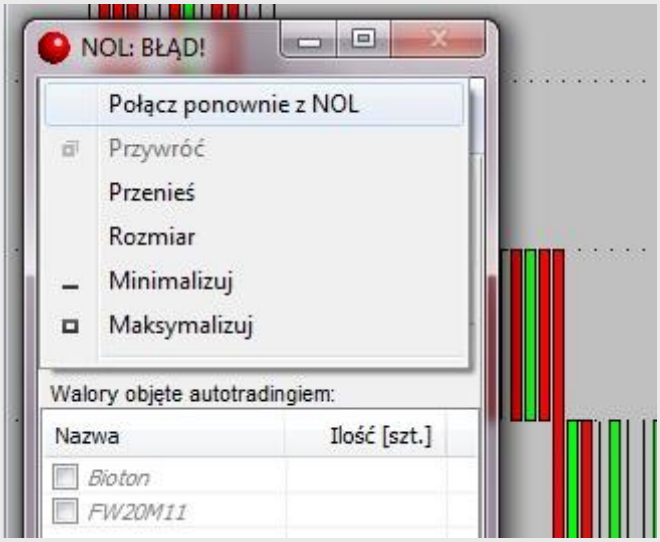
- Lista „**Bieżące zlecenia”** wyświetla listę złożonych zleceń w skrótowej formie (opis poniżej).
- **Walory objęte autotradingiem** – wyświetla listę walorów wybranych w Amibrokerze do tradingu oraz posiadane walory na rachunkach. Kliknięcie prawym przyciskiem myszki na wybranym zleceniu wyświetla menu z akcjami:
  - **Informacje** – wyświetla okienko z podstawowymi informacjami o zleceniu

- **Odśwież** – wysyła żądanie pobrania stanu zlecenia poprzez NOL
- **Anuluj** – anuluje dane zlecenie

Na liście wyświetlane są nazwy walorów zarówno objętych tradingiem jak i zdeponowanych na rachunkach maklerskich. Przyjęto następującą konwencję:
- nazwa wyświetlana *kursywą* w kolorze szarym oznacza walor zdeponowany na rachunku, ale nie podlegający autotradingowi. Takiego waloru nie można zaznaczyć jako włączony do tradingu
- nazwa wyświetlana ***kursywą*** w kolorze czarnym oznacza walor wybrany do autotradingu
- nazwa wyświetlana **pogrubioną** czarną czcionką prostą oznacza walor dla którego włączono autotrading.

Kolumna **Ilość [szt.] (blokada)** informuje o ilości posiadanych walorów dostępnych do sprzedaży. W nawiasie podana jest ilość akcji zablokowana do sprzedaży. W przykładzie użytkownik posiada łącznie 8 akcji BIOTON w tym 5 wolnych do sprzedaży i 3 zablokowane do sprzedaży (złożono dyspozycję sprzedaży na 3 akcje).

Wybranie kursorem myszki lewej ikonki systemowej („kulka", rys. 2) wyświetla systemowe menu z pozycją **„Połącz ponownie z NOL"**. Kliknięcie tej pozycji wymusza ponowne zalogowanie do NOL.



Rys. 2

Kolor ikonki informuje o aktualnym stanie plugina:
- - plugin zalogowany, normalna prędkość transmisji
- - wolna transmisja
- - plugin w trakcie łączenia z NOL lub nieznany stan plugina
- - błąd plugina, plugin odłączony od NOL, lub plugin OFF-Line

Zlecenia w „Historii zleceń" opisane są następującymi ikonami:
- P - zlecenie przyjęte przez biuro maklerskie do realizacji
- W - zlecenie wykonane (może być wykonane częściowo!)
- X - zlecenie anulowane
- O - zlecenie odrzucone przez biuro maklerskie lub GPW
- A - zlecenie archiwalne
- ? - nieznany stan zlecenia, może być np. w trakcie anulaty lub modyfikacji
- B - zlecenie „buy"
- S - zlecenie „sell"

Menu główne plugina:

- **Zlecenia**

    - **Nowe -** uruchamia okno zleceń manualnych
    - **Anuluj wszystkie** – anuluje wszystkie aktywne zlecenia złożone poprzez plugin (także manualne). Anulowanie odbywa się zlecenie po zleceniu.

- **Informacje** – składa się z następujących pozycji podmenu

    - **O zleceniach..** – wyświetla okno ze złożonymi zleceniami (opis dalej)

      Każde zlecenie pokazane jest w postaci: data i czas, BUY lub SELL, nazwa waloru, ilość, identyfikator zlecenia lub opis błędu

    - **O kontach..** – wyświetla informacje o rachunkach użytkownika, rodzaj rachunku, posiadanej gotówce do dyspozycji i wolnym depozycie na danym rachunku.
    - **O limitach..** – podaje informacje o ilości wykonanych zleceń/modyfikacji/anulat i pozostałej ilości takich operacji. Domyślnie dopuszczalne jest 2000 operacji na sesję.
    - **O programie..** – wyświetla krótkie informacje o pluginie.

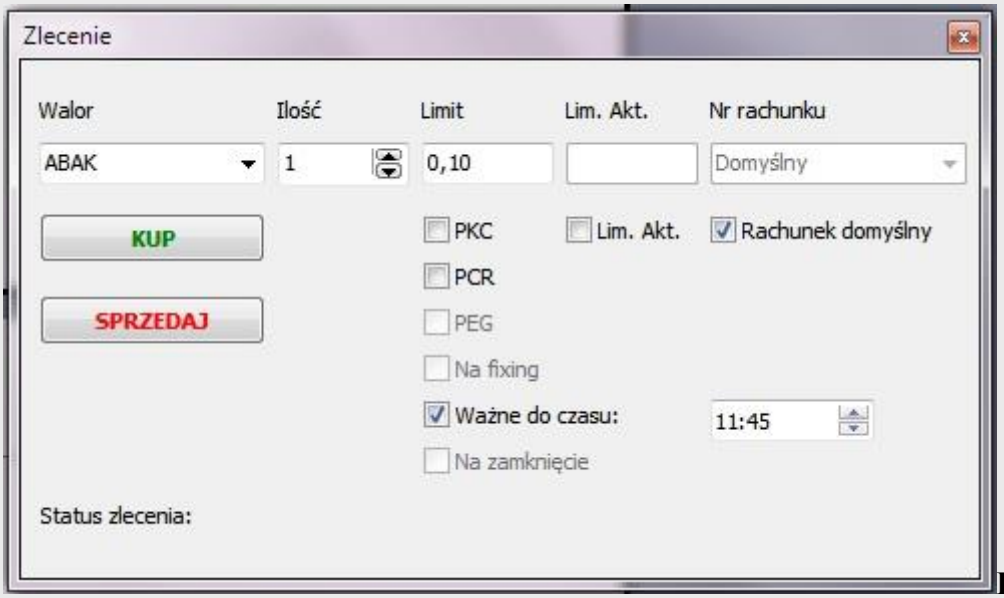
- **Opcje** – daje dostęp do następujących opcji programu:

    - **Ustawienia…** – uruchamia okno ustawień plugina (opis dalej)
    - **Tryb testowy** – włącza/wyłącza tryb testowy plugina. W tym trybie plugin nie składa zleceń. Po uruchomieniu funkcji „**Bar replay**” w Amibrokerze plugin reaguje na pojawiające się sygnały kupna/sprzedaży generując odpowiednie zapisy w okienku z informacjami o zleceniach (**Informacje -> O zleceniach**)
      Tryb testowy można włączyć tylko przy wyłączonym autotradingu!

      Tryb Trace – włącza automatyczne wywoływanie w Amibrokerze funkcji _TRACE przy wywołaniu funkcji zlecenia/modyfikacji/anulaty. Upraszcza śledzenie wywoływnia tych funkcji. Komunikaty _TRACE wyświetlane są oknie „Trace” Amibrokera.
    - **Pokaż okienko logów** – rozszerza okno programu i pokazuje istotne informacje o działaniu plugina.
    - **Wyczyść zabezpieczenia…** – pozwala na zresetowanie zastosowanych zabezpieczeń (poprzez skrypt AFL). Opcja istotna jeżeli użytkownik stosował zabezpieczenia warunków tradingu dla danego waloru i chce je zmienić. Usunięcie zabezpieczeń możliwe jest tylko przy wyłączonym autotradingu. Ponowne ustawienie zabezpieczeń na ogół wiąże się z koniecznością ponownego uruchomienia Amibrokera.

- **Wycisz** – wyłącza alert dźwiękowy

Okienko zleceń manualnych



Rys. 3

Aby je uruchomić należy wybrać menu **Zlecenia->Nowe**

- Rozwijalna lista **„Walor”** daje możliwość wybrania waloru do zlecenia. Na liście wyświetlane są tylko walory, które znajdują się na liście walorów objętych autotradingiem
- Pole edycyjne **„Ilość”** służy do zdefiniowania ilości papierów w zleceniu.
- Pole edycyjne **„Limit”** – tu należy wpisać limit zakupu/sprzedaży.
- Pole edycyjne **„Limit akt.”** – tu należy wpisać limit aktywacji zlecenia.
- Rozwijalna lista **„Nr rachunku”** pozwala wybrać rachunek z którego będzie dokonywana transkacja. Rachunek „**Domyślny**” oznacza, że plugin samodzielnie przydzieli odpowiedni rachunek na podstawie nazwy waloru.
- Pole **„PKC” –** jeśli jest zaznaczone, to zlecenie wysyłane jest jako PKC
- Pole **„PCR” –** jeśli jest zaznaczone, to zlecenie wysyłane jest jako PCR
- Pole **„Lim. Akt.” –** jeśli zaznaczone, to zlecenie posiada limit aktywacji
- Pole **„PEG”** – jeśli zaznaczone, to zlecenie jest zleceniem PEG, jeśli dodatkowo wpisano limit ceny w polu **Limit**, to zlecenie jest zleceniem PEG z limitem.
- Pole **„Na fixing”** - jeśli zaznaczone, to zlecenie ważne jest na najbliższym fixingu
- Pole **„Ważne do czasu”** – jeśli zaznaczone, to zlecenie ważne jest do godziny określonej w polu edycyjnym z prawej strony.
- Pole **„Na zamknięcie”** – jeśli zaznaczone, to zlecenie ważne jest na zamknięciu sesji.

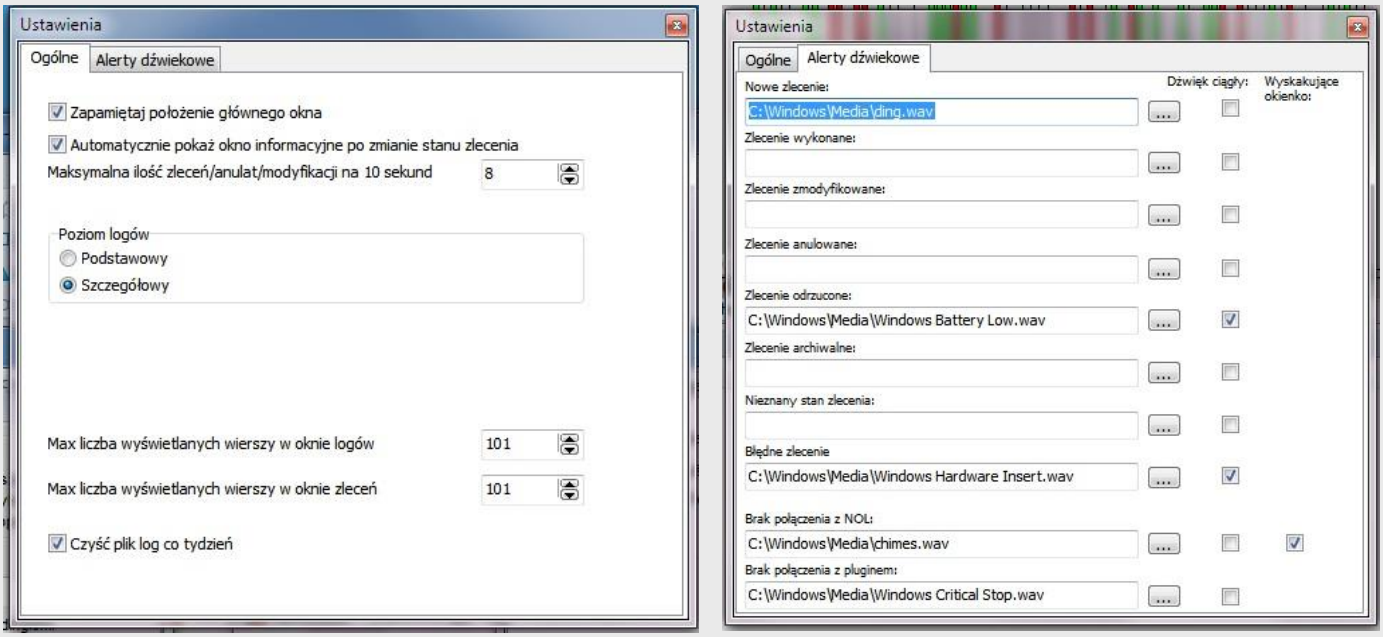
Ceny w zleceniach sprawdzane są pod kątem poprawności formalnej (zapis). Nie są natomiast sprawdzane czy są w danym momencie dopuszczalne przez giełdę.

Aby wysłać zlecenie należy nacisnąć przycisk **„KUP”** lub **„SPRZEDAJ”**. Prawidłowo wysłane zlecenie w polu **„Stan zlecenia”** generuje komunikat **„Zlecenie przyjęte”**, w przeciwnym przypadku pojawi się **„Nie powiodło się wysyłanie zlecenia”**. Pozytywny komunikat o przyjęciu zlecenia nie świadczy o tym, że zlecenie jest ważne – może zostać odrzucone w dalszym etapie przetwarzania przez system transakcyjny lub giełdę.

Zlecenia WDC (Ważne Do Czasu) oznaczone są dodatkowo końcówką **_C**, np. zlecenie kupnaWDC będzie widoczne jako **BUY_C**

Zlecenia **Na fixing** oznaczone są końcówką **_F**.

Okienko ustawień (Opcje -> Ustawienia…)



Rys. 4

Okno ustawień składa się z dwóch zakładek:

- **Ogólne** :
    - **Zapamiętaj położenie głównego okna** – zapamiętuje położenie okna plugina, przy ponownym uruchomieniu Amibrokera okno jest wyświetlane w zapamiętanym położeniu na ekranie.
    - **Automatycznie pokaż okno informacyjne** – po każdej zmianie statusu zlecenia zostanie pokazane okno z informacjami o zleceniach
    - **Maksymalna ilość zleceń/anulat/modyfikacji na 10 sekund** – należy ustawić częstość wymienionych akcji, domyślnie powinno być 8
    - **Poziom logów** – **Podstawowy** i **Szczegółowy**. Na poziomie podstawowym zapisywane są tylko najważniejsze informacje np. o błędach. Logi szczegółowe zawierają treść komunikatów wysyłanych i otrzymywanych do NOL, a także komunikaty wewnętrznej plugina. Przy wielu operacjach plik **NolAMI.log** będzie szybko zwiększał objętość, więc co pewien czas należy go skasować jeżeli nie ma potrzeby analizy komunikatów.
    - **Tryb TRACE –** Jeśli została włączona ta opcja, to każde zlecenie lub anulata/modyfikacja zlecenia wywoła funkcję _TRACE Amibrokera z podstawowymi informacjami o wykonanej akcji. Pomocne przy debugowaniu skryptu AFL
    - **Max liczba wyświetlanych wierszy w oknie…** – okno będzie zawierać nie więcej niż ustalona liczba wierszy. W razie potrzeby najstarsze wpisy zostaną
    - **Czyść plik log co tydzień –** zaznaczenie opcji spowoduje przeniesienie poprzedniego pliku **NolAmi.log** do **NolAmi_kopia.log** (poprzedni zostanie usunięty) i utworzenie nowego pliku **NolAmi.log**. Opcja służy oszczędności miejsca na dysku.

- **Alerty dźwiękowe** – tu ustawia się ścieżki dostępu do plików dźwiękowych dla poszczególnych alertów. Przy każdym polu edycyjnym znajduje się przycisk „**…**” do

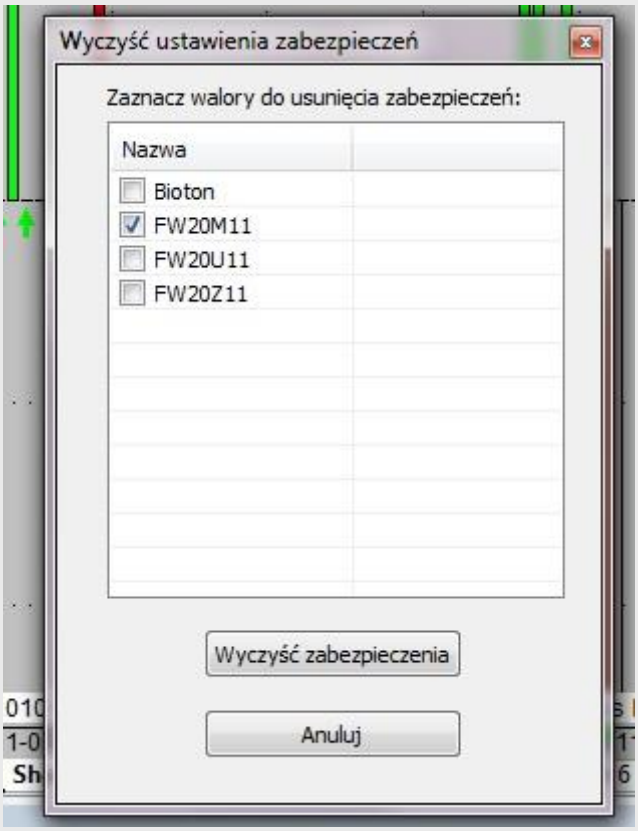
uruchomienia okna dialogowego wczytywania nazwy dźwięku. Zaznaczenie opcji **Dźwięk ciągły** powoduje, że sygnał będzie generowany w zamkniętej pętli.

Alert dźwiękowy można wyłączyć pozycją menu **Wycisz**.

Dodatkowo przy opcji **Brak połączenia z NOL** znajduje się pole opcji wyskakującego okienka w przypadku braku połączenia z NOL lub gdy NOL przejdzie w stan OFF-Line.

Okno alarmujące o problemie z NOL jest chowane automatycznie gdy NOL ponownie uzyska status ON-Line.

Okienko usuwania zabezpieczeń (Opcje -> Wyczyść zabezpieczenia…)



Rys. 5

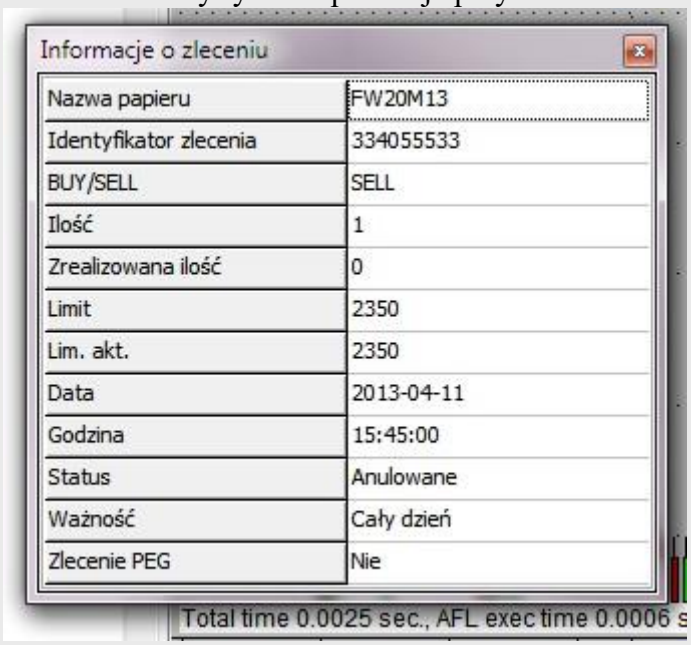
Okno wyświetla listę walorów. Aby usunąć zabezpieczenia należy zaznaczyć odpowiednie walory i nacisnąć przycisk **Wyczyść zabezpieczenia**. Informacja o usunięciu zabezpieczeń pojawi się w okienku logów.
Uwaga! Opcji należy używać bardzo ostrożnie, bo może zmienić warunki tradingu.

Działanie tej opcji może interferować z wykonywanymi kodami AFL: jeżeli w kodzie AFL używa się funkcji ustawiających zabezpieczenia wykonywane przy każdym odświeżaniu wykresu, to usuwanie zabezpieczeń będzie nieskuteczne lub doprowadzi do ich zmiany (jeżeli np. zmieniono jednocześnie nazwę waloru na wykresie).

Poziom zabezpieczeń można kontrolować także z poziomu funkcji plugina za pomocą funkcji nolSetSecurity .

Okienko z informacjami o zleceniu

Poniższy rysunek pokazuje przykładowe okno z informacjami o zleceniu:



Informacje o zleceniu można uzyskać wybierając odpowiednie zlecenie z listy zleceń i klikając na nim prawym przyciskiem myszki i wybierając z menu opcję „**Informacje**”

7. Spis funkcji udostępnianych przez plugin

nolInit	- inicjalizacja i logowanie do NOL
nolMode	- pobranie trybu pracy
nolSetTrading	- włącza/wyłącza trading dla danego waloru
nolOnlyReverse	- włączanie/wyłącznaie dla waloru transakcji naprzemiennych
nolSetInterval	- ustawianie interwału dla danego waloru
nolGetSecurity	- pobranie ustawionych zabezpieczeń
nolSetSecurity	- ustawienie zabezpieczeń dla wybranego waloru
nolSetChartID	- ustawienie identyfikatora wykresu dla danego waloru
nolBuy	- zlecenie kupna
nolBuyPEG	- zlecenie kupna PEG
nolSell	- zlecenie sprzedaży
nolSellPEG	- zlecenie sprzedaży PEG
nolCancel	- anulowanie zlecenia
nolModify	- modyfikacje zlecenia
nolAskState	- żądanie pobrania stanu zlecenia z serwera
nolGetState	- pobranie stanu zlecenia z NOL
nolAddTicker	- dodanie waloru do listy w okienku zleceń manualnych
nolPositions	- pobranie ilości pozycji waloru
nolBlockedPositions	- pobranie ilości papierów zablokowanych do sprzedaży
nolPositionsAcc	- pobranie ilości pozycji waloru na określonym rachunku
nolBlockedPositionsAcc	- pobranie ilości papierów zablokowanych do sprzedaży na określonym rachunku
nolCash	- pobranie ilości gotówki na rachunku
nolFreeDeposit	- pobranie wartości należności/wolnego depozytu
nolGetError	- pobranie numeru ostatniego błędu
nolErrorDescription	- pobranie opisu błędu
nolConnection	- sprawdzenie stanu połączenia z NOL
nolSessionStat	- sprawdzenie fazy sesji dla waloru
nolVersion	- zwraca numer wersji plugina w postaci liczby 5-o cyfrowej
nolFIXML	- wysyła komunikat FIXML podany przez klienta

8. Opis funkcji

Funkcje inicjalizujące:

nolInit()

Funkcja inicjalizująca plugin i logująca do NOL
Wymagana!
Powinna być wywoływana jednokrotnie.

Parametry: brak

Zwracana wartość: 0 (zero) jeśli nastąpiło poprawne zalogowanie do NOL

Typowe wywołanie funkcji to

```
nolInit();
```

---

nolMode(string walor)

Funkcja zwracająca tryb pracy plugina dla danego waloru.
Nie wymagana.

Parametry:

- ***walor*** - zmienna łańcuchowa: nazwa waloru do kupienia, jeśli nazwa jest pusta, to plugin pobiera nazwę instrumentu z wykresu Amibrokera

Zwracana wartość: 0 – autotrading dla waloru został wyłączony
1 – tryb testowy, autotrading wyłączony
2 – uruchomiony autotrading
3 – tryb testowy, uruchomiony autotrading

Typowe wywołanie funkcji to

```
If(nolMode("")%2) printf("\nWłączony tryb testowy");
Wyświetla w okienku Interpretation Amibrokera informację o włączonym trybie testowym
```

---

nolSetTrading(string walor, int onoff)

Funkcja włączająca/wyłączająca autotrading dla danego waloru.

Parametry

walor - zmienna łańcuchowa: nazwa waloru do kupienia, jeśli nazwa jest pusta, to plugin pobiera nazwę instrumentu z wykresu Amibrokera

onoff - zmienna float, jeśli >0, to autotrading jest włączony dla danego waloru pod warunkiem, że w interfejsie włączono kontrolę autotradingu z poziomu AFL. Parametr 0 wyłącza autotrading dla waloru.

Zwracana wartość: zawsze 0 (zero) lub numer błędu.

Typowe wywołanie funkcji to

```
nolSetTrading("BIOTON", 1);
```

---

nolOnlyReverse(string walor, int reverse)

Funkcja włączająca/wyłączająca blokadę piramidowania pozycji. Zabezpieczenie.
Typowo wywołuje się jednokrotnie w sekcji inicjalizacyjnej.
Nie wymagana, zalecana aby zwiększyć bezpieczeństwo tradingu. Przy włączonym zabezpieczeniu kolejne zlecenie musi być odwrotne do poprzedniego.
Zabezpieczenie obowiązuje jedynie w trakcie jednej sesji giełdowej.

Parametry

walor - zmienna łańcuchowa: nazwa waloru do kupienia, jeśli nazwa jest pusta, to plugin pobiera nazwę instrumentu z wykresu Amibrokera

reverse - zmienna float, jeśli !=0, to plugin nie pozwala na składanie kolejnych zleceń w tym samym kierunku (zablokowane piramidowanie pozycji)

Zwracana wartość: zawsze 0 (zero) , funkcja nie sprawdza poprawności danych

Typowe wywołanie funkcji to

```
nolOnlyReverse("BIOTON", 1);
```

---

nolSetInterval(string walor, float interval)

Funkcja ustawiająca interwał dla waloru. Zabezpieczenie.
Typowo wywołuje się jednokrotnie w sekcji inicjalizacyjnej.
Nie wymagana, zalecana aby zwiększyć bezpieczeństwo tradingu. Przypadkowa zmiana interwału na wykresie Amibrokera przy ustawionym interwale w pluginie powoduje zablokowanie składania zleceń dla danego waloru.

Parametry:

walor - zmienna łańcuchowa: nazwa waloru do kupienia, jeśli nazwa jest pusta, to plugin pobiera nazwę instrumentu z wykresu Amibrokera

interval - zmienna float; ustawiana wartość interwału: dodatnia w sekundach dla interwałów czasowych, ujemna dla interwałów tickowych

Zwracana wartość: 0 (zero) lub -19 w przypadku niepowodzenia.

Typowe wywołanie funkcji to

```
nolSetInterval("BIOTON", 900);
```
Ustawia na walorze BIOTON interwał 15 min

---

nolGetSecurity(string walor, float sec)

Funkcja pobierająca zabezpieczenie dla danego waloru.

Parametry:

walor - zmienna łańcuchowa: nazwa waloru do kupienia, jeśli nazwa jest pusta, to plugin pobiera nazwę instrumentu z wykresu Amibrokera

sec - zmienna float: liczba całkowita wskazująca który z parametrów zabezpieczenia ma zwrócić funkcja:

- 1 – znacznik włączenia autotradingu, 0 – wyłączony, 1 – włączony, -1 włączony dla wybranego waloru
- 2 – identyfikator wykresu
- 3 – interwał (dodatni w sekundach, ujemny w tikach)
- 4 – ostatni kierunek zlecenia, 1 – kupno, 2 – sprzedaż
- 5 – znacznik czasu, czas ostatniego zlecenia jako liczba w formacie hhmmss
- 6 – znacznik blokady piramidowania, 0 – brak blokady

Zwracana wartość: wartość zabezpieczenia lub błąd nr -23 – Uwaga! W przypadku pobierania interwału kod błędu może być potraktowany jako interwał tikowy (23 tiki)
Wartość 0 oznacza brak ustawionego zabezpieczenia

Typowe wywołanie funkcji:

```
interwal = nolGetSecurity(”BIOTON”, 3);
```

---

nolSetSecurity(string walor, float sec)

Funkcja włączająca/wyłączająca poszczególne zabezpieczenia dla danego waloru.

Parametry:

walor - zmienna łańcuchowa: nazwa waloru do kupienia, jeśli nazwa jest pusta, to plugin pobiera nazwę instrumentu z wykresu Amibrokera

sec - zmienna float: liczba całkowita wskazująca które z zabezpieczeń ma być włączone a które wyłączone. Kolejne bity binarnej reprezentacji liczby sec włączają/wyłączają następujące zabezpiecznie (bit ustawiony oznacza włączenie zabezpiecznia)

- 1 – zabezpiecznie identyfikatora wykresu chartID (najmłodszy bit)
- 2 – zabezpiecznie interwału
- 3 – zabezpiecznie znacznika czasu (godzina)
- 4 – zabezpiecznie kierunku zlecenia
- 5 – zabezpiecznie znacznika daty

Zwracana wartość: 0 jeśli operacja wykonana bez błędu, w przeciwnym przypadku numer błędu.

Aby włączyć wszystkie zabezpieczenia należy ustawić argument sec=31 (sec=b011111, wszystkie znaczące bity ustawione). Aby pozostawić tylko zabezpieczenie czasu i daty (domyślne ustawienie dla każdego waloru) sec=20 (b010100).

Aby efektywnie włączyć zabezpieczenia 1,2,3 należy je wcześniej ustawić za pomocą odpowiednich fukcji (nolSetChartID, nolSetInterval, nolOnlyReverse). Funkcja **nolSetSecurity** jedynie włącza/wyłącza stosowanie zabezpiecznia bez ustawiania ich parametrów!

Wyłączanie zabezpieczeń za pomocą tej funkcji pozwala na zachowanie stanu odpowiednich parametrów, np. znacznika czasu, daty czy kierunku ostatniego zlecenia i prawidłowego ich odtworzenia po ponownym włączeniu zabezpieczenia.

Usunięcie zabezpiecznia daty i/lub czasu spowoduje, że nazwa waloru w okienku interfesju będzie wyświetlana w kolorze czerwonym.

Uwaga:

1) Usunięcie zabezpiecznia znacznika czasu i/lub daty może skutkować wysłaniem zleceń, które byłyby prawidłowe w przeszłości, ale nie w danym momencie. Może też skutkować wysyłaniem wielokrotnych zleceń.
2) Wyłączenie zabezpieczenia daty powinno być stosowane jedynie w przypadku, gdy klient chce składać zlecenia przed otwarciem sesji GPW.
3) Mimo wyłączenia zabezpieczeń czasu i daty wysłanie zlecenia modyfikuje odpowiednie znaczniki czasu i daty!

Typowe wywołanie funkcji:

```
res = nolSetSecurity(”BIOTON”, 21);
```

Zabezpieczenie interwału i piramidowania zostaną wyłączone, pozostałe zabezpieczenia są włączone, stan zabezpiecznia identyfikatora wykresu zależy od wcześniejszego wywołania funkcji nolSetChartID

nolSetChartID(string walor, float id)

Funkcja ustawiająca identyfikator dla waloru. Zabezpieczenie.
Nie wymagana, zalecana aby zwiększyć bezpieczeństwo tradingu.
Powinna być użyta w sekcji inicjalizacyjnej.
Jeżeli dla danego waloru zostanie zdefiniowany identyfikator wykresu (ChartID w Amibrokerze), to zlecenia będą wysyłane tylko w przypadku gdy wykres o danym ChartID przedstawia właśnie ten walor. Ponadto nie będą możliwe zlecenia dla danego waloru na innych wykresach o innych identyfikatorach. Należy pamiętać, że możliwe jest otwarcie nowego wykresu o zduplikowanym identyfikatorze (cecha korzystania z szablonów Amibrokera), może to prowadzić do zmniejszenia skuteczności tego zabezpieczenia. W takich przypadkach warto stosować także zabezpieczenie ustawiające interwał nolSetInterval

Parametry:

walor - zmienna łańcuchowa: nazwa waloru do kupienia, jeśli nazwa jest pusta, to plugin pobiera nazwę instrumentu z wykresu Amibrokera

id - zmienna float (liczba całkowita); ustawiana wartość identyfikatora

Zwracana wartość: wartość identyfikatora lub kod błędu: -21 – ChartID już został uprzednio zdefiniowany dla waloru, -22 – brak struktury opisującej zabezpieczenia dla waloru

Typowe wywołanie funkcji:

```
nolSetChartID("BIOTON", 1002);
```

Funkcje zleceń:

nolBuy(string walor, string rachunek, float ilość, float limit, float limakt, float czas)

Funkcja wysyłająca zlecenie kupna waloru

Parametry:

- ***walor*** - zmienna łańcuchowa: nazwa waloru do kupienia, jeśli nazwa jest pusta, to plugin pobiera nazwę instrumentu z wykresu Amibrokera
- ***rachunek*** - zmienna łańcuchowa: numer rachunku, jeśli jest pusta, to plugin automatycznie przyporządkowuje rachunek
- ***ilość*** - zmienna float (liczba całkowita): liczba papierów do kupienia
- ***limit*** – limit ceny w zleceniu. Wartość 0 (zero) oznacza zakup PKC.
- ***limakt*** – limit aktywacji ceny w zleceniu. Wartość **0** (zero) oznacza brak limitu aktywacji. Domyślna wartość 0.
- ***czas*** – godzina ważności zlecenia. Czas podawany w formacie HHMMSS, np. godz. 16:30:05 przekazywana jest jako 163005. Wartość **0** (zero) oznacza zlecenie ważne cały dzień, wartość **-1** (minus jeden) oznacza zlecenie ważne na najbliższy fixing. Domyślna wartość 0. **Uwaga! Funkcja akceptuje dowolną wartość sekund, ale są one ignorowane - wewnątrz plugina ustawiana jest zerowa część sekundowa.**

Zwracana wartość: zmienna łańcuchowa z identyfikatorem zlecenia. Identyfikator należy zapamiętać w zmiennej statycznej jeśli przewiduje się np. możliwość anulowania zlecenia. W przypadku nieprawidłowych danych (np. niepoprawny limit względem limitu aktywacji) lub nieudanego wysłania zlecenia, zwracany jest pusty łańcuch **””**.
Uwaga! Pusty łańcuch zwracany jest także w przypadku gdy zlecenie jest poprawne formalnie, ale już zostało wysłane dla danego sygnału Buy/Sell/Short/Cover ze skryptu AFL.

Uwagi: przy składaniu zleceń należy zadbać o prawidłowe wygenerowanie sygnału w skrypcie AFL. Np. proste przecięcie się dwóch średnich może generować sygnał w danej świeczce w czasie rzeczywistym wielokrotnie, w zależności od chwilowej ceny close tejże świeczki. Plugin nie dopuszcza do wysłania zlecenia wcześniejszego lub z tej samej chwili co poprzednio wysłane zlecenie (jeśli było). Tak więc, plugin zabezpiecza przed wielokrotnym wysłaniem zlecenia dla danego zlecenia jeśli wykres odświeżany jest periodycznie (na skutek użycia funkcji `RequestTimedRefresh();`), nie zabezpiecza jednak gdy sygnały powstają w różnych momentach czasowych!
W dalszej części (**Przykłady**) pokazane są prawidłowe i nieprawidłowe sposoby wysyłania zleceń.

Przykładowe wywołania funkcji:

`nolBuy("", "", 100, 0, 0);` – kupno 100 szt papierów z bieżącego wykresu po cenie PKC na domyślnym rachunku

`nolBuy("", "", 100, 5.20, 5.15);` - kupno 100 szt papierów z bieżącego wykresu po maksymalnej cenie 5.20 wyzwolone limitem aktywacji 5.15

nolBuyPEG(string walor, string rachunek, float ilość, float limit, float czas)

Funkcja wysyłająca zlecenie PEG kupna waloru

Parametry:

walor - zmienna łańcuchowa: nazwa waloru do kupienia, jeśli nazwa jest pusta, to plugin pobiera nazwę instrumentu z wykresu Amibrokera

rachunek - zmienna łańcuchowa: numer rachunku, jeśli jest pusta, to plugin automatycznie przyporządkowuje rachunek

ilość - zmienna float (liczba całkowita): liczba papierów do kupienia

limit – limit ceny w zleceniu PEG. Wartość 0 (zero) oznacza zlecenie bez limitu.

czas – godzina ważności zlecenia. Czas podawany w formacie HHMMSS, np. godz. 16:30:05 przekazywana jest jako 163005. Wartość **0** (zero) oznacza zlecenie ważne cały dzień. Domyślna wartość 0. **Uwaga! Funkcja akceptuje dowolną wartość sekund, ale są one ignorowane - wewnątrz plugina ustawiana jest zerowa część sekundowa.**

Zwracana wartość: zmienna łańcuchowa z identyfikatorem zlecenia. Identyfikator należy zapamiętać w zmiennej statycznej jeśli przewiduje się np. możliwość anulowania zlecenia. W przypadku nieprawidłowych danych (np. niepoprawny limit względem limitu aktywacji) lub nieudanego wysłania zlecenia, zwracany jest pusty łańcuch **””**.
Uwaga! Pusty łańcuch zwracany jest także w przypadku gdy zlecenie jest poprawne formalnie, ale już zostało wysłane dla danego sygnału Buy/Sell/Short/Cover ze skryptu AFL.

Uwagi: przy składaniu zleceń należy zadbać o prawidłowe wygenerowanie sygnału w skrypcie AFL. Np. proste przecięcie się dwóch średnich może generować sygnał w danej świeczce w czasie rzeczywistym wielokrotnie, w zależności od chwilowej ceny close tejże świeczki. Plugin nie dopuszcza do wysłania zlecenia wcześniejszego lub z tej samej chwili co poprzednio wysłane zlecenie (jeśli było). Tak więc, plugin zabezpiecza przed wielokrotnym wysłaniem zlecenia dla danego zlecenia jeśli wykres odświeżany jest periodycznie (na skutek użycia funkcji `RequestTimedRefresh();`), nie zabezpiecza jednak gdy sygnały powstają w różnych momentach czasowych!
W dalszej części (**Przykłady**) pokazane są prawidłowe i nieprawidłowe sposoby wysyłania zleceń.

Przykładowe wywołania funkcji:

`nolBuyPEG("", "", 100, 0);` - kupno 100 szt papierów z bieżącego wykresu po najwyższej cenie oferty kupna, na rachunku domyślnym

`nolBuyPEG("", "", 50, 5);` - kupno 50 szt papierów z bieżącego wykresu po najwyższej cenie oferty kupna ale nie wyższej niż 5PLN, na rachunku domyślnym

`nolBuyPEG("", "", 100, 0, 153000);` - kupno 100 szt papierów z bieżącego wykresu po najwyższej cenie oferty kupna, na rachunku domyślnym, ważne do godz. 15:30

nolSell(string walor, string rachunek, int ilość, float limit, float limakt, float czas)

Funkcja wysyłająca zlecenie sprzedaży waloru

Parametry:

walor - zmienna łańcuchowa: nazwa waloru do sprzedania, jeśli nazwa jest pusta, to plugin pobiera nazwę instrumentu z wykresu Amibrokera

rachunek - zmienna łańcuchowa: numer rachunku, jeśli jest pusta, to plugin automatycznie przyporządkowuje rachunek

ilość - zmienna float (liczba całkowita): liczba papierów do sprzedania

limit – limit ceny w zleceniu. Wartość 0 (zero) oznacza sprzedaż PKC.

limakt – limit aktywacji ceny w zleceniu. Wartość 0 (zero) oznacza brak limitu aktywacji. Domyślna wartość 0.

czas – godzina ważności zlecenia. Czas podawany w formacie HHMMSS, np. godz. 16:30:05 przekazywana jest jako 163005. Wartość **0** (zero) oznacza zlecenie ważne cały dzień, wartość **-1** (minus jeden) oznacza zlecenie ważne na najbliższy fixing. Domyślna wartość 0. **Uwaga! Funkcja akceptuje dowolną wartość sekund, ale są one ignorowane - wewnątrz plugina ustawiana jest zerowa część sekundowa.**

Zwracana wartość: zmienna łańcuchowa z identyfikatorem zlecenia. Identyfikator należy zapamiętać w zmiennej statycznej jeśli przewiduje się np. możliwość anulowania zlecenia. W przypadku nieprawidłowych danych (np. niepoprawny limit względem limitu aktywacji) lub nieudanego wysłania zlecenia, zwracany jest pusty łańcuch **””**.
Uwaga! Pusty łańcuch zwracany jest także w przypadku gdy zlecenie jest poprawne formalnie, ale już zostało wysłane dla danego sygnału Buy/Sell/Short/Cover ze skryptu AFL.

Uwagi: przy składaniu zleceń należy zadbać o prawidłowe wygenerowanie sygnału w skrypcie AFL. Np. proste przecięcie się dwóch średnich może generować sygnał w danej świeczce w czasie rzeczywistym wielokrotnie, w zależności od chwilowej ceny close tejże świeczki. Plugin nie dopuszcza do wysłania zlecenia wcześniejszego lub z tej samej chwili co poprzednio wysłane zlecenie (jeśli było). Tak więc, plugin zabezpiecza przed wielokrotnym wysłaniem zlecenia dla danego zlecenia jeśli wykres odświeżany jest periodycznie (na skutek użycia funkcji `RequestTimedRefresh();`), nie zabezpiecza jednak gdy sygnały powstają w różnych momentach czasowych!
W dalszej części (**Przykłady**) pokazane są prawidłowe i nieprawidłowe sposoby wysyłania zleceń.

Przykładowe wywołania funkcji:

`nolSell("", "", 100, 0, 0);` - sprzedaż 100 szt papierów z bieżącego wykresu po cenie PKC na domyślnym rachunku

`nolSell("", "", 100, 0, 2400);` - sprzedaż 100 szt papierów z bieżącego wykresu po cenie PKC wyzwolone limitem aktywacji 2400

nolSellPEG(string walor, string rachunek, int ilość, float limit, float czas)

Funkcja wysyłająca zlecenie PEG sprzedaży waloru

Parametry:

walor - zmienna łańcuchowa: nazwa waloru do sprzedania, jeśli nazwa jest pusta, to plugin pobiera nazwę instrumentu z wykresu Amibrokera

rachunek - zmienna łańcuchowa: numer rachunku, jeśli jest pusta, to plugin automatycznie przyporządkowuje rachunek

ilość - zmienna float (liczba całkowita): liczba papierów do sprzedania

limit – limit ceny w zleceniu PEG. Wartość 0 (zero) oznacza sprzedaż bez limitu.

czas – godzina ważności zlecenia. Czas podawany w formacie HHMMSS, np. godz. 16:30:05 przekazywana jest jako 163005. Wartość **0** (zero) oznacza zlecenie ważne cały dzień, wartość **-1** (minus jeden) oznacza zlecenie ważne na najbliższy fixing. Domyślna wartość 0. **Uwaga! Funkcja akceptuje dowolną wartość sekund, ale są one ignorowane - wewnątrz plugina ustawiana jest zerowa część sekundowa.**

Zwracana wartość: zmienna łańcuchowa z identyfikatorem zlecenia. Identyfikator należy zapamiętać w zmiennej statycznej jeśli przewiduje się np. możliwość anulowania zlecenia. W przypadku nieprawidłowych danych (np. niepoprawny limit względem limitu aktywacji) lub nieudanego wysłania zlecenia, zwracany jest pusty łańcuch **””**.
Uwaga! Pusty łańcuch zwracany jest także w przypadku gdy zlecenie jest poprawne formalnie, ale już zostało wysłane dla danego sygnału Buy/Sell/Short/Cover ze skryptu AFL.

Uwagi: przy składaniu zleceń należy zadbać o prawidłowe wygenerowanie sygnału w skrypcie AFL. Np. proste przecięcie się dwóch średnich może generować sygnał w danej świeczce w czasie rzeczywistym wielokrotnie, w zależności od chwilowej ceny close tejże świeczki. Plugin nie dopuszcza do wysłania zlecenia wcześniejszego lub z tej samej chwili co poprzednio wysłane zlecenie (jeśli było). Tak więc, plugin zabezpiecza przed wielokrotnym wysłaniem zlecenia dla danego zlecenia jeśli wykres odświeżany jest periodycznie (na skutek użycia funkcji `RequestTimedRefresh();`), nie zabezpiecza jednak gdy sygnały powstają w różnych momentach czasowych!
W dalszej części (**Przykłady**) pokazane są prawidłowe i nieprawidłowe sposoby wysyłania zleceń.

Przykładowe wywołania funkcji:

`nolSellPEG("", "", 100, 0);` - sprzedaż 100 szt papierów z bieżącego wykresu po najniższej cenie oferty sprzedaży, na rachunku domyślnym

`nolSellPEG("", "", 100, 14, 160500);` - sprzedaż 100 szt papierów z bieżącego wykresu po najniższej cenie oferty sprzedaży ale nie taniej niż po 14PLN, na rachunku domyślnym, zlecenie traci ważność o godz. 16:05

nolCancel(string id)

Funkcja anulująca zlecenie

Parametry:

id - zmienna łańcuchowa: identyfikator zlecenia

Zwracana wartość: 0 (zero) w przypadku pomyślnego wysłania żądania anulowania. W przeciwnym wypadku kod błędu.

Przykładowe wywołania funkcji:

```
nolCancel("222071829");
```
– anulowanie zlecenia o identyfikatorze 222071829

---

nolModify(string id, float side, float ilość, float limit, float limakt, float czas)

Funkcja wysyłająca zlecenie kupna bieżącego waloru

Parametry:

id - zmienna łańcuchowa: nazwa waloru do sprzedania, jeśli nazwa jest pusta, to plugin pobiera nazwę instrumentu z wykresu Amibrokera
side - zmienna float (liczba całkowita): kierunek zlecenia: 1 – kupno, 2 - sprzedaż
ilość - zmienna float (liczba całkowita): liczba papierów w zleceniu
limit – limit ceny w zleceniu. Wartość 0 (zero) oznacza zlecenie PKC; -1 - PCR
limakt – limit aktywacji ceny w zleceniu. Wartość 0 (zero) oznacza brak limitu aktywacji
czas – godzina ważności zlecenia. Czas podawany w formacie HHMMSS, np. godz. 16:30:05 przekazywana jest jako 163005. Wartość **0** (zero) oznacza zlecenie ważne cały dzień, wartość **-1** (minus jeden) oznacza zlecenie ważne na najbliższy fixing. **Uwaga! Funkcja akceptuje dowolną wartość sekund, ale są one ignorowane - wewnątrz plugina ustawiana jest zerowa część sekundowa.**

Zwracana wartość: zmienna łańcuchowa z identyfikatorem zlecenia. Identyfikator należy zapamiętać w zmiennej statycznej jeśli przewiduje się np. możliwość anulowania zlecenia. W przypadku nieprawidłowych danych (np. niepoprawny limit względem limitu aktywacji) lub nieudanego wysłania zlecenia, zwracany jest pusty łańcuch **””**.
Parametry które można zmodyfikować zależą od wersji API NOL.
Uwaga! Pusty łańcuch zwracany jest także w przypadku gdy zlecenie jest poprawne formalnie, ale już zostało wysłane dla danego sygnału Buy/Sell/Short/Cover ze skryptu AFL.

Przykładowe wywołanie funkcji:

```
nolModify("222071829", 1, 100, 0, 0);
```
– zlecenie nr 222071829 będzie zleceniem kupna 100 szt papierów po cenie PKC

---

nolAskStat (string id)

Funkcja wysyłająca żądanie pobrania statusu zlecenia z serwera

Parametry:

- *id* - zmienna łańcuchowa:
  identyfikator zlecenia otrzymany w zwrocie z funkcji kupna/sprzedaży.

Zwracana wartość: float, wartość nieujemna w przypadku pomyślnej odpowiedzi.
-1 w przypadku błędu (np. błąd transmisji).
Możliwe zwracane wartości:

0 - nowe
67 - archiwalne
69 - w trakcie modyfikacji
1 - wykonane/aktywne
2 - wykonane
4 - anulowane
6 - w trakcie anulaty
8 - odrzucone

Przykładowe wywołanie funkcji:

```
nolAskStat("222071829");
```

– żądanie pobrania statusu zlecenia o identyfikatorze 222071829

---

nolGetStat (string id)

Funkcja pobierająca statusu zlecenia z bufora NOL

Parametry:

- *id* - zmienna łańcuchowa, identyfikator zlecenia otrzymany w zwrocie z funkcji kupna/sprzedaży.

Zwracana wartość: float, wartość nieujemna w przypadku pomyślnej odpowiedzi.
-1 w przypadku błędu (np. błąd transmisji).
Możliwe zwracane wartości jak w funkcji nolAskState

Uwagi: funkcja pobiera ostatnio transmitowany do NOL stan zlecenia więc nie ma pewności co do jego aktualności.

Przykładowe wywołanie funkcji:

```
Status = nolGetStat("222071829");
```

– żądanie pobrania statusu zlecenia o identyfikatorze 222071829

Pozostałe funkcje:

nolAddTicker(string walor)

Funkcja dodająca walor do listy w okienku interfejsu.
Dzięki temu można wykonywać zlecenia manualne poprzez Plugin na danym walorze.
Opcjonalna. Zalecane stosowanie w sekcji inicjalizacyjnej jednokrotnie wykonywanej.

Parametry:

walor - zmienna łańcuchowa: nazwa waloru do kupienia, jeśli nazwa jest pusta, to plugin pobiera nazwę instrumentu z wykresu Amibrokera

Zwracana wartość: wartość interwału lub 0 (zero) w przypadku niepowodzenia.

Typowe wywołanie funkcji:

```
nolAddTicker("FW20M11");
```

---

nolPositions(string walor)

Funkcja pobierająca ilość pozycji waloru.

Parametry:

- ***walor*** - zmienna łańcuchowa: nazwa waloru, jeśli nazwa jest pusta, to plugin pobiera nazwę instrumentu z wykresu Amibrokera

Zwracana wartość: ilość papierów zapisanych na rachunku
-9000000 w przypadku błędu (np. Brak przyporządkowania kodu Isin do nazwy waloru lub brak informacji o rachunkach).

Wywołanie funkcji:

```
nolPositions("");
```

---

nolBlockedPositions(string walor)

Funkcja pobierająca ilość pozycji zablokowanych do sprzedaży.

Parametry:

- ***walor*** - zmienna łańcuchowa: nazwa waloru, jeśli nazwa jest pusta, to plugin pobiera nazwę instrumentu z wykresu Amibrokera

Zwracana wartość: ilość papierów na rachunku zablokowanych do sprzedaży
-9000000 w przypadku błędu (np. Brak przyporządkowania kodu Isin do nazwy waloru lub brak informacji o rachunkach).

Wywołanie funkcji:

```
nolBlockedPositions("");
```

---

nolPositionsAcc(string walor, rachunek)

Funkcja pobierająca ilość pozycji waloru z określonego rachunku

Parametry:

- ***walor*** - zmienna łańcuchowa: nazwa waloru, jeśli nazwa jest pusta, to plugin pobiera nazwę instrumentu z wykresu Amibrokera
- **rachunek** - zmienna łańcuchowa: numer rachunku, jeśli nazwa jest pusta, to plugin pobiera dane z domyślnego rachunku

Zwracana wartość: ilość papierów zapisanych na rachunku
-9000000 w przypadku błędu (np. Brak przyporządkowania kodu Isin do nazwy waloru lub brak informacji o rachunkach).

Przykładowe wywołanie funkcji:

```
nolPositions("", 00-22-123456");
```

---

nolBlockedPositionsAcc(string walor, rachunek)

Funkcja pobierająca ilość pozycji zablokowanych do sprzedaży.

Parametry:

- ***walor*** - zmienna łańcuchowa: nazwa waloru, jeśli nazwa jest pusta, to plugin pobiera nazwę instrumentu z wykresu Amibrokera
- **rachunek** - zmienna łańcuchowa: numer rachunku, jeśli nazwa jest pusta, to plugin pobiera dane z domyślnego rachunku

Zwracana wartość: ilość papierów na rachunku zablokowanych do sprzedaży
-9000000 w przypadku błędu (np. Brak przyporządkowania kodu Isin do nazwy waloru lub brak informacji o rachunkach).

Przykładowe wywołanie funkcji:

```
nolBlockedPositions("","00-22-123456");
```

---

nolCash (string rachunek)

Funkcja pobierająca dostępną gotówkę na rachunku

Parametry:

rachunek - zmienna łańcuchowa:
numer rachunku w formacie 00-xx-xxxxxx gdzie x – cyfry dziesiętne

Zwracana wartość: float, wartość nieujemna w przypadku pomyślnej odpowiedzi.
-9000000 w przypadku błędu (np. nieznany numer rachunku it.p.)

Uwagi: funkcja pobiera ostatnio transmitowany do NOL stan gotówki więc nie ma pewności co do jego aktualności. Nie ma też możliwośći wymuszenia odświeżenia stanu gotówki.
Ze względu na to, że Amibroker korzysta z liczb typu float, zwracana wartość może być zaokrąglona.

Przykładowe wywołanie funkcji:

```
Printf("\nDostępna gotówka " + nolCash("00-55-123456"));
```

Wyświetla stan gotówki z rachunku akcyjnego

---

nolFreeDeposit (string rachunek)

Funkcja pobierająca należności (rachunek akcji) lub wolny depozyt (derywaty)

Parametry:

rachunek - zmienna łańcuchowa:
numer rachunku w formacie 00-xx-xxxxxx gdzie x – cyfry dziesiętne

Zwracana wartość: float, wartość nieujemna w przypadku pomyślnej odpowiedzi.
-9000000 w przypadku błędu (np. nieznany numer rachunku it.p.)

Uwagi: funkcja pobiera ostatnio transmitowany do NOL stan gotówki więc nie ma pewności co do jego aktualności. Nie ma też możliwośći wymuszenia odświeżenia stanu gotówki.
Ze względu na to, że Amibroker korzysta z liczb typu float, zwracana wartość może być zaokrąglona.

Przykładowe wywołanie funkcji:

```
Printf("\nDostępna gotówka " + nolFreeDeposit("00-55-
123456"));
```

Wyświetla stan należności z rachunku akcyjnego

---

nolGetError ()

Funkcja pobierająca kod ostatniego błędu w pluginie.

Parametry: brak

Zwracana wartość: kod błędu lub 0 (zero)

Przykładowe wywołanie funkcji:

```
error = nolGetError();
```

---

nolErrorDescription (float error)

Funkcja pobierająca kod ostatniego błędu w pluginie.

Parametry:

error - zmienna typu float (liczba całkowita), kod błędu

Zwracana wartość: łańcuch znakowy z opisem błędu

Przykładowe wywołanie funkcji:

```
printf(nolErrorDescription(-12));
```

wyświetla w okienku Interpretation Amibrokera opis błędu dla kodu -12 (błędna data)

---

nolConnection(float login)

Funkcja zwracająca stan połączenia plugina z aplikacją NOL lub wymuszająca logowanie do NOL.

Parametry:

login – zmienna wskazująca czy ma być wykonane ponowne logowanie do NOL

Jeżeli parametr ***login***>0, to plugin wykonuje operację wylogowania i ponownego zalogowania do NOL.

Zwracana wartość: liczba całkowita binarna wskazująca na stan połączenia w momencie wywołania funkcji. Aby odczytać stan połączenia należy wykonać operację maskowania bitów. Znaczenie kolejnych bitów jest następujące (gdy ustawiony):

0 – plugin połączony z NOL
1 – plugin odłączony od NOL
2 – plugin zalogowany do NOL
3 – plugin wylogowany z NOL
7 – stan połączenia nieznany
8 – NOL Off-line
9 – NOL On-line

Przykładowe wywołanie funkcji:

```
Res = (nolErrorDescription(0) & (1<<8));
If(Res) printf("NOL OFF-LINE");
```

Uwaga Nie należy nadużywać wymuszania logowania do NOL ponieważ może to zakłócić płynność działania komunikacji z NOL. Ponowne logowanie powinno mieć miejsce tylko w przypadku rozłączenia lub wylogowania z NOL (np. na skutek wyłączenia aplikacji NOL).

---

nolSessionStat(string walor)

Funkcja pobierająca fazę sesji dla danego waloru.

Parametry:

- *walor* - zmienna łańcuchowa: nazwa waloru, jeśli nazwa jest pusta, to plugin pobiera nazwę instrumentu z wykresu Amibrokera

Zwracana wartość: liczba całkowita liczba określająca fazę sesji dla danego waloru:

1 – obrót zawieszony
2 – poranny monitoring
3 – przed otwarciem
4 – otwarcie
5 – notowania ciągłe
6 – preopening na zamknięcie
7 – zamknięcie
8 – dogrywka
9 – po sesji
0 – faza nie rozpoznana

Przykładowe wywołanie funkcji:

```
Res = nolSessionStat("");
If(Res==5) printf(„Notowania ciągłe”);
```

---

nolVersion()

Funkcja zwracająca numer wersji plugina.

Zwracana wartość: liczba całkowita liczba wersję plugina.

Przykładowe wywołanie funkcji:

```
Printf("Plugin ver. " + nolVersion(""));
```

---

nolFIXML(string fix, string id, float del)

Funkcja wysyłająca do aplikacji NOL komunikat FIXML zadeklarowany przez klienta..

Parametry:

- *fix* - zmienna łańcuchowa: komunikat FIXML.
- *id* - zmienna łańcuchowa: nazwa atrybutu identyfikatora
- *del* - liczba: wskazuje czy usunąć plik **clientFIXML.txt**.

Komunikat FIXML powinien być sformatowany zgodnie z dokumentacją API NOL. Nie może zawierać otwierającego fragmentu `<FIXML v="5.0" r="20080317" s="20080314">` jest on dopisywany przez plugin. Każdy z komunikatów API zawiera atrybut identyfikatora komunikatu (np. ID, StatReqID itp.), każdy z nich musi mieć właściwą wartość, która nadawana jest przez plugin. Wartość tego atrybutu w wysyłanym komunikacie musi wynosić **%d** zamiast wartości liczbowej.

Aby skasować plik Aby usunąć plik **clientFIXML.txt** należy użyć wartości *del* większej od zera. W tym przypadku pierwsze dwa argumenty funkcji mogą być puste.

Ewentualne zwrotne komunikaty asynchroniczne zapisywane są w pliku **clientFIXML.txt** w katalogu Amibrokera, ich interpretacja należy do użytkownika.

Zwracana wartość: łańcuch znakowy zawierający odpowiedź NOL lub NULL w przypadku błędu.

Uwaga: Nie wolno wysyłać komunikatów ważnych dla stabilności działania plugina: logowanie/wylogowanie, aby nie zakłócić pracy plugina.

Przykładowe wywołanie funkcji:

```
printf(nolFIXML("<SecListReq ReqID=\"%d\" ListReqTyp=\"0\" ><Instrmt Sym=\"FW20M1620\" /></SecListReq>", "ReqID", 0));
```

Zwraca komunikat zawierający informacje o walorze FW20M1620.

---

9. Instrukcja

Ustawienia Amibrokera

Do poprawnego działania zabezpieczeń wbudowanych czasowych w plugin niezbędne jest użycie odpowiednich ustawień Amibrokera. Plugin przed wysłaniem zlecenia pobiera czas ostatniej transakcji (dla danego waloru) i zapamiętuje go, aby dla tej samej sygnatury czasowej nie ponowić zlecenia, nawet gdy odświeżenie wykresu ponownie wygeneruje sygnał BUY/SELL. Czas 0 (zero) traktowany jest jako błędny, dlatego wymagane jest zaznaczenie w ustawieniach bazy danych „**File->Database settings->Intraday settings**" opcji „**Allow mixed EOD/Intraday data**" (przy posługiwaniu się źródłem danych ze Statici oraz innymi dostarczającymi dane EOD i intraday wspólnie).
Należy także wybrać **Tools->Preferences->Intraday** i zaznaczyć opcję **time of LAST tick inside bar**.
Korzystając ze źródła danych firmy **Statica** (**Notowania**) należy wybrać **Długie nazwy** – kliknij prawym przyciskiem na ikonę stanu plugina **Notowań** w prawym dolnym rogu okna **Amibrokera** i zaznacz tę opcję.

Szablony w Amibrokerze

W Amibrokerze do tworzenia wykresów często używane są gotowe szablony. Korzystanie z gotowych szablonów jest wygodne ale powoduje, że wszystkie okna wykresów uzyskują taki sam identyfikator wykresu (zwracany funkcją **GetChartID**). W automatycznych systemach napisanych w języku AFL praktycznie niezbędne jest używanie zmiennych statycznych. Powinny one być unikalne dla każdego skryptu AFL aby nie wpływały na działanie innych skryptów.
Łatwo sprawdzić, że zmienne statyczne są widziane przez wszystkie uruchomione formuły AFL. W tym celu wyedytuj treść kodu na wykresie (kliknij prawym przyciskiem myszki na wykresie i wybierz **Edit formula**) i wpisz np. `StaticVarSetText("test_var", "Jestem widoczny!\n");`
Zapisz formułę i zamknij okno edytora. Teraz wybierz okno z innym wykresem, wyedytuj jego formułę i wpisz `printf(StaticVarGetText("test_var"));` Zapisz formułę i kliknij na wykres. W okienku **Interpretation** pojawi się napis „**Jestem widoczny!**" Należy pamiętać żeby operacje te przeprowadzić na wykresach utworzonych **nie** poprzez szablony lub **Default Chart**, w przeciwnym wypadku wszelkie zmiany w formule pojawią się w formułach z tego samego szablonu.
W okienku **Parameters** każdego wykresu można sprawdzić jaki identyfikator ma wykres (**ChartID**).
Jeżeli korzystasz z generatora kodu AFL **bossaABKreator**, to trzeba pamiętać, że kreator tworzy unikalne nazwy zmiennych statycznych na bazie **ChartID** oraz nazwy waloru. Jeżeli obie zmienne są identyczne na dwóch lub więcej arkuszach w Amibrokerze, to zmienne statyczne w ten sposób wygenerowane nie będą już unikalne i doprowadzi to do trudnych do przewidzenia skutków ubocznych (wzajemne interakcje między różnymi arkuszami). Dlatego korzystając z szablonów należy każdorazowo ręcznie zmienić nazwy zmiennych statycznych aby uzyskać ich niepowtarzalność. Można np. do każdej już zdefiniowanej zmiennej dodać kolejne liczby:
`svInit = "Init" + symbol + chartID + "12";`
Tu pierwotna zmienna statyczna **svInit** wygenerowana przez **bossaABKreator** została zmodyfikowana przez dodanie liczby „12".

Inicjalizacja

Po poprawnym zainstalowaniu plugina jego funkcje stają się dostępne z poziomu języka skryptowego AFL Amiborkera®.
Aby wykorzystać możliwość automatycznego składania zleceń należy zainicjalizować połączenie z NOL.
Jak wiadomo, aktywny skrypt AFL Amibrokera (czyli taki, który „naniesiony" został na wykres) wykonywany jest po każdej transakcji.
Dlatego pożądane jest aby wywołania funkcji, których wykonanie wymagane jest jednokrotnie, rzeczywiście były wykonywane jeden raz.
Przyspiesza to interpretację skryptu. Tym nie mniej, funkcje plugina zawierają własne zabezpieczenia i np. inicjalizacja może być wywoływana wielokrotnie: po pierwszej inicjalizacji następne wywołania nic nie robią.

Proponowana przykładowa sekwencja inicjalizacyjna w skrypcie:

```
// tu następuje definicja nazw istotnych zmiennych statycznych:

svInit = "Init" + Name()+GetChartID();                              // zapamiętuje stan
inicjalizacji
svCanBuy = "CanBuy" + Name()+NumToStr(GetChartID(),1.0,False);    // wskazuje czy można
składać zlecenia kupna
svCanSell = "CanSell" + Name()+NumToStr(GetChartID(),1.0,False); // wskazuje czy można
składać zlecenia sprzedaży
svIdBuy = "IdBuy" + Name()+NumToStr(GetChartID(),1.0,False);      // przechowuje
identyfikator zlecenia kupna
svIdSell = "IdSell" + Name()+NumToStr(GetChartID(),1.0,False);    // przechowuje
identyfikator zlecenia sprzedaży

if(Status("action")==1)                                 // inicjalizacja następuje tylko w
czasie operacji na wykresie
if(Nz(StaticVarGet(svInit))==0)                         // jeśli plugin ma działać także w
trybie SCAN, to należy
                                                        // zastosować konstrukcję
if(Status("action")==1 || Status("action")==3)
{
  StaticVarSet(svInit, 1);                              // zapewnia jednokrotne uruchomienie
inicjalizacji
  StaticVarSet(svCanBuy, 1);                            // włącza możliwość kupna
  StaticVarSet(svCanSell, 1);                           // włącza możliwość sprzedaży

  nolInit();
  nolSetChartID("");
  nolSetInterval("", 900);
}
```

Poprawne będzie także wywoływanie tych funkcji poza blokiem „if”:

```
if(Nz(StaticVarGet(svInit))==0)
{
  StaticVarSet(svInit, 1);
      // tu inne inicjalizacje, np. zmiennych

}

nolInit(); // komenda inicjalizacji plugina
nolSetChartID("");
nolSetInterval("", 900);
```

Trzeba jednak pamiętać, że w każdej z tych funkcji wykonywany jest pewien kod, co niepotrzebnie obciąża procesor.

Kod inicjalizacyjny musi znajdować się na wykresie każdego waloru dla którego użytkownik chce uruchomić automatyczny trading.

Jeśli użytkownik planuje korzystać z ręcznego wysyłania zleceń z wykorzystaniem plugina na walorze dla którego nie zainicjalizowano plugina, to może być niezbędne użycie funkcji nolAddTicker.
Dodaje ona nazwę waloru do listy walorów w pluginie. Powinna być umieszczona w skrypcie każdego wykresu, którego walor chcemy umieścić na liście do zleceń manualnych (Okienko zleceń manualnych).

Po zainicjalizowaniu połączenia z NOL plugin gotowy jest do wysyłania zleceń. Służą do tego dwie podstawowe funkcje nolBuy i nolSell.
W opisie tych funkcji dodano uwagi o konieczności zapewnienia jednokrotnego generownia sygnałów. Niestety, nie można tego zadania zautomatyzować gdyż zależy ono od konkretnego systemu transakcyjnego. Poniżej przedstawiony zostanie przykład poprawnego zaimplementowania tych funkcji.
Założenia: system generuje sygnał zajęcia pozycji długiej gdy cena **Close** na zamknięciu świeczki będzie wyższa od średniej **MA(C, 14)**.
Zamknięcie pozycji nastąpi gdy cena **Close** będzie poniżej średniej.

```
if(Nz(StaticVarGet(svInit))==0)                                    // wykorzystano wcześniej
zdefiniowaną zmienną statyczną svInit
{
       //… niezbędne inicjalizacje

}
srednia = MA(Close, 14);
Buy = Cross(Ref(Close, -1), Ref(srednia, -1));                      // wektor z sygnałami
kupna
Sell = Cross(Ref(srednia,-1), Ref(Close, -1));                      // wektor z sygnałami
sprzedaży

id_Buy = "";
id_Sell = "";

mBuy = 0;
mSell = 0;

mBuy = LastValue(Buy, false);
mSell = LastValue(Short, false);

if(mBuy) id_Buy = nolBuy("", "", 2, 0, 0);                  // kupno 2 szt. PKC
if(StrLen(id_Buy)) StaticVarSetText("LastOrderBuy", id_Buy);

if(mSell) id_Sell = nolSell("", "", 2, 0, 0);                // sprzedaż 2 szt. PKC
if(StrLen(id_Sell)) StaticVarSetText("LastOrderSell", id_Sell);
```

W powyższym kodzie warto zwrócić uwagę na sposób generowania sygnałów. Wykorzystano w tym celu funkcję AFL `Ref(…)` .
Przecięcie (`Cross`) dotyczy tu zamknięcia poprzedniej świeczki z wartością średniej w poprzedniej świeczce.
Dzięki temu, porównujemy rzeczywiste zamknięcie świeczki z ustabilizowaną wartością średniej, sygnał pada (lub nie) na otwarciu ostatniej świeczki (rozważamy system w czasie rzeczywistym). Jeśli zastosujemy narzucającą się konstrukcję: **`Buy`** `= Cross(`**`Close`**`, srednia);`, to w każdej chwilowej transakcji ostatniej świeczki funkcja `Cross` może zwracać różne wartości. To oczywiście generowałoby liczne sygnały, niekonieczne zgodne z wynikiem gdy świeczka zostanie ostatecznie uformowana. Ponieważ sygnały w tym przypadku będą miały różne sygnatury czasowe, to zostanie wysłanych wiele zleceń.
Uwaga! Powyższa konstrukcja ma sens w przypadku interwałów intraday. Dla interwałów dziennych może być niewłaściwa.

Zmienna **id_Buy** zawiera identyfikator zlecenia. Jeśli nie jest on pusty (sprawdzamy długość łańcucha funkcją `StrLen(id_Buy)`, to zapamiętujemy go w zmiennej statycznej `LastOrderBuy`. Jest to konieczne jeśli chcemy np. sprawdzać stan zlecenia, albo je anulować. Następne bowiem wywołania funkcji nolBuy będą zwracać pusty łańcuch (aż do wystąpienia następnego takiego typu sygnału i pomyślnego złożenia zlecenia).

Funkcje zleceń nolBuy i nolSell pobierają nazwę waloru, pobierają czas wystąpienia sygnału, przyporządkowują właściwy rachunek i wysyłają zlecenie do NOL. Czas wystąpienia sygnału jest bardzo ważnym parametrem, dzięki temu następny sygnał z czasem starszym zostanie zignorowany. Pomyślnie wysłane zlecenia zapisywane są w pliku **orders_list.txt**.

Powyższy przykład jest skrajnie uproszczony dla czytelności. W rzeczywistym systemie należy sprawdzić czy pozycje zostały zajęte i jaka jest ich wielkość, aby zlecenia zamykające posługiwało się odpowiednią liczbą papierów w zleceniu.

10. Dodatkowe informacje i uwagi

Jak zwiększyć bezpieczeństwa tradingu?

Zalecane postępowanie:

- Na wykresie Amibrokera należy zablokować kłódką nazwę waloru na wykresie (w dolnej belce wykresu)
- Należy stosować funkcje włączające zabezpieczenia nolSetInterval, nolSetChartID, nolOnlyReverse
- W kodzie AFL należy stosować własne zabezpieczenia (zwłaszcza zapewnienie jednoznacznego definiowania sygnałów **Buy**/**Sell**)
- Po włączeniu autotradingu nie należy zmieniać interwału oraz nazwy waloru na wykresie
- Nie ingerować w kod AFL w trakcie jego pracy bez uprzednich testów
- Przed rzeczywistym handlem należy gruntownie przetestować kod AFL w trybie testowym Plugina.

Pomocne może być też uruchomienie kodu w trybie rzeczywistym ale z funkcjami nolBuy i nolSell zawierającymi celowo błędne dane, aby zlecenia nie były wykonywane. W takim przypadku zostaną wygenerowane komunikaty o błędach – pomogą w sprawdzeniu czy funkcje te są wykonywane w żądanych momentach.

W kodzie warto umieścić instrukcje _TRACE(…) Amibrokera wypisujące w okienku zakładki **Trace** komunikaty generowane przez użytkownika. Pomaga to śledzić pracę kodu w czasie rzeczywistym.

Przełączanie interwału w czasie pracy w trybie „autotrading" może prowadzić do powstawania fałszywych sygnałów lub niewykonania zleceń dla sygnałów prawidłowych! Użytkownik powinien we własnym zakresie zabezpieczyć się przed taką sytuacją.

O czym warto zawsze pamiętać konstruując automatyczny system transakcyjny?

- Włącz odpowiednie ustawienia w Amibrokerze (p. **9 Ustawienia**)
- Używaj arkuszy o indywidualnych identyfikatorach wykresu (**ChartID**), w tym celu twórz je poprzez polecenie **File->New->Blank Chart** i dopiero na pusty arkusz nanieś potrzebne wykresy (ceny, średnie, itp.)
- Tam gdzie można w wywołaniach funkcji plugina wstawiaj nazwy walorów dla których napisany jest system oraz posługuj się numerami rachunków – zwiększa to bezpieczeństwo transakcji i nieco przyspiesza wykonywany kod AFL
- Dokładnie przeanalizuj w którym momencie powstają sygnały kupna/sprzedaży w Twoim systemie i odpowiednio dostosuj kod AFL (lub ustaw odpowiednią opcję w kreatorze **bossaABKreator**)
- Kod generowany przez **bossaABKreator** jest zasadniczo przeznaczony dla operacji intraday, tzn nie ma możliwości pobrania stanu zleceń z poprzednich sesji. Dotyczy to także samego plugina. Jeżeli zamierzasz tworzyć systemy bardziej skomplikowane i np. średnio- lub długoterminowe może być konieczne dopisanie kodu, który przechowuje (i odczytuje) w pliku stan ostatniego sygnału kupna/sprzedaży oraz stan jego realizacji.
- Jeżeli chcesz zamykać wszystkie pozycje na koniec sesji, to należy powielić moduły kupna/sprzedaży i obudować je blokami sprawdzającymi bieżący czas, np.:

```
if(Now(4)<165000)     // kupno tylko przed godziną 16:50 według systemowych sygnałów
{
   if(mBuy) idBuy = nolBuy("sdg", "12345", 1, 0, 0);
   // ... pozostały kod
}

StaticVarSet(svCanSell, 1);
if(Now(4)>165000)     // sprzedaż wszystkich walorów na koniec sesji PKC
{
   If(StaticVarGet(svCanSell)) idSell = nolSell("sdg", "12345", liczba, 0, 0);
   If(StrLen(idSell)>0) StaticVarSet(svCanSell, 0);
   // ... pozostały kod
}
```

Nie ma jednak żadnej gwarancji, że powyższy kod zamknie pozycje – zależy to także od tego czy ktokolwiek na rynku zechce dokonać odwrotnej transakcji. W przykładzie w zleceniu jest cena PKC, ale taka konstrukcja ma sens praktycznie tylko dla bardzo płynnych walorów.

11. Lista błędów zwracanych przez plugin.

-1	Błąd komunikacji synchronicznej
-2	Błąd komunikacji asynchronicznej
-3	Błąd gniazda
-4	Błąd logowania
-5	Błąd wylogowania
-6	Błąd filtra papierów
-7	Błąd odczytu danych portów
-8	Aplikacja NOL nieaktywna
-9	Błąd kreacji wątku asynchronicznego
-10	Plugin uż zainicjalizowany
-11	Nieznane konto
-12	Nieaktualna data
-13	Nieaktualna godzina (HHMMSS)
-14	Nieznane zlecenie
-15	Nie otrzymano identyfikatora zlecenia
-16	Nieprawidłowy parametr zlecenia, pusty walor lub ilość=0
-17	Wykonywanie innego polecenia synchronicznego
-18	Zlecenie nie jest odwrotne do poprzedniego
-19	Interwał niezgodny z zadeklarowanym
-20	Autotrading dla waloru wyłączony
-21	Niewłaściwy identyfikator wykresu
-22	Brak opisu security dla waloru
-23	Nie można włączyć/wyłączyć tradingu
-30	Błąd odczytu danych portów Interfejsu
-31	Nie można połaczyć się z Interfejsem
-32	Nie można utworzyć wątku dla Interfejsem
-33	Nie można uruchomić Interfejsu
-34	Błąd gniazda Interfejsu
-35	Błąd komunikacji z Interfejsem
-40	Plugin nie zalogowany
-41	PLugin off-line
-42	Plugin odłączony
-43	Plugin wylogowany
-45	Błąd wysyłania FIXML klienta
-46	Błąd alokacji pamięci