

System CDN OPTIMA v. 15.0



Moduł Analizy
Edycja wykresów

31-864 Kraków, Al. Jana Pawła II 41g
tel. (12) 681 43 00, fax (12) 687 71 00

Dział Wsparcia Klienta i Partnera: (12) 681 43 00
www.comarch.pl/cdn
info.cdn@comarch.pl



Pomyśl o środowisku zanim zdecydujesz się wydrukować ten dokument !

1 Spis treści

1	SPIS TREŚCI	2
2	OGÓLNA IDEA WYKRESÓW	3
3	USZCZEGÓLOWIENIE KONCEPCJI.....	5
3.1	HIERARCHICZNA STRUKTURA WYKRESÓW (NAWIGACJA PO GRAFIE WYKRESÓW).....	5
3.1.1	PRZYCISKI WYKRESÓW NA TOOLBARZE, PUNKTY WYWOŁANIA WYKRESÓW	5
3.1.2	WYKRESY STARTOWE, DOMYŚLNY I BIEŻĄCY WYKRES STARTOWY.....	5
3.1.3	ZAGNIEŹDŻANIE WYKRESÓW (PODWYKRESY).....	5
3.1.4	GRAF WYKRESÓW – NAWIGACJA W CZASIE WYKONYWANIA	6
3.1.5	EDYCJA WYKRESÓW.....	6
3.2	ŹRÓDŁA DANYCH DO WYKRESÓW.....	6
3.3	PARAMETRY DYNAMICZNE WYKRESÓW	7
3.4	ZMIENNE DYNAMICZNE PRZEKAZYWANE Z APLIKACJI DO BIEŻĄCEGO WYKRESU STARTOWEGO	8
3.5	WYRAŻENIA	8
3.6	PREDEFINIOWANE WSTAWKI TEKSTOWE.....	9
3.6.1	WSTAWKI TEKSTOWE	9
3.6.2	TEKSTY WARUNKOWE.....	10
3.6.3	ROZSZERZONE PARAMETRY TEKSTOWE (PARAMETRY TYPU X).....	10
3.7	PARAMETRY GRAFICZNE WYKRESU	10
3.8	WYWOŁYWANIE FUNKCJI	12
3.9	WYKRESY Z KOLEJKI	12
3.10	WYKRESY BEZ WYKRESÓW - TABELA Z WYNIKAMI	12
3.11	PRZEKAZYWANIE DO WYKRESÓW ZAZNACZONYCH POZYCJI.....	12
3.12	WYSYŁANIE WYNIKÓW WYKRESU	12
4	INTERFACE UŻYTKOWNIKA.....	14
4.1	EDYCJA WYKRESÓW	14
4.1.1	DRZEWO HIERARCHII WYKRESÓW	14
4.1.2	FORMULARZ WYKRESU.....	17
4.1.3	LISTA WYKRESÓW	19
4.1.4	LISTA WSTAWEK TEKSTOWYCH DO WYKRESÓW.....	20
4.1.5	FORMULARZ WSTAWKI TEKSTOWEJ DO WYKRESU.....	20
4.2	OKNO OPERACYJNE WYKRESÓW	20
5	IMPLEMENTACJA TECHNICZNA.....	23
5.1	SILNIK WYKRESÓW	23
5.2	BAZA DANYCH.....	24
5.2.1	LOKALIZACJA DANYCH	24
5.3	WYDRUKI WYKRESÓW.....	24
6	PRZYKŁAD	25

2 Ogólna idea wykresów

Idea wykresów polega na stworzeniu jednolitego, spójnego mechanizmu pozwalającego m.in. na wygodne definiowanie wykresów (inaczej: analiz) prezentujących graficznie i/albo tabelarycznie wybrane dane z bazy, podłączanie wykresów w przewidzianych (przez programistę) punktach programu i wykonywanie. Jest to więc mechanizm analogiczny jak w przypadku wydruków. Z jedną zasadniczą różnicą. Mianowicie wydruki są z samej swej natury pasywne – wykresy natomiast posiadają bardzo ważną cechę aktywności.

Aktywność wykresu polega na możliwości jego interakcji z Użytkownikiem, w tym także na możliwości wykonania wykresu podrzędnego po kliknięciu przez Użytkownika w wybrany punkt. Interakcja następuje po wykonaniu przez Użytkownika odpowiedniej akcji na bieżącym wykresie i może mieć postać:

▪ Eksploracji wykresu

Eksploracja polega na wybraniu jednego z punktów bieżącego wykresu i zagłębieniu się w niego to znaczy wykonaniu wykresu podrzędnego dla danych pobranych z tego punktu.

Można np. kliknąć na 'kawałku tortu' obrazującym łączne obroty na wybranym rejestrze kasowym/bankowym i dostać podwykres obrazujący rozbitcie wartości tego 'kawałka' na obroty poszczególnych raportów. Mamy tu do czynienia z tzw. akcją *drill-down*. Następnie można wybrać jeden z raportów i, klikając w punkt jemu odpowiadający, wyświetlić listę zapisów składających się na obroty tego raportu (ta z kolei akcja, polegająca na wyświetleniu listy a nie podwykresu jest nazywana *drill-through*).

Oczywiście mamy możliwość powrotu z eksploracji do poprzedniego wykresu (tzw. nadwykresu; jest to tzw. akcja *drill-up*) aż do wykresu, od którego zaczęliśmy. Całość przypomina więc surfowanie po stronach WWW – nieprzypadkowo zatem, jak się przekonamy później, okno wynikowe z wykresami posiada wiele analogii do przeglądarki internetowej.

▪ Przeładowania (odświeżenia) danych na wykresie

Po zmianie przez Użytkownika parametrów wpływających na dane na wykresie jest możliwe przeładowanie tych danych, a więc powtórne wykonanie wykresu. Odświeżenie wykresu może również być wykonane (na żądanie) bez zmiany jakichkolwiek wartości parametrów (np. ze względu na możliwość zmian dokonanych w bazie w związku z normalną pracą tej albo innych stacji roboczych).

▪ Zmiany parametrów graficznych wykresu

Wśród możliwych do zmiany parametrów mogą się także znajdować parametry graficzne nie wpływające na same dane wykresu ale określające jego postać graficzną (np. powiększenie, typ wykresu, wymiarowość: 3D/2D,...). Te parametry również można zmieniać na bieżąco z poziomu okna z wykresem.

▪ Wydrukowania wykresu na drukarce graficznej

Wydruk wykresu jest realizowany standardowym mechanizmem wydruków. Będzie o tym mowa w rozdziale poświęconym wydrukowi wykresów.

▪ Zapisu danych z wykresu do pliku (z opcjonalnym przekazaniem pliku do innego programu)

Zapis danych z wykresu do pliku może być dokonywany w jednym ze standardowych formatów (COMMA, XLS, WMF, XML/XSL). Lokalizację pliku docelowego podaje Użytkownik. Taki plik może być następnie podany do otwarcia innemu programowi (np. Notepad.exe, Excel.exe itp.).

Integracja modułu wykresów z resztą systemu polega na udostępnieniu przez moduł wykresów mechanizmu podobnego jak w przypadku wydruków: w dowolnym oknie aplikacji programista może udostępnić obsługę analiz.

Z punktu widzenia Użytkownika wykresy są zwykle dostępne w tych samych punktach programu (tzw. punkty wywołania wykresów, w skrócie: wywołania), co wydruki. Do każdego punktu wywołania można podłączyć dowolną ilość wykresów (tzw. wykresów startowych), z których jeden jest domyślny (domyślność analiz można zmieniać przez przełączanie specjalnej flagi określonej na poziomie podłączenia, a w przypadku nie ustawienia tej flagi na żadnym z połączeń wykresów startowych domyślnym wykresem jest pierwszy w kolejności lp – kolejność tą można zmieniać wykorzystując mechanizm bąbelkowania).

Wykresy mogą być zagnieżdżane tzn. każdy wykres może mieć dowolną ilość podłączonych do niego (pod)wykresów (dzieci). Z drugiej strony ten sam wykres może być równocześnie podłączony do dowolnej ilości (nad)wykresów (ojcowie). Zagnieżdżanie wykresów może zawierać zapętlenia, tzn. wykres A może mieć podwykres B, który z kolei posiada podwykres A (co więcej, wykres może być podłączony nawet do samego siebie).

Edycja (dodawanie, modyfikacja, testowanie, usuwanie) wykresów, ich podłączanie (także warunkowe) do punktów wywołania oraz do siebie nawzajem, ustalanie domyślności i kolejności a także różne inne operacje są dostępne z poziomu specjalnego drzewa hierarchii wykresów.

Wykresy, podobnie jak wydruki, są podzielone na dwie grupy:

- wykresy standardowe (dostarczane przez CDN w pliku *cdn_wykr.tps*)
- wykresy użytkownika (definiowane w bazie SQL).

To czy wykres jest wykresem standardowym czy Użytkownika określamy mianem **bazy wykresu**.

Edycja wykresów standardowych przez Użytkownika jest bardzo ograniczona – sprowadza się do zmiany dostępności wykresu (globalny / operatora lub stanowiska), zmiany ustawień opcji wysyłania wyniku oraz przełączania domyślności wykresów.

Wyniki wykonania wykresów (ich graficzna postać i/albo dane tabelaryczne składające się na serie danych wykresu) są umieszczane na specjalnym oknie wynikowym analiz. Okno to stanowi dla Użytkownika właściwe środowisko pracy z analizami. To tu Użytkownik widzi wyniki analizy, może zmieniać parametry i przeliczać wykres, tu także może nawigować po grafie wykresów (eksplorować wykresy czyli przechodzić do podwykresów i wracać z powrotem) oraz

wykonywać inne, pomocnicze operacje (np. drukowanie wyników, ich zapis do pliku w wybranym formacie czy wysyłanie wyników analizy do innych programów).

3 Uszczegółowienie koncepcji

3.1 Hierarchiczna struktura wykresów (nawigacja po grafie wykresów)

3.1.1 Przyciski wykresów na toolbarze, punkty wywołania wykresów

Jak powiedziano na początku, w programie istnieją specjalne punkty (tzw. punkty wywołania wykresów, w skrócie: wywołania), do których można podłączać analizy. Od strony Użytkownika punkty te są zazwyczaj tożsame z punktami podłączania wydruków: tam gdzie na toolbarze pojawiają się przyciski wydruków, zwykle pojawiają się także kontrolki do obsługi wykresów. Są to dwa przyciski: przycisk z ikoną wykresu, którego naciśnięcie powoduje wykonanie domyślnego wykresu startowego oraz przycisk z ikoną strzałki w dół powodujący rozwinięcie popup-menu z wyborem wykresu startowego, który należy wykonać. Wykonanie akcji na jednej z kontrolki wykresu powoduje załadowanie danych i otwarcie okna z wynikami (wykresem w postaci graficznej i/albo tabelarycznej).

Okno z wynikami posiada własne przyciski wydruków; pozwala to na definiowanie wydruków analiz. Ponieważ jest jedno wspólne okno wynikowe, na którym pojawiają się wszystkie wykresy – definicje wydruków dla wszystkich wykresów w systemie są wspólne; może tak być bo zawsze będziemy robić wydruk na podstawie kolejki o ustalonej strukturze, w której zgromadzone są dane wykresu.

3.1.2 Wykresy startowe, domyślny i bieżący wykres startowy

Do każdego wywołania można podłączyć dowolną ilość wykresów – są to tzw. wykresy startowe danego wywołania. Wykresy te są odpowiednio posortowane (wg lp) a jeden z nich jest domyślnym wykresem startowym danego wywołania. Domyślność wykresu startowego oznacza dla Użytkownika najprostszą możliwość wykonania wykresu – poprzez jeden klik lewym przyciskiem myszy w główny przycisk analiz. Każdy inny wykres startowy w tym punkcie może być wykonany tylko poprzez jego wybór z menu kontekstowego.

Określanie domyślnego wykresu startowego danego punktu wywołania przebiega wg następującego algorytmu:

1. Sprawdzamy czy do wywołania (np. kontekstu) jest podłączony wykres użytkownika z zaznaczoną flagą *Domyślny*. Jeśli tak – wykres ten jest domyślnym wykresem startowym – kończymy poszukiwanie.

Jeśli nie:

2. Sprawdzamy czy do kolejnych gałęzi nadrzędnych wywołania (np. procedury, potem roota) jest podłączony wykres użytkownika z zaznaczoną flagą *Domyślny*. Jeśli tak – wykres ten jest domyślnym wykresem startowym – kończymy poszukiwanie.

Jeśli nie:

3. Sprawdzamy czy do wywołania (np. kontekstu) jest podłączony wykres standardowy z zaznaczoną flagą *Domyślny*. Jeśli tak – wykres ten jest domyślnym wykresem startowym – kończymy poszukiwanie.

Jeśli nie:

4. Sprawdzamy czy do kolejnych gałęzi nadrzędnych wywołania (np. procedury, potem roota) jest podłączony wykres standardowy z zaznaczoną flagą *Domyślny*. Jeśli tak – wykres ten jest domyślnym wykresem startowym – kończymy poszukiwanie.

Jeśli nie:

5. Pobieramy pierwszy w kolejności lp wykres startowy użytkownika podłączony do wywołania a w przypadku jego braku do kolejnych gałęzi nadrzędnych wywołania. Jeśli taki wykres znajdziemy – wykres ten jest domyślnym wykresem startowym – kończymy poszukiwanie.

Jeśli nie:

6. Pobieramy pierwszy w kolejności lp wykres startowy standardowy podłączony do wywołania a w przypadku jego braku do kolejnych gałęzi nadrzędnych wywołania. Jeśli taki wykres znajdziemy – wykres ten jest domyślnym wykresem startowym – kończymy poszukiwanie.

Obsługa przycisków wykresu jest analogiczna jak w przypadku wydruków; mamy tu trzy akcje:

- Wykonaj domyślny wykres startowy (przycisk główny wykresów lub klawisz F12),
- Wybierz wykres startowy z menu kontekstowego i wykonaj go (dodatkowy przycisk wykresów z ikoną strzałki lub klawisz Alt-F12) oraz
- Edycja węzła wykresów (Ctrl + lewy klawisz myszy na przycisku głównym lub klawisz Ctrl-F12).

Po wykonaniu domyślnego lub wybranego wykresu startowego jest otwierane okno wynikowe analiz. Obiekt wykresu po aktywacji (otwarcie okna z wykresem) widzi tylko węzeł narysowanego wykresu a więc mimo, że do danego wywołania możemy mieć podłączoną dowolną ilość wykresów startowych – tylko jeden z nich jest 'bieżącym wykresem startowym'.

Wykresy startowe mogą być filtrowane przez zastosowanie dynamicznych warunków, które decydują czy w danej sytuacji (przy bieżących ustawieniach przez aplikację zmiennych występujących w warunku podłączenia wykresu) dany wykres startowy ma być widoczny w danym kontekście czy też nie.

3.1.3 Zagnieżdżanie wykresów (podwykresy)

Każdy wykres (w szczególności także wykresy startowe dla danego wywołania) może mieć dowolną ilość wykresów podrzędnych (podwykresów) podłączonych do niego w uporządkowany sposób; jeden z tych podwykresów jest traktowany jako domyślny (decyduje ustawienie flagi *Domyślny* podłączenia a w przypadku jej nie ustawienia dla

żadnego podwykresu - kolejność podwykresów wg lp; kolejność tą można zmieniać na drzewie hierarchii wykresów używając mechanizmu bąbelkowania).

Z drugiej strony każdy wykres może być podłączony do dowolnej ilości wykresów nadrzędnych w stosunku do niego.

Nie ma tu ograniczenia aby wykresy 'nie zapętlały' się tzn. aby dla żadnego łańcucha wykresów nie zachodziła sytuacja, że jakiś wykres występuje w tym łańcuchu więcej niż jeden raz. Takie podejście, poza prostotą implementacji ma tą dodatkową zaletę, że pozwala wykonać ten sam wykres jako swój podwykres, ale z innymi parametrami.

Nie przewidujemy na razie implementacji warunkowego zagnieżdżenia wykresów.

3.1.4 Graf wykresów – nawigacja w czasie wykonywania

Ponieważ każdy wykres może posiadać dowolną liczbę wykresów podrzędnych - mamy do czynienia z ukierunkowanym grafem i podczas wykonywania wykresów zachodzi konieczność umożliwienia nawigacji poprzez ten graf.

Przed wszystkim w każdym węźle-wykresie (poza 'liściami' nie zawierającymi już podwykresów) musimy umożliwić tzw. drill-down czyli eksplorację do węzłów (wykresów) podrzędnych – jest to realizowane jako akcja wykonywana przy kliknięciu w punkt wykresu. Jeśli klikamy prawym klawiszem – pojawia się menu kontekstowe zawierające pozycje odpowiadające wykresom podrzędnym. Natomiast lewy klik na punkcie wykresu powoduje wykonanie domyślnego podwykresu.

Ponieważ zbiór wykresów podłączonych do wywołania stanowi graf, po którym możemy się poruszać – istnieje potrzeba zapamiętywania przebytych ścieżek i ponownego poruszania się po nich (w ramach jednej 'sesji' z danym wywołaniem). Ciąg wykonań wykresów danego punktu wywołania, począwszy od bieżącego wykresu startowego nazywamy historią wykonań wykresów. Zapis wykonania wykresu do historii następuje przy wczytywaniu bieżącego wykresu startowego, a następnie przy wykonywaniu akcji *Eksploruj*. W odpowiednich zmiennych zapamiętywane są dane pozwalające odtworzyć wyświetlony uprzednio wykres w niezminionej postaci (z dokładnością do zmian wynikających z odświeżenia wyników); dane te to oczywiście zapytanie i wartości parametrów ostatnio użyte w tych zapytaniach oraz ostatnie wartości parametrów graficznych. Na każdym wykresie (poza bieżącym startowym dla opcji *Na początek*), na specjalnych przyciskach, są dostępne dodatkowo cztery opcje: *Na początek* (powrót do bieżącego wykresu startowego), *Wstecz* (powrót do poprzednio wykonanego wykresu), *Wprzód* (przejdźcie do następnego 'historycznie' wykresu) i *Na koniec* (przejdźcie do ostatniego 'historycznie' wykresu).

Opcja *Wstecz* (także *Na początek*) nie pojawia się w przypadku gdy jesteśmy na bieżącym wykresie startowym. Wszystkie parametry wykresu (z zapytania i graficzne), do którego się cofamy są odtwarzane na podstawie odpowiednich zmiennych ale dane wykresu są przeładowywane (jest to robione ze względów technicznych – nie przewidujemy cache-owania danych wynikowych wykresu).

Analogiczne uwagi dotyczą opcji *Wprzód*, którą dodatkowo trzeba odróżnić od *Eksploruj* także w tym sensie, że nawet jeśli *Wprzód* powoduje przejście do podrzędnego wykresu (tak jak *Eksploruj*) to po pierwsze nie konieczne jest to ten sam wykres podrzędny (jeśli jest ich więcej), a po drugie opcja *Eksploruj* dotyczy klikniętego w momencie jej wybrania punktu a opcja *Wprzód* bierze pod uwagę punkt poprzednio kliknięty, tj. kliknięty w trakcie wykonywania opcji *Eksploruj*, które wprowadziło dany wykres do historii.



Uwaga. Zapisujemy do kolejki historii tylko bieżący wykres startowy i eksploracje (każdą). Cofanie ani posuwanie się wprzód nie dopisuje do kolejki rekordów a tylko przesuwają pointer wskazujący bieżącą pozycję. Eksploracja usuwa wszystkie rekordy z kolejki historii będące 'po' rekordzie odpowiadającym eksplorowanemu wykresowi, dopisuje rekord docelowego podwykresu i ustawia na nim pointer. Powoduje to 'zapominanie' części historii po wykonaniu eksploracji ale jest to standardowe zachowanie (por. MSIE).

3.1.5 Edycja wykresów

Do wygodnego podłączania wykresów startowych do wywołań oraz wykresów do wykresów (zagnieżdżanie) zostało utworzone specjalne wielopoziomowe drzewo (korzeń, procedura, kontekst, wykresy startowe, wykresy podrzędne). Poza podstawowymi operacjami takimi jak dodawanie, podłączanie i odłączanie wykresów, ich edycja (w tym testowanie) i usuwanie dostępne będą także różne funkcje pomocnicze, np. włączanie/wyłączanie flagi domyślności, bąbelkowanie, szukanie wykresu zawierającego w definicji podany ciąg znaków i inne.

3.2 Źródła danych do wykresów

Każdy wykres może być tworzony w oparciu o jedno z niżej wymienionych źródeł:

- **Zapytanie SQL i dane w bazie**

Jest to podstawowy rodzaj źródła danych dla wykresów.

Przez zapytanie rozumiemy tu ciąg dowolnych1 instrukcji T-SQL, w którym ostatnią jest instrukcja SELECT o strukturze zgodnej ze strukturą specjalnej kolejki. Struktura ta, w przypadku gdy mamy do czynienia z prawdziwym wykresem, a nie tylko z tabelą wynikową, nakłada pewne ograniczenia na typy i zawartość pierwszych trzech kolumn zapytania.

1 Dokładniej: nie zmieniających struktury ani zawartości bazy danych – instrukcje takie są programowo zablokowane; ich umieszczenie w zapytaniu spowoduje sygnalizację błędu (nie dotyczy to tworzonych tabel tymczasowych).

Pierwsza kolumna numeruje wiersze wyniku zatem musi być całkowita a jej wartości unikalne, druga kolumna określa wartości na osi X - musi więc być rzeczywista lub całkowita a jej wartości unikalne, wreszcie trzecia kolumna określa opisy osi X – powinna być znakowa a jej wartości unikalne choć akurat te wymagania są tylko zaleceniami a nie formalnymi wymogami. W przypadku wykresów typu 'tylko tabela' powyższe ograniczenia nie obowiązują.

Wszystkie kolumny zwracane przez SELECT muszą być nazwane. Jeśli więc w liście tej występują agregaty lub stałe – należy koniecznie użyć aliasów (np. *SELECT PunktID=1, TrN_TrNID, TrN_NumerPelny, RazemNetto=SUM(TrE_WartoscNetto) FROM ...*).

Zapytanie jest wykonywane przez ADO.

W tekście zapytania w definicji wykresu mogą występować tzw. **parametry dynamiczne**, których wartości są wyliczane i wstawiane w miejsce definicji (odwołań do) tych parametrów przed przesłaniem wynikowego zapytania na serwer. Poza parametrami dynamicznymi w zapytaniu można umieszczać wyrażenia, zmienne tekstowe, wstawki tekstowe i wiele innych konstrukcji, które przed podaniem zapytania do serwera SQL są rozwijane (wyliczane i podstawiane wartościami).

Język użyty w zapytaniu wykresu jest więc połączeniem T-SQL i specjalnego języka obsługi parametrów dynamicznych, tzw. DPL, opisanego w osobnym dokumencie.

- **Kolejka globalna wypełniona przez aplikację**

Dane serii wykresu mogą także pochodzić ze specjalnych kolejek. Ma to miejsce w przypadkach, gdy aplikacja dysponuje już danymi (często uzyskanymi w inny sposób niż tylko wykonanie SELECT-ów).

- **Wywołanie procedury**

Wykonanie 'wykresu' może polegać nie tylko na uzyskaniu wyników i ich graficznej i/albo tabelarycznej prezentacji (wykresy normalne / wykresy typu 'tylko tabela'), ale także na wywołaniu procedury (tzw. drill-through).

W tym przypadku składnia 'zapytania' nie ma nic wspólnego z SQL-em – wywołanie procedury następuje zgodnie ze składnią języka DPL.

3.3 Parametry dynamiczne wykresów

Parametrem dynamicznym wykresu (w skrócie: parametrem) nazywamy zmienną występującą w zapytaniu lub parametrach graficznych (można np. sparametryzować tytuł wykresu), której wartość jest ustalana i wstawiana w miejsce definicji parametru w czasie wykonywania wykresu (tzw. runtime).

Ogólna składnia definicji parametru dynamicznego jest następująca:

?@[H]Picture[(_Q|_NQ)]|[Nazwa] Prompt:Default[?] (PD)

W przypadku, gdy chcemy użyć w parametrze bardziej wyrafinowanych opcji lub zadeklarować wszystkie parametry w jednym miejscu – używamy deklaracji parametru dynamicznego mającej postać:

@PAR Definicja Opcje PAR@ (PD-DEKL)

gdzie *Definicja* jest określona przez (PD).

Deklaracja różni się od definicji pod jeszcze jednym, istotnym względem. Mianowicie deklaracja przy rozwinięciu jest usuwana z tekstu. Definicja parametru jest natomiast zamieniana jego wartością.

Niezależnie od tego czy zadeklarujemy czy też zdefiniujemy parametr dynamiczny – najczęściej występuje potrzeba wielokrotnego odwoływania się jego wartości w innych częściach tekstu – korzystamy wówczas z odwołania do parametru dynamicznego (nazywanego także definicją powtórnego parametru):

??[(_Q|_NQ)]Nazwa (PD2)

Szczegółowy opis parametrów dynamicznych został umieszczony w osobnym dokumencie.

Początkowe wartości parametrów (tzn. te, które są użyte przy pierwszym wykonaniu wykresu w danej sesji wywołania) mogą być przekazywane do wykresu z poniższych źródeł i wg poniższej kolejności określającej priorytet źródła:

- Z danych pobranych z eksplorowanego punktu wykresu nadrzędnego – nie dotyczy to oczywiście bieżącego wykresu startowego.
- Z kolejki parametrów dynamicznych wykresu nadrzędnego (wiązanie po nazwie) – nie dotyczy to oczywiście bieżącego wykresu startowego

Jeśli np. w zapytaniu nadrzędnym i podrzędnym mamy parametr o nazwie *RaportyZamkniete* to eksplorując wykres z ustawioną wartością parametru dziedziczymy tą wartość na podwykresie.

- Parametry dynamiczne zdefiniowane w parametrach graficznych dziedziczą wartości z parametrów dynamicznych zdefiniowanych w zapytaniu bieżącego wykresu.
- Z kolejki zmiennych dynamicznych przekazanych do wykresu z aplikacji (dokładniej: z okna, z którego uruchamiamy analizę) – dotyczy to tylko bieżącego wykresu startowego.
- Z wartości domyślnych określonych w definicji parametru (także włączając opcjonalny odczyt ostatnio użytych wartości z rejestru).

Z takimi początkowymi wartościami parametrów tworzony jest wykres po raz pierwszy (w sesji danego wywołania). Przed tym jednak na oknie wykresu są tworzone kontrolki wszystkich parametrów a ich wartości ustalane są na wartości początkowe. Po narysowaniu wykresu możliwa jest edycja parametrów za pomocą wykreowanych kontrolki i odświeżenie wykresu z nowymi wartościami parametrów.

Na oknie wynikowym analiz znajduje się znacznik określający, czy pobrane dane są zgodne z bieżącymi ustawieniami kontrolek parametrów czy też nie (podobnie jak przycisk lejka w konstruktorze filtra).



Uwaga: Jeśli w parametrach graficznych definicji wykresu umieścimy parametr /@AutoRefresh to odświeżanie będzie następowało automatycznie (tj. bez wciskania przycisku odświeżenia) przy każdej zmianie któregoś z parametrów dynamicznych (zapytania lub graficznych).

Oto szczegóły mechanizmu, którego używa obiekt przy ustalaniu wartości parametrów wykresu.

Przed wykonaniem bieżącego wykresu startowego lub potomnego po raz pierwszy w danej sesji wywołania obiekt przeszukuje zapytanie i parametry graficzne wykresu w poszukiwaniu parametrów dynamicznych. Po znalezieniu wszystkich parametrów, przed wykonaniem wykresu obiekt stara się ustalić ich wartości:

- Jeśli mamy do czynienia z bieżącym wykresem startowym – są przeszukiwane parametry przekazane z aplikacji. Wiązanie odbywa się po *Nazwie* parametru. Dotyczy to tylko parametrów dynamicznych zdefiniowanych w zapytaniu.
- Jeśli wykonujemy opcję Eksploruj - najpierw sprawdzane jest zapytanie wykresu nadrzędnego (dokładniej kolekcja Fields jego wynikowego recordsetu) w poszukiwaniu kolumn o nazwie zgodnej z *Nazwą* parametru. Jeśli taka kolumna zostanie znaleziona – jest pobierany kliknięty rekord z kolejki z danymi nadrzędnego wykresu a następnie w miejsce parametru wstawiana jest wartość z odpowiedniej kolumny tego rekordu. Dotyczy to zarówno parametrów zapytania jak i graficznych. Jeśli szukana kolumna nie zostanie znaleziona - jest przeszukiwana kolejka parametrów dynamicznych nadwykresu (czyli eksplorowanego wykresu). Jeśli w kolejce tej zostanie znaleziony parametr o nazwie *Nazwa* – następuje odziedziczenie wartości. Oczywiście dla parametrów zdefiniowanych w zapytaniu przeszukiwana jest kolejka parametrów dynamicznych 'nadzapytania' a dla parametrów zdefiniowanych w parametrach graficznych – kolejka parametrów dynamicznych 'nadparametrów' graficznych.
- Jeśli mamy do czynienia z parametrem dynamicznym zdefiniowanym w parametrach graficznych, niezależnie od wyników poprzednich poszukiwań, następuje próba odziedziczenia wartości tego parametru z parametru dynamicznego o takiej samej nazwie, zdefiniowanego w zapytaniu bieżącego wykresu. Mamy tu zatem do czynienia z bezwarunkowym dziedziczeniem wartości graficznych parametrów dynamicznych z wartości zapytaniowych parametrów dynamicznych bieżącego wykresu.
- W obu przypadkach jeśli poszukiwania nie dadzą rezultatu – wartość parametru jest ustalana na podstawie jego *Defaultu* (wartości odczytanej z definicji parametru)

Wszystkie parametry danego wykresu, łącznie z ustalonymi wartościami zapisywane są w odpowiedniej kolejce. Na podstawie tej kolejki kreowane są dynamicznie kontrolki parametrów i od razu rysowany wykres.

Przed wykonaniem wykresu, który już wcześniej został wczytany (*Na początek, Wstecz, Wprzód Na koniec, Odśwież*) następuje wykreowanie kontrolek parametrów tego wykresu z wartościami ostatnio użytymi i narysowanie wykresu w oparciu o te wartości. Teraz Użytkownik może manipulować wartościami parametrów poprzez kontrolki – wartości te są zapisywane do kolejki w momencie wykonania akcji *Odśwież*.

3.4 Zmienne dynamiczne przekazywane z aplikacji do bieżącego wykresu startowego

Zmienne dynamiczne mają za zadanie zainicjalizować wybrane parametry dynamiczne wykresu wartościami przekazanymi z aplikacji. Wartości tych zmiennych są ustalane w czasie wykonywania programu i przesyłane do bieżącego wykresu startowego. Można np. przekazać z procedury-listy zmienną zawierającą aktualnie zastosowany filtr na liście.



Uwaga: Zmienne dynamiczne są podstawiane pod parametry dynamiczne tylko dla bieżącego wykresu startowego (nie dotyczą podwykresów – tam występuje dziedziczenie wartości parametrów o zgodnej nazwie).

3.5 Wyrażenia

Wyrażenia to ciągi znaków występujące w zapytaniach oraz w parametrach graficznych wykresu takie, których wartości są obliczane przed wykonaniem wykresu. W wyrażeniach mogą występować m.in. odwołania do parametrów dynamicznych a także zmienne i funkcje systemowe. Wyrażeń używa bardzo często do określania wartości początkowych parametrów dynamicznych (np. bieżąca data).

Składnia wyrażenia jest następująca:

{ciąg_znaków}

(W)

Szczegółowy opis wyrażeń został umieszczony w osobnym dokumencie.

3.6 Predefiniowane wstawki tekstowe

3.6.1 Wstawki tekstowe

Potrzeba wprowadzenia wstawek tekstowych wynika z trudności, jakie występowały przy utrzymywaniu standaryzacji dużej liczby wykresów oraz potrzeby wyodrębnienia części wspólnych wielu analiz (likwidacja redundancji).

Co do standaryzacji chodzi o to aby np. powtarzające się na wielu wykresach analogiczne kontrolki (lub grupy kontrolek, np. zakres dat) wyglądały na każdym wykresie tak samo ('dymek', opcje graficzne, warunek poprawności, komunikat itp.). Do tej pory zmiana jakiegokolwiek własności takiej kontrolki wiązała się z koniecznością wykonania tej zmiany we wszystkich odpowiednich wykresach. Stosując predefiniowane wstawki tekstowe – robimy zmianę w jednym miejscu. Podobnie ma się rzecz, jeśli chodzi o parametry graficzne, które, poza tytułem, typem wykresu i wyborem serii powinny być w zasadzie identyczne dla wszystkich wykresów.

Wyodrębnienie wspólnych części wykresów polega na tym, że jeżeli mamy kilka wykresów o bardzo podobnej konstrukcji (różniących się tylko detalami) to za pomocą predefiniowanych wstawek tekstowych można jak gdyby stworzyć jedną, sparametryzowaną definicję wykresu. Oczywiście, formalnie rzecz biorąc, definicji będzie kilka, tyle tylko, że będą one posiadały jedynie wywołanie odpowiedniej, wspólnej wstawki z różnymi parametrami. Tyle ogólnych impresji. Przejdźmy do konkretów.

Pierwszą czynnością, jaką wykonuje moduł wykresów po wczytaniu definicji wykresu z bazy, jeszcze przed podaniem tekstów (zapytanie i grafika) do preparsingu, jest poszukiwanie w tych tekstach deklaracji predefiniowanych wstawek tekstowych (PWT) i ich rozwijanie.

Deklaracja PWT w tekście zapytania bądź grafiki ma następującą składnię:

@INS *Nazwa* | [*Parametry*] [*ParametryX*] INS@ (PWT0)

gdzie *Nazwa* jest unikalną (w ramach bazy wykresu) nazwą wstawki (NO CASE) a opcjonalne człony *Parametry* i *ParametryX* posiadają następującą składnię:

***Parametr*[, *Parametr*...]** (PWTP)

**^*NazwaParametruX=WartoscParametruX*
[^*NazwaParametruX=WartoscParametruX*,...]** (PWTPX)

gdzie *Parametr* jest dowolnym ciągiem znaków (NO CASE), nie zawierającym przecinka (separator parametrów), znaku ^ (definicja parametru typu X) ani ciągu znaków `INS@` (spacja, INS@ - terminator PWT). Parametry typu X muszą występować po parametrach zwykłych i nie mogą być z nimi wymieszane. Zostaną one omówione w dalszej części tego rozdziału.

Parametry są przekazywane do definicji wstawki i decydują o wstawianiu do niej tekstów warunkowych (patrz następny rozdział) i wstawek podrzędnych (tzw. podwstawek).

Definicja wstawki jest traktowana jak normalny tekst – wynika stąd, że wstawki mogą być zagnieżdżone. Deklaracja (pod)wstawki we wstawce ma postać:

@INS *Nazwa* | [*Warunek*] | [*Parametry*] INS@ (PPWT)

gdzie *Nazwa* jest unikalną (w ramach bazy wykresu) nazwą wstawki podrzędnej (NO CASE) a opcjonalny człon *Warunek* posiada następującą składnię:

***Segment*[, *Segment*...]** (PWTW)

gdzie *Segment* jest dowolnym ciągiem znaków (NO CASE), nie zawierającym przecinka (separator segmentów) ani ciągu znaków

`INS@` (spacja, INS@ - terminator PWT). Ponieważ podwstawka może zostać wywołana z parametrami (deklaracja w tekście nadrzędnym) – sprawdzamy czy parametry te spełniają *Warunek* wstawienia podwstawki. Jeśli *Warunek* jest spełniony - następuje zamiana deklaracji podwstawki na jej rozwinięcie, jeśli nie – usunięcie deklaracji podwstawki. Walidacja warunku odbywa się w następujący sposób:

- Jeśli *Warunek* jest pusty – uznaje się, że jest spełniony
- Jeśli *Warunek* nie jest pusty – uznaje się, że jest spełniony, jeśli dla co najmniej jednego z jego członów (*Segment*) istnieje odpowiednik (równość NO CASE) wśród przekazanych do podwstawki parametrów zwykłych (nie typu X).



Uwaga. Walidacja dotyczy parametrów przekazanych z nadrzędnego tekstu. Parametry określone w deklaracji podwstawki są jedynie przekazywane do jej rozwinięcia w przypadku, gdy to rozwinięcie jest wstawiane.



Uwaga. Warto zwrócić uwagę na fakt, że przed sekwencją kończącą INS@ musi występować co najmniej jedna spacja, zatem deklaracja:

@INS Ala

INS@

nie jest poprawna gdyż bezpośrednio przed końcowym INS@ występuje ENTER zamiast spacji.

Przy rozwijaniu wstawki jej rozwinięcie pobierane jest z tej bazy (wykresów standardowych lub użytkownika), z której pobrano definicję wykresu. W przypadku nie znalezienia wstawki o podanej nazwie, jeśli mamy do czynienia z wykresem użytkownika, wstawka jest następnie poszukiwana w bazie wykresów standardowych.

3.6.2 Teksty warunkowe

Omawiając parametry przekazywane w deklaracji wstawki wspomnieliśmy o tzw. tekstach warunkowych. Tekst warunkowy jest dowolnym ciągiem znaków, który może zostać wstawiony do definicji wstawki przy spełnieniu określonych warunków.

Tekst warunkowy może występować jedynie w definicji predefiniowanej wstawki tekstowej, a jego składnia jest następująca:

@IF(*Warunek*) *Tekst* IF@ (TW)

gdzie *Warunek* jest analogiczny jak w definicji (PPWT) a *Tekst* jest dowolnym ciągiem znaków nie zawierającym sekwencji zakończenia `IF@` (spacja, IF@). Ponieważ TW może występować jedynie w definicji PWT – są określone parametry przekazane do definicji PWT z jej deklaracji w tekście nadrzędnym. Parametry te pozwalają na walidację warunku *Warunek* (mechanizm dokładnie taki sam jak w przypadku PPWT).

Uzupełniającym elementem składni tekstu warunkowego jest zaprzeczony tekst warunkowy:

@IFNOT(*Warunek*) *Tekst* IFNOT@ (TWNOT)

różni się on od TW tylko tym, że tekst jest wstawiany jeśli warunek jest pusty albo żaden jego człon nie ma odpowiednika wśród przekazanych parametrów.

Uwaga.

Warto zwrócić uwagę na fakt, że przed sekwencją kończącą IF@ (IFNOT@) musi występować co najmniej jedna spacja, zatem deklaracja:

@IF(FAPA)

xxx

IF@

nie jest poprawna gdyż bezpośrednio przed końcowym IF@ występuje ENTER zamiast spacji.

Analogicznie do powyższych zostały wprowadzone teksty warunkowe:

@IFAND(*Warunek*) *Tekst* IFAND@ (TWAND)

oraz

@IFANDNOT(*Warunek*) *Tekst* IFANDNOT@ (TWANDNOT)

różniące się od opisanych jedynie sposobem sprawdzania *Warunku*, który w tym przypadku uważa się za spełniony tylko wtedy gdy jest pusty lub dla wszystkich jego członów (*Segment*) istnieje odpowiednik (równość NO CASE) wśród przekazanych do podstawki parametrów zwykłych (nie typu X).

3.6.3 Rozszerzone parametry tekstowe (parametry typu X)

Opisany wyżej mechanizm nie pozwala na parametryzowanie wstawek tekstowych. Przypuśćmy, że chcemy zdefiniować wstawkę wstawiającą zakres dat. Wstawki tej chcemy użyć zarówno w analizie a1 gdzie wartości domyślne dat mają być odpowiednio Today() i Today()+1 i w analizie a2 z wartościami domyślnymi Date(Month(Today()), 1, Year(Today())) i Today(). Aby wstawka mogła zrealizować oba te zadania – musi obsługiwać parametry rozszerzone.

Składnia rozszerzonego parametru tekstowego wstawianego do wstawki jest następująca:

@INSP(*Nazwa*) (INSP)

gdzie *Nazwa* jest nazwą parametru przekazanego z tekstu nadrzędnego (zapytania, grafiki bądź podstawki).

Składnia przekazania w/w parametru tekstowego jest określona przez (PWTPX). We wzorze tym *NazwaParametruX* odpowiada *Nazwie* w (INSP) (równość jest typu NO CASE). Wartość parametru może być ustalona ale może też zawierać element @INSP().

3.7 Parametry graficzne wykresu

Aby nie zmieniać struktury tabeli z wykresami przy dodawaniu w przyszłości obsługi kolejnych parametrów graficznych wprowadzamy jedną, parsowaną kolumnę stringową *ParametryGraficzne*. Kolumna ta powinna zawierać ustawienia wszystkich potrzebnych parametrów graficznych danego wykresu w postaci:

/@*SymbolGr*[=*WartoscGr*(*BezApostrofów*)] (PG)

gdzie *SymbolGr* jest nazwą parametru graficznego a *WartoscGr* jego wartością początkową. Kolumna *ParametryGraficzne* jest parsowana przy wczytywaniu definicji wykresu i składana przy zapisie.

Opcjonalna wartość *WartoscGr* jest wartością parametru, z którą wykres jest rysowany pierwszy raz; wartość ta może zostać zmieniona w trakcie oglądania wykresu przez edycję odpowiedniej kontrolki dynamicznej, wykreowanej na podstawie parametru dynamicznego wykresu dotyczącego wartości danego parametru graficznego. Wartość *WartoscGr* może bowiem zawierać parametry dynamiczne.

W wartościach parametrów graficznych obu typów mogą występować wyrażenia.

Oto najważniejsze symbole dostępnych parametrów graficznych:

<i>SymbolGr</i>	Znaczenie	Wartości	D	PD	D(*)	PD(*)
HeaderName	Tytuł wykresu			1		1
XAxisName	Tytuł osi X			1		1
YAxisName	Tytuł osi Y			1		1
Type	Typ wykresu	1 – kolumnowy, 2 – liniowy, 4 – kołowy,...	1	1	1	1
Depth	Głębokość 3D	0 – wykres 2D, >0 – wykres 3D	0	1	100	1
LineWidth	Grubość linii	1-10 (dotyczy typu 2)	2	1		0
PointWidth	Grubość słupka	1-10 (dotyczy typu 1)		0		0
ShowDataLabels	Wyświetla wartości punktów	0 – nie pokazuje, 1 – pokazuje	1	1	1	1
Series	Definicje serii	<i>Kol</i> ,..., <i>Kol</i> gdzie <i>Kol</i> ,..., <i>Kol</i> - numery kolumn zapytania ($4 \leq Kol \leq 24$).	4	1	4	1
NoHeadOnCols	Wykaz kolumn, które mają nie mieć headera	<i>Kol</i> ,..., <i>Kol</i> gdzie <i>Kol</i> ,..., <i>Kol</i> - numery kolumn zapytania ($1 \leq Kol \leq 24$).		0		0
HideCols	Wykaz kolumn, które mają być ukryte (zerowej szerokości)	<i>Kol</i> ,..., <i>Kol</i> gdzie <i>Kol</i> ,..., <i>Kol</i> - numery kolumn zapytania ($1 \leq Kol \leq 24$).		0		0
AutoRefresh	Automatyczne odświeżanie po zmianie któregośkolwiek parametru	0 – nie, 1 – tak	0	1	1	1
SkipFirstExec	Nie wykonywanie akcji (zapytania, procedury) za pierwszym razem, dopiero po odświeżeniu. Dotyczy tylko wykresu startowego.	0 – wykonuj akcję także przy pierwszym wykonaniu wykresu, 1 – nie wykonuj akcji przy pierwszym wykonaniu wykresu (np. nie chcemy wykonywać zapytania z domyślnymi wartościami parametrów)	0	1	1	1
NoPunktIDCol	Zapytanie nie zawiera kolumny z licznikiem wierszy	0 – zawiera kolumnę z ID wiersza, 1- nie zawiera kolumny z ID wiersza, tworzona jest techniczna kolumna z ID wiersza niewidoczna dla użytkownika.	0		1	

Default jest wartością, która jest podstawiana, gdy w parametrach graficznych dany parametr w ogóle nie występuje i *PodstawDefault=1*. Gdy w parametrach graficznych dany parametr w ogóle nie występuje i *PodstawDefault=0* – to wartość parametru nie jest w ogóle podstawiana.

Default()* jest wartością, która jest podstawiana, gdy w parametrach graficznych dany parametr występuje ale nie ma podstawionej wartości (np. /@3D, /@PokazujEtykietyDanych) i *PodstawDefault(*)=1*. Gdy w parametrach graficznych dany parametr występuje ale nie ma podstawionej wartości i *PodstawDefault(*)=0* – to wartość parametru nie jest w ogóle podstawiana.

Uwaga. Parametr *NoHeadOnCols* mimo, że formalnie przynależy do parametrów graficznych dotyczy tabeli z wynikami a nie wykresu. Stąd nie należy się dziwić, że wykres typu 'tylko tabela' ma aktywne pole z parametrami graficznymi.

Kompletny wykaz obsługiwanych parametrów i sposób ich ustawiania opisuje osobny dokument.

Graficzne zmienne symboliczne.

Graficzne zmienne symboliczne to zmienne postaci *@NAZWAZMIENNEJ*, występujące w parametrach graficznych wykresu, których wartości podstawiane są do parametrów graficznych przed narysowaniem wykresu ale już po załadowaniu wyników. Poniższa tabela zawiera wszystkie obsługiwane w tej chwili zmienne graficzne:

Nazwa zmiennej	Opis (wartość)
RECORDS	Zwraca ilość rekordów wyniku zapytania. Przykład użycia: /@ShowXLabelsEvery={@Records/3}



Uwaga: Nazwy zmiennych graficznych są CASE SENSITIVE.

3.8 Wywoływanie funkcji

Akcja wykonywana podczas eksploracji wykresu (lub wykonywania wykresu startowego) nie zawsze musi polegać na narysowaniu wykresu czy pobraniu danych w postaci tabelarycznej – zamiast tego może także być wywołana funkcja (procedura). Najczęstsze zastosowanie polega na wywołaniu procedury okna z odfiltrowaną listą zapisów składających się na wartość klikniętego punktu wykresu - mamy wówczas do czynienia z tzw. drill-through. Składnia wywoływania funkcji jest zgodna z językiem DPL i została opisana w oddzielnym dokumencie.

3.9 Wykresy z kolejki

Czasami zdarza się, że na oknie mamy dane liczone w specyficzny sposób, zaszyty w kodzie programu. Dane te trudno uzyskać wykonując zapytanie SQL, a chcielibyśmy je umieścić na wykresie. W tym celu zaimplementowano możliwość ustanowienia jako źródła danych dla wykresu specjalnych kolejek wypełnianych przez aplikację.

Wybranie w definicji wykresu typu akcji 'Kolejka' powoduje, że pole *Zapytanie* staje się niedostępne – to aplikacja, wypełniając w/w kolejki, decyduje o wartościach danych i o typach kolumn – Użytkownik na poziomie definicji wykresu nie ma na to wpływu.

3.10 Wykresy bez wykresów - tabele z wynikami

Moduł analiz posiada możliwość pobierania danych bez rysowania wykresu. W takim przypadku wyniki nie są prezentowane graficznie a tylko w postaci tabelarycznej. Oczywiście tabela danych wynikowych, podobnie jak wykres, jest aktywna. Oznacza to, że można eksplorować jej rekordy wykonując analizy podrzędne.

Wykres typu 'tylko tabela' nie posiada rzecz jasna żadnych parametrów graficznych.



Uwaga: Wykres typu 'tylko tabela' nie zawiera ograniczeń jakie narzucamy dla zapytań normalnych wykresów – zapytania mogą tu zwracać 1-20 kolumn zupełnie dowolnych typów i o dowolnym znaczeniu.

3.11 Przekazywanie do wykresów zaznaczonych pozycji

Wykres może odwoływać się do pozycji zaznaczonych na liście. Oczywiście aby takie odwołanie było możliwe – okno aplikacji musi nie tylko posiadać, ale także eksportować zaznaczenia do modułu wykresów (innymi słowy programista danego okna musiał przewidzieć, że Użytkownik będzie chciał w tym miejscu odwołać się w definicji wykresu do zaznaczeń na liście). Jeśli w/w jest spełniony, zbudowanie wykresu opartego o zaznaczenia polega na:

- We właściwościach ogólnych wykresu:
Ustawieniu na wykresie flagi **Wykres oparty o tabelę zaznaczeń**
- W zapytaniu wykresu:
 - Zadeklarowaniu ukrytego parametru dynamicznego **@PAR ?@Hs50 | ZazGUID | ZazGUID:" PAR@**
 - Wykonaniu **JOIN-a do tabeli WkrZaznaczenia2** (np. ... JOIN WkrZaznaczenia2 ON BNa_BNaID = WkZ2_GIDNumer ...).
 - Umieszczeniu w filtrze zapytania warunku **WkZ2_GUID=??_QZazGUID** na odziedziczony ze zmiennej dynamicznej GUID

3.12 Wysłanie wyników wykresu

Wyniki wykonania analizy (wykres w postaci graficznej oraz tabela z wynikami) są zawsze (o ile nie jest to wynik skalarny) zapisywane do pliku. Plik ten (tzw. plik wynikowy) może mieć jeden z następujących **formatów**:

- COMMA (dane tabelaryczne)
- Excel (dane tabelaryczne)
- XML (dane tabelaryczne)
Jest tu możliwość zastosowania podanego szablonu XSL.
- WMF (grafika)

W przypadku formatu COMMA istnieje możliwość dopisywania danych do istniejącego pliku.

W/w plik jest zapisywany z podaną nazwą (lub systemową nazwą tymczasową, jeśli nie podano nazwy docelowej) i w podanym katalogu (lub systemowym katalogu tymczasowym, jeśli nie podano katalogu docelowego).

Tak zapisany plik wynikowy może być następnie przekazany do podanego programu (.COM, .EXE, .BAT). W tym przypadku należy podać specyfikację odpowiedniego pliku wykonywalnego.



Uwaga: Wskazany program jest uruchamiany poleceniem RUN Clariona, dlatego nie można w ten sposób uruchamiać skryptów. Rozwiązaniem problemu jest zastosowanie przesiadkowego batch-a wykonującego odpowiedni skrypt (batch może posiadać parametr %1, w którym zostanie podana specyfikacja pliku z wynikami).

Silnik wykresów uruchamia polecenie wpisane, jako „Przełącz do programu” otaczając je cudzysłowami i dodając po spacji specyfikację pliku wynikowego otoczoną cudzysłowami.

Może to więc wyglądać następująco:

„notepad.exe” „d:\x\wynik.txt”

Mechanizm doklejania cudzysłowów nie pozwala na użycie jako „Przełącz do programu” ciągu

cmd /c start plik.pdf

ani

cmd /c start wynik_raportu.txt

bo nastąpiłaby próba wykonania komendy

„cmd /c start plik.pdf” „wynik_raportu.txt”

lub

„cmd /c start” „wynik_raportu.txt”

co jest niepoprawne.

Niepoprawne jest też użycie jako „Przełącz do programu” ciągu

plik.pdf

gdyż Clarion próbuje uruchomić w/w polecenie RUN-em, więc ciąg musi odpowiadać plikowi wykonywalnemu.

Rozwiązanie:

- Jeśli chcemy uruchamiać plik niezależny od raportu, należy utworzyć bezparametrowego batch-a startującego ten plik komendą START i w „Przełącz do programu” wpisać tego batcha (np. d:\x\StartSR1.bat).

Batch może mieć poniższą postać:

START plik.pdf

- Jeśli chcemy uruchamiać plik wynikowy raportu, należy utworzyć jedno-parametrowego batch-a startującego ten plik (bez komendy START) i w „Przełącz do programu” wpisać tego batcha (np. d:\x\StartSR2.bat).

Batch może mieć poniższą postać:

%1

Niestety w/w składnia powoduje otwarcie dodatkowego okna DOS-owego (nie można tu użyć *START %1* ani *CMD /C START %1*).

Jeśli zamiast jako program podamy specyfikację pliku .DLL – wówczas nie plik wynikowy, ale kolejka z wynikami zostanie przekazana do eksportowanej funkcji podanego .DLL-a. W tym przypadku składnia jest następująca:

Specyfikacja_Pliku_DLL[@Nazwa_Funkcji]

W przypadku żądania przekazania pliku wynikowego do programu i nie podania pliku programu następuje uruchomienie pliku wynikowego zgodnie z jego rozszerzeniem (wykonanie ShellExecute z akcją open na pliku wynikowym).

Jeśli wykres produkuje wynik skalarny, następuje uruchomienie pliku programu z parametrami *?NazwaKolumny1= WartośćKolumny1...?NazwaKolumnyN= WartośćKolumnyN*.

Plik wynikowy może też być przekazany jako załącznik w e-mailu zaadresowanym do podanych (oddzielonych średnikami) adresatów.

4 Interface Użytkownika

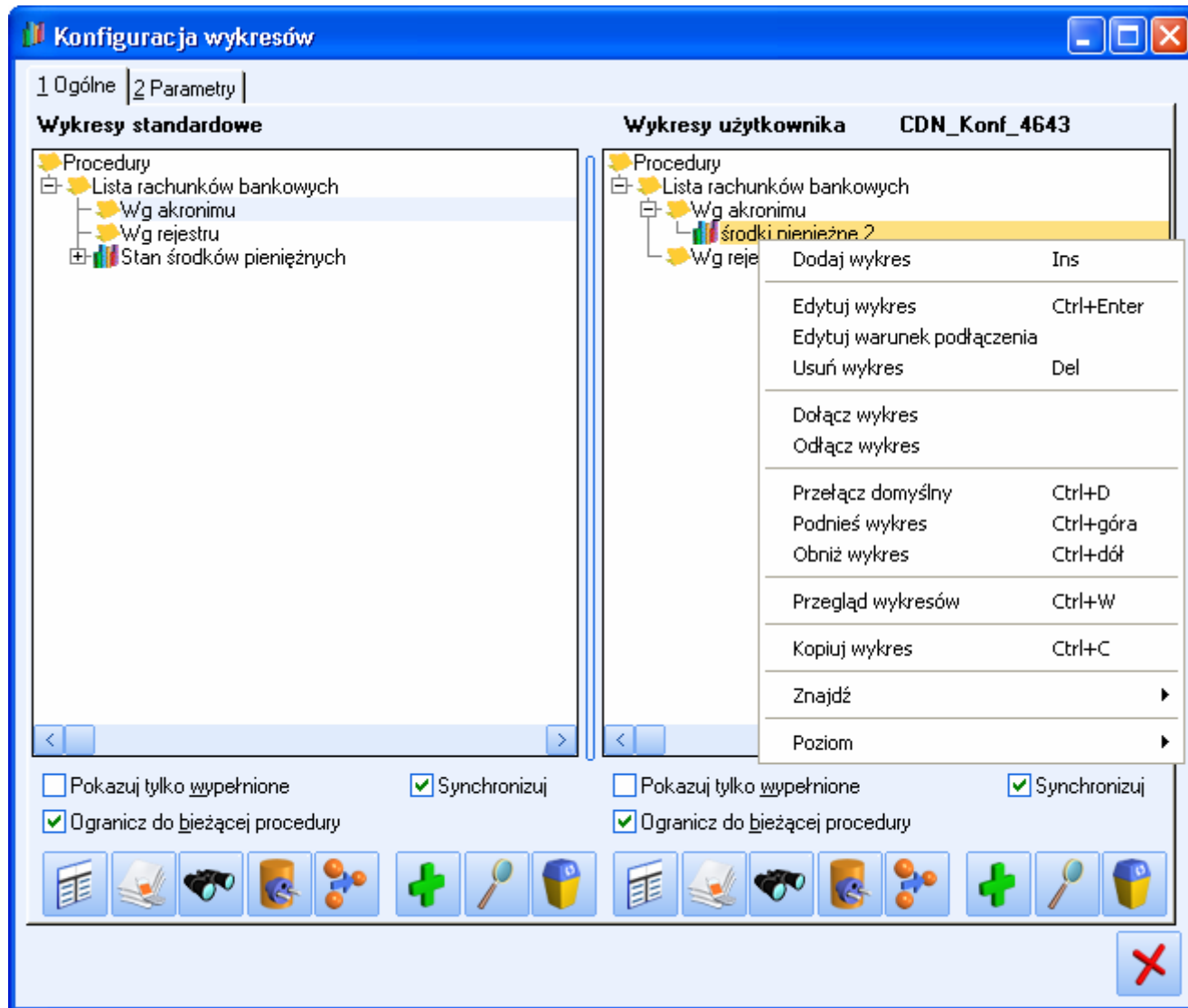
4.1 Edycja wykresów

4.1.1 Drzewo hierarchii wykresów

Podstawową procedurą edycji wykresów jest **drzewo hierarchii wykresów**. Procedura ta umożliwia wykonywanie wszystkich operacji edycyjnych dla wykresów standardowych i użytkownika:

- oglądanie struktury połączeń wykresów (do wywołań oraz wykresów do siebie nawzajem),
- podłączanie i odłączanie wykresów (w tym edycja warunków podłączenia wykresów startowych),
- określanie domyślności i przestawianie kolejności wykresów (bąbelkowanie)
- dodawanie, edycję, testowanie i usuwanie wykresów (można usuwać tylko wykresy nie posiadające podwykresów),
- szukanie podanego ciągu znaków w nazwie wykresu, kontekstu lub procedury,
- proste filtrowanie drzewa,
- aktualizację pliku z wydrukami standardowymi,
- oglądanie zmiennych dynamicznych wysyłanych z aplikacji do wykresu w danym momencie.

Należy zwrócić uwagę, że edycja wykresów standardowych przez zwykłego (tj. nie pracującego w specjalnym trybie super-user) Użytkownika ma bardzo ograniczony charakter. Użytkownik nie może zmienić ani logicznej struktury wykresów standardowych (podłączanie/odłączanie) ani żadnych parametrów identyfikujących te wykresy (np. nazwa), wpływających na wyniki (np. zapytanie) czy też wygląd (parametry graficzne) wykresów standardowych. Wykresy standardowe nie mogą być także dodawane ani usuwane przez Użytkownika. Jedyne możliwe w tym przypadku modyfikacje dotyczą opcji wysyłania wyniku, ustalania domyślności wykresów i warunków ich podłączenia. Zmiany te jednak nie są zapisywane do pliku z wykresami standardowymi (który można traktować jako READ/ONLY), a do specjalnych tabel SQL-owych, będących podziorami funkcjonalnymi odpowiednich pełnych tabel (patrz rozdział poświęcony bazie danych). W tabelach tych są zapisywane modyfikacje ustawień wykresów standardowych dla poszczególnych Użytkowników.



Rys. 1 Drzewo hierarchii wykresów – zakładka Ogólne

Wszystkie akcje dostępne dla podświetlonego obiektu (gałąź na drzewie) są możliwe do wykonania z poziomu podręcznego otwieranego przez kliknięcie prawym klawiszem myszy na obiekcie. W menu są również podane skróty klawiszowe do poszczególnych akcji. Dla podstawowych i najczęściej wykonywanych akcji pod listami umieszczono dodatkowo przyciski.

Poniżej krótki opis znaczenia poszczególnych kontroltek.



- otwiera płaską listę wykresów, opisaną w rozdziale 3.1.3. Na liście tej można m.in. znaleźć wykresy nie podłączone do żadnego wywołania a więc niewidoczne na drzewie hierarchii (i nie możliwe do wykonania z poziomu programu).



- otwiera listę wstawek tekstowych, opisaną w rozdziale 3.1.4.



- wyszukiwanie podanego ciągu znaków w nazwach procedur, kontekstów i wykresów.



- wybór wykresu z listy i podłączenie go do podświetlonej gałęzi.



- odłączenie podświetlonego wykresu.



- dodanie nowego wykresu i podłączenie go do podświetlonej gałęzi. Kliknięcie przycisku z Ctrl lub z Shift powoduje zainicjalizowanie dodawanego wykresu danymi podświetlonego wykresu (wzorca), z tym, że użycie Shift wymusza podłączenie nowego wykresu na tym samym poziomie co wzorec czyli do ojca wzorca (a nie do wzorca).



- edycja wykresu (otwarcie formularza definicji wykresu).



- usunięcie wykresu. Usuwany wykres nie może mieć podłączonych podwykresów.

Pokaż tylko wypełnione - odfiltrowuje (lub nie) gałęzie procedur i kontekstów nie wypełnione, tj. takie, do których nie są podłączone żadne wykresy.

Ogranicz do bieżącej procedury - odfiltrowuje (lub nie) gałęzie procedur innych niż aktualna (tj. ta, z której zostało wywołane drzewo konfiguracji wykresów).

Synchronizuj - pozycjonuje belkę danego panelu zgodnie z ustawieniem belki na sąsiednim panelu.

Wybrane pozycje menu:

Edytuj warunek podłączenia – akcja dostępna tylko dla wykresów startowych. Pozwala ustalić warunek jaki musi być spełniony aby dany wykres był możliwy do wykonania (wyboru) w danym punkcie. Warunek musi mieć postać wyrażenia, w którym mogą występować funkcje systemowe i funkcje a także zmienne zabindowane przez aplikację.

Przełącz domyślny – ustawia dany wykres jako domyślny w danym punkcie (z dokładnością do bazy wykresu) lub usuwa takie ustawienie.

Kopiuj/Wklej wykres – kopiuje/wkleja definicję wykresu (akcja pomocna w kopiowaniu definicji pomiędzy panelami, głównie przy kopiowaniu wykresów standardowych na drzewo wykresów użytkownika).

Poziom procedur/kontekstów – (roz)zwija gałęzie drzewa do wybranego poziomu.

Druga zakładka dotyczy przekazanych z aplikacji zmiennych dynamicznych.

Na liście mamy nazwy i aktualne wartości zmiennych, które byłyby w tym momencie przekazane do wykresu.

Nazwa	Typ	Wartość
ZMIENNE DYNAMICZNE PRZEKAZYWANE Z APLIKACJI DO WYKRESU		
STERUJĄCE		
Filtr		
FiltrApp		(BRa_Nieaktywny=0 AND 1=1)
GLOBALNE		
LOKALNE (ZALEŻNE OD PROCEDURY)		
Predefiniowane		
Dedykowane		
PokazujTezNieAktywne	@s255	0
OperatorID	@s255	1

Rys. 2 Drzewo hierarchii wykresów – zakładka Parametry



- przelicza wartości parametrów.

4.1.2 Formularz wykresu

Do edycji wykresu służy **formularz definicji wykresu**. Składa się on z trzech zakładek, które kolejno omówimy.

Zakładka **Ogólne**

The screenshot shows a Windows-style window titled "Definicja wykresu - zostanie zmieniona". It has three tabs: "1 Ogólne", "2 Zapytanie", and "3 Opcje wysyłania wyników". The "Ogólne" tab is active. At the top right, there is a field for "ID źródła / wykresu:" with the value "0/8".


The form contains the following sections:

- Nazwa:** A text box containing "Stan środków pieniężnych".
- Typ akcji:** Three radio buttons: "Zapytanie SQL" (selected), "Kolejka", and "Wywołanie procedury".
- Prezentacja wyników:** Two radio buttons: "Wykres" (selected) and "Tylko tabela". To the right of "Wykres" is a checkbox "XY" which is unchecked. To the right of "Tylko tabela" is a checkbox "Wynik skalarny" which is unchecked.
- Dostępność wykresu:** Two radio buttons: "Globalny" (selected) and "Operator". To the right of "Operator" is a dropdown menu.
- Oczekiwanie na wykonanie zapytania:** Three radio buttons: "Jak w konfiguracji" (selected), "Nieskończony czas", and "Ustalony czas (s):". To the right of "Ustalony czas (s):" is a spin box with the value "0".
- Opis:** A large text area with a vertical scrollbar.
- Autor:** A text box.
- Nr wersji:** A text box.
- Utworzono:** A text box showing "-- 0:00".
- Ostatnia modyfikacja:** A text box showing "2004-09-20 15:32".

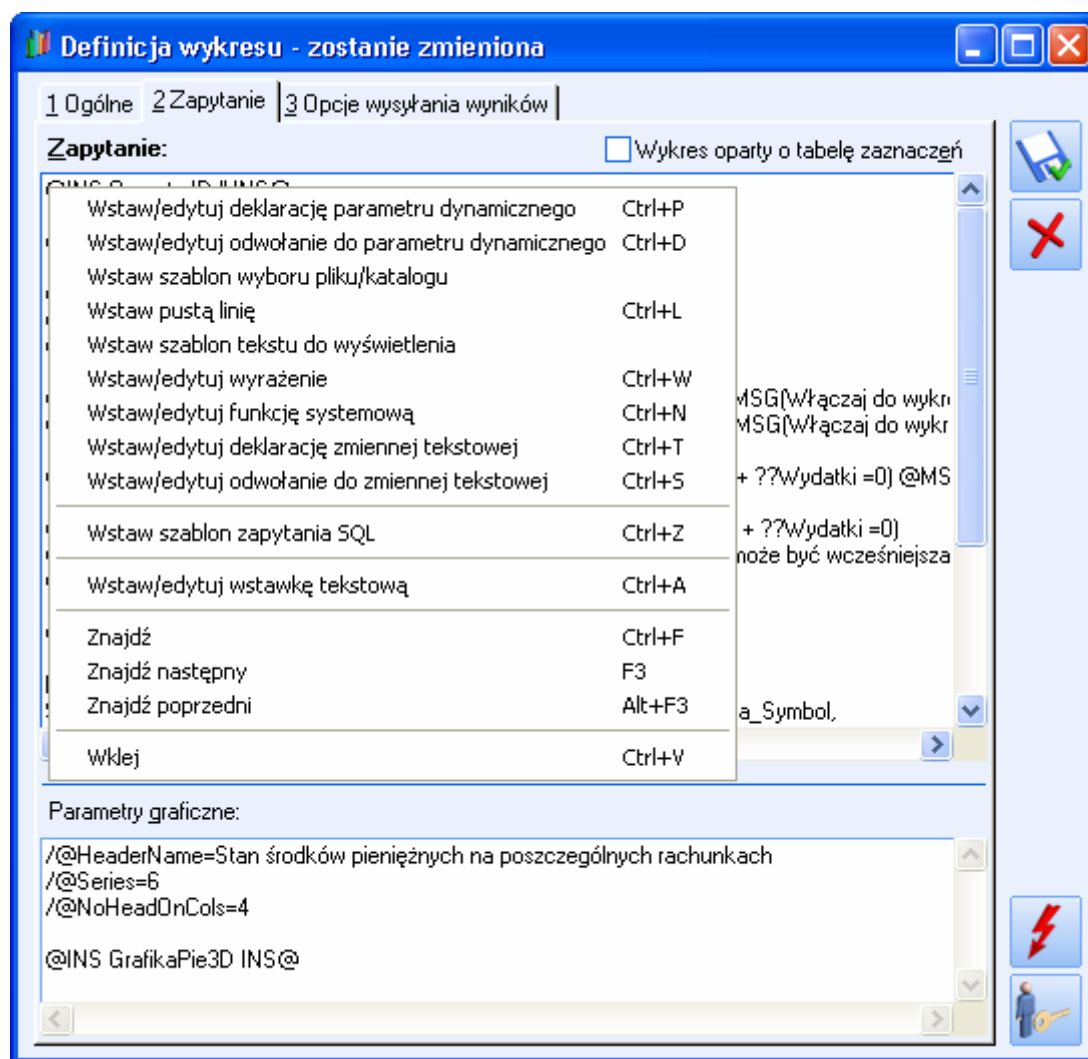
On the right side of the form, there are several icons: a document with a checkmark, a red 'X', a lightning bolt, and a person icon.

Rys. 3 Formularz definicji wykresu – zakładka Ogólne




Przycisk  służy do założenia hasła blokującego zmiany w definicji wykresu dla Użytkowników nie znających hasła. Przy wejściu na formularz tak zabezpieczonego wykresu nie mamy możliwości dokonania jakichkolwiek zmian chyba, że wciśniemy przycisk z kluczykiem i wpisujemy poprawne hasło. W takim przypadku definicja zostanie odblokowana i będzie również możliwa zmiana hasła (w tym jego wyczyszczenie czyli odbezpieczenie definicji).

Zakładka Zapytanie



Rys. 4 Formularz definicji wykresu – zakładka Zapytanie



Warto zwrócić uwagę na przycisk , służący do testowania edytowanego wykresu. Dokładniej przycisk ten służy do wykonania testu edytowanego wykresu z zastosowaniem bieżących ustawień na formularzu. Po jego naciśnięciu jest wykonywany w trybie testowym edytowany wykres. Tryb testowy polega na tym, że bezpośrednio przed przesłaniem zapytania na serwer zapytanie to (ze wstawionymi bieżącymi wartościami parametrów dynamicznych) jest podawane do edycji Użytkownika. Po wykonaniu wykresu jest możliwa normalna nawigacja począwszy od węzła odpowiadającego edytowanemu wykresowi. Wykonaniu każdego wykresu w trybie testowym towarzyszy okno pozwalające na edycję zapytania przed przesłaniem go do serwera.

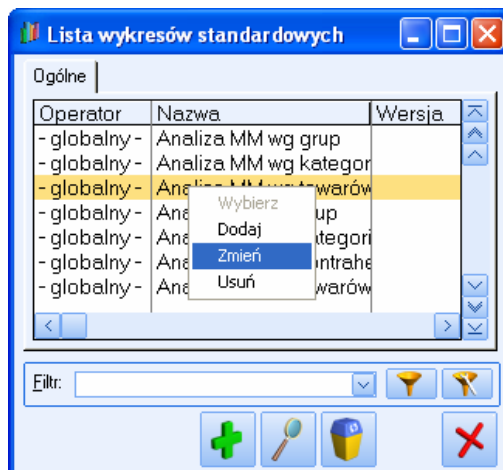
Pozostałe kontrolki są albo standardowe albo ich opis w pełni wyjaśnia zastosowanie. Dlatego tutaj ograniczymy się tylko do nadmienia o jednej istotnej sprawie. Mianowicie w celu ułatwienia posługiwania się językiem stosowanym w definicji wykresu (połączenie SQL i języka parametrów dynamicznych) opracowano system zagnieżdżonych wizardów. Wizardy te pozwalają tworzyć i edytować deklaracje i odwołania do parametrów dynamicznych, wyrażenia, zmienne tekstowe, szablon zapytania SQL itp.. Są one dostępne po kliknięciu prawym klawiszem myszy tekstu zapytania lub parametrów graficznych wykresu.

Zakładka **Opcje wysyłania wyników**

Rys. 5 Formularz definicji wykresu – zakładka Opcje wysyłania wyników

4.1.3 Lista wykresów

Do wyboru wykresu, który chcemy podłączyć jak również do edycji wykresów zaimplementowano **listy wykresów** (rys. 6 przedstawia listę wykresów standardowych; wykresy użytkownika posiadają analogiczną listę). Lista ta może służyć również do edycji wykresów nie podłączonych do żadnego wywołania.



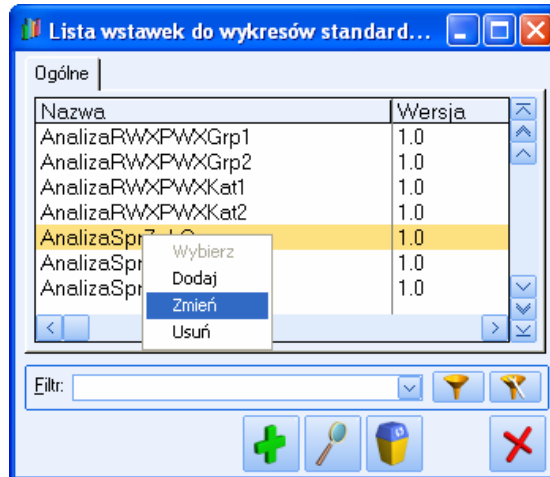
Rys. 6 Lista wykresów (standardowych)

Po wykonaniu akcji dodania lub edycji wykresu jest otwierany opisany w poprzednim rozdziale formularz definicji wykresu.

Należy zwrócić uwagę na to, że dodanie definicji wykresu z tego poziomu nie spowoduje jego podłączenia na drzewie hierarchii. Wykres taki nie będzie nigdzie podłączony do chwili wykonania na nim akcji 'Podłącz wykres' dostępnej na drzewie hierarchii wykresów.

4.1.4 Lista wstawek tekstowych do wykresów

Do edycji wstawek tekstowych służą **listy wstawek tekstowych do wykresów** (rys. 7 przedstawia listę wstawek do wykresów standardowych; wstawki użytkownika posiadają analogiczną listę).



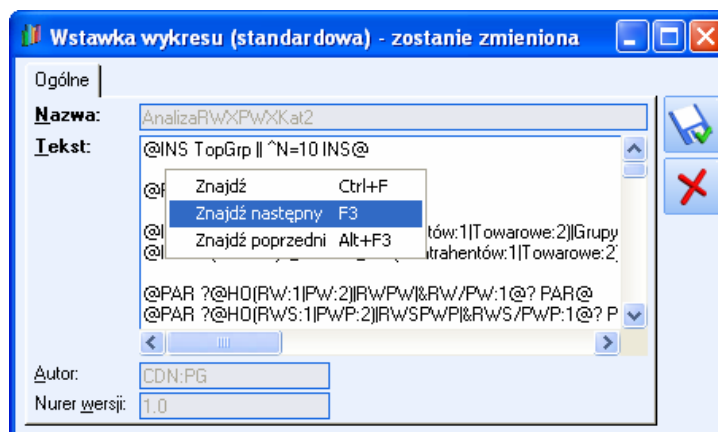
Rys. 7 Lista wstawek tekstowych (do wykresów standardowych)



Uwaga: Z powodu braku relacji program nie sprawdza czy wstawka została gdziekolwiek użyta. Wynika stąd możliwość skasowania wstawki, do której istnieje odwołanie. Należy uważać na to aby nie dochodziło do takich sytuacji.

4.1.5 Formularz wstawki tekstowej do wykresu

Formularz wstawki tekstowej do wykresu wygląda następująco (rys. 8 przedstawia formularz wstawki standardowej; wstawki użytkownika posiadają analogiczny formularz).



Rys. 8 Formularz wstawki tekstowej (standardowej)

4.2 Okno operacyjne wykresów

Głównym elementem modułu wykresów z punktu widzenia Użytkownika jest tzw. okno operacyjne wykresów. Okno to zaraz po otwarciu wyświetla wykres startowy a następnie pozwala na nawigację po grafie wykresów i zmianę parametrów dynamicznych wykonywanych wykresów.

Głównym elementem okna jest kontrolka REGION zawierająca wykres.

Pod wykresem dostępne są przyciski nawigacyjne, przycisk odświeżania wykresu oraz panel z kontrolkami parametrów dynamicznych.

Na drugiej zakładce umieszczone zostały kontrolki dynamicznych parametrów graficznych wykresu (jeśli ich brak – zakładka ta nie pokazuje się) a na trzeciej – lista z danymi serii wykresu przedstawionymi w formie tabelarycznej. Czwarta zakładka w całości dotyczy opcji wysyłania danych wynikowych.


Na toolbarze okna są dostępne przyciski obsługujące wydruki wykresu i eksport danych do Excela. Wydruk z poziomu pierwszych dwóch zakładek produkuje plik .wmf (którego link jest wstawiany do definicji wydruku) a z trzeciej i czwartej zakładki – kolejkę z danymi serii wykresu (wydruk tu podłączony powinien więc operować na kolejce). Eksport danych do Excela obejmuje dane serii wykresu a więc dane z listy na trzeciej zakładce.

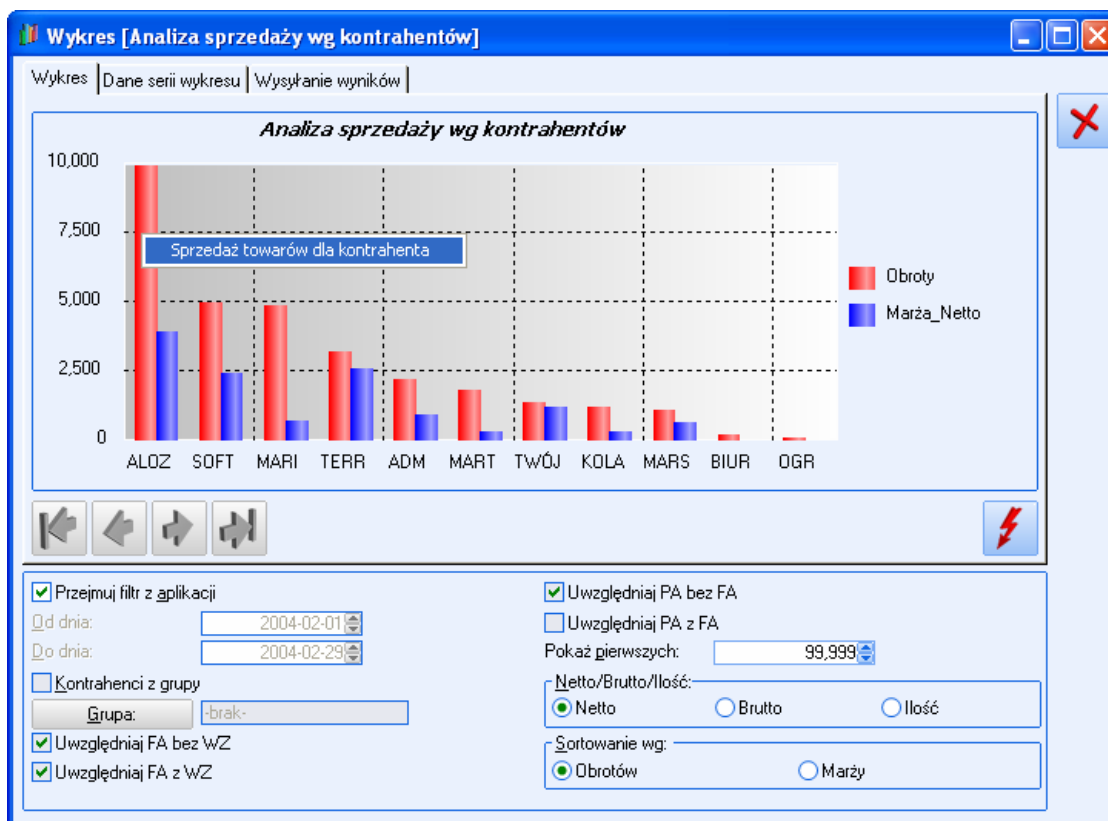
Jedną z podstawowych akcji wykonywanych na wykresie jest eksploracja do wykresu podrzędnego. Następuje to przez kliknięcie lewym klawiszem myszy w wybrany punkt wykresu (wykonanie podwykresu domyślnego) lub kliknięcie prawym klawiszem myszy w wybrany punkt wykresu i wybór z menu podwykresu, który chcemy wykonać.

Umieszczone pod wykresem przyciski nawigacyjne służą do poruszania się po historii wykonanych wykresów od startowego poczynszy. Odpowiada to analogicznym przyciskom w internetowych browserach. Należy jednak zwrócić uwagę, że wykonanie każdej akcji nawigacji (Wstecz, Wprzód itp.) wiąże się z odświeżeniem danych (powtórne załadowanie – brak cache'owania wyniku).

Jak powiedziano wcześniej, bieżący wykres startowy zostaje po raz pierwszy wykonany z wartościami początkowymi parametrów dynamicznych (nadpisany ewentualnie wartościami zmiennych dynamicznych przekazanych z aplikacji).



Po zmianie któregoś z parametrów przycisk odświeżenia zmienia ikonę na  co ma sygnalizować, że dane na wykresie nie odpowiadają bieżącym nastawom parametrów. Aby uzgodnić dane z parametrami należy wykonać odświeżenie. Sytuacja taka nie występuje jeśli w parametrach graficznych wykresu użyto parametru /@AutoRefresh – wówczas dane odświeżają się automatycznie po każdej zmianie dowolnego parametru.



Rys. 9 Okno operacyjne wykresów – zakładka Wykres

Wykres [Analiza sprzedaży wg kontrahentów]

Wykres | Dane serii wykresu | Wysyłanie wyników

PunktID	KntID	KntKod	Obroty	Marża Netto
11	11	11	31,700.06	13,749.96
1	17	ALOZA	9,964.95	4,013.10
2	21	SOFTLAND	5,028.36	2,500.31
3	22	MARIZA	4,946.20	776.70
4	18	TERRA	3,284.43	2,633.43
5	32	ADM	2,281.42	983.07
6	39	MART	Sprzedaż towarów dla kontrahenta }	
7	19	TWÓJGRÓD	1,446.50	1,282.75
8	27	KOLASA	1,260.96	383.61
9	30	MARSZALIK	1,176.37	694.77
10	24	BIUROWIEC	263.75	52.75
11	35	OGRODNIK	162.39	32.49

Przejmij filtr z aplikacji
 Od dnia: 2004-02-01
 Do dnia: 2004-02-29
 Kontrahenci z grupy
 Grupa: -brak-
 Uwzględnij FA bez WZ
 Uwzględnij FA z WZ

Uwzględnij PA bez FA
 Uwzględnij PA z FA
 Pokaż pierwszych: 99,999
 Netto/Brutto/Ilość:
 Netto Brutto Ilość
 Sortowanie wg:
 Obrotów Marży

Rys. 10 Okno operacyjne wykresów – zakładka Dane serii wykresu

Wykres [Analiza sprzedaży wg kontrahentów]

Wykres | Dane serii wykresu | Wysyłanie wyników

Wynik w postaci pliku: Analiza sprzedaży wg kontrahentów.xls dopisuj na końcu istniejącego pliku

w formacie:
 Comma Excel WMF XML Szablon:

Zapisz na dysku w katalogu:
 Przekaż do programu: excel.exe
 Prześlij pocztą e-mail na adres:

Przejmij filtr z aplikacji
 Od dnia: 2004-02-01
 Do dnia: 2004-02-29
 Kontrahenci z grupy
 Grupa: -brak-
 Uwzględnij FA bez WZ
 Uwzględnij FA z WZ

Uwzględnij PA bez FA
 Uwzględnij PA z FA
 Pokaż pierwszych: 99,999
 Netto/Brutto/Ilość:
 Netto Brutto Ilość
 Sortowanie wg:
 Obrotów Marży

Rys. 11 Okno operacyjne wykresów – zakładka Wysyłanie wyników

5 Implementacja techniczna

5.1 Silnik wykresów

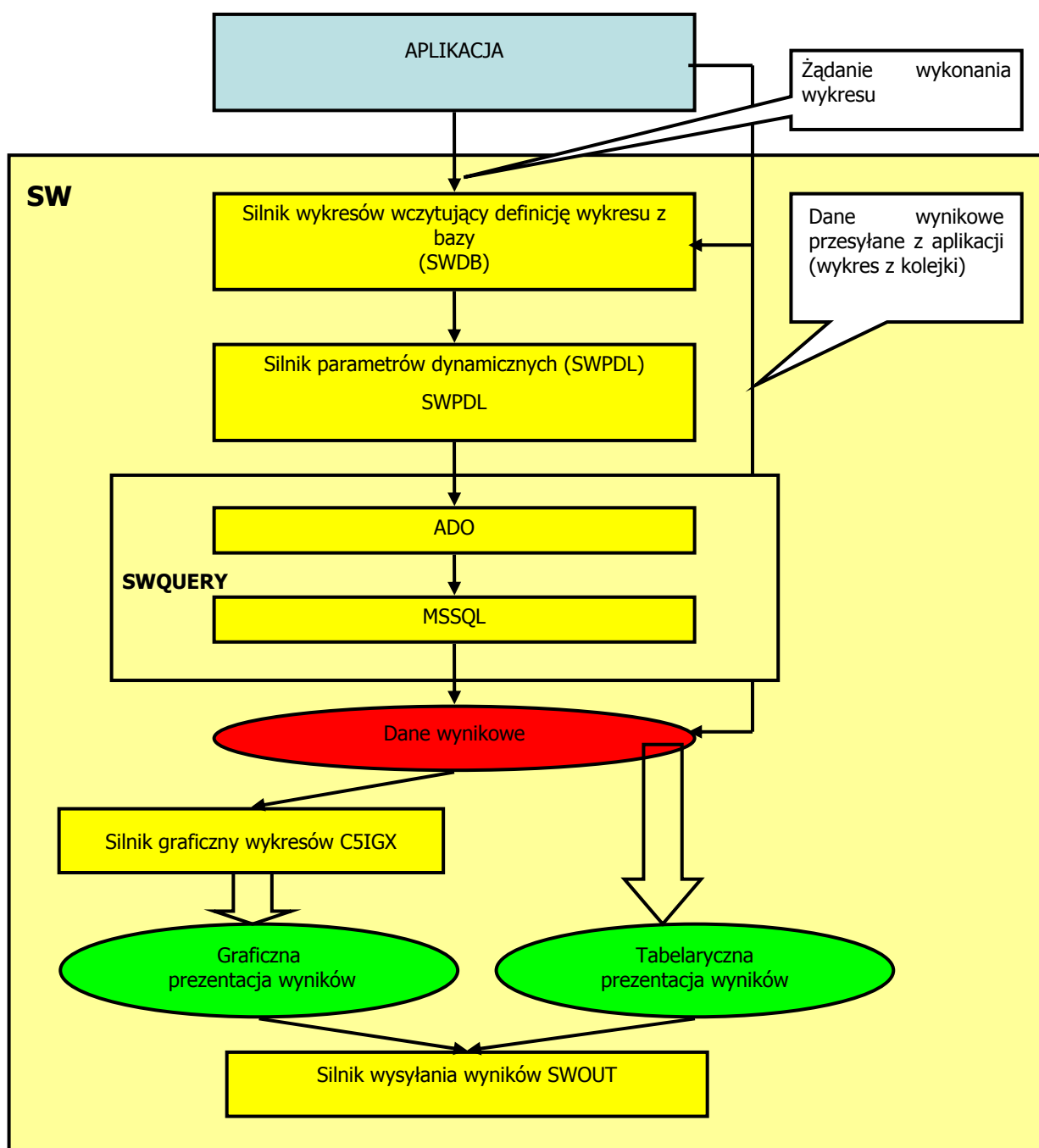
Jak powiedziano wcześniej, za załadowanie danych, a więc najczęściej wykonanie zapytania SQL, odpowiada silnik zapytań wykresów (SWQUERY). Silnik ten wykorzystuje obiekty COM, które odwołują się do bazy danych poprzez ADO. Oczywiście na najniższym poziomie dostępu do danych mamy silnik MSSQL.

Aby jednak wyprodukować ostateczną postać zapytania, które jest podawane do SWQUERY, należy najpierw wczytać z bazy definicję wykresu (robi to silnik wczytujący/zapisujący wykresy SWDB) a następnie poddać zapytanie z definicji odpowiednim przekształceniom (rozwiniecie zmiennych tekstowych, wstawienie wartości parametrów dynamicznych, obliczenie wyrażeń itp.). Wszystkie te czynności są wykonywane przez silnik parametrów dynamicznych SWPDL.

Załadowane przez SWQUERY dane można wyświetlić w postaci tabelarycznej albo graficznej. W tym drugim przypadku wykorzystywany jest kolejny silnik – zewnętrzna biblioteka graficzna C5IGX.DLL. Biblioteka ta wymaga przekazania jej kolejki z danymi na wykresie.

Uzyskane wyniki można na koniec wysłać do urządzenia zewnętrznego (np. zapisać do pliku, podać innemu programowi itp.) – tymi czynnościami zajmuje się silnik wysyłania wyników SWOUT.

Poniżej przedstawiono schemat blokowy silnika analiz.



5.2 Baza danych

5.2.1 Lokalizacja danych

Dane dotyczące wykresów **standardowych** są składowane w postaci tabel topspeedowych, umieszczonych w pliku **CDN_WYKR.TPS** znajdującym się w katalogu z plikami .dll i .exe programu.

Dane związane z wykresami **użytkownika** są przechowywane w analogicznych tabelach (o identycznej strukturze oraz nazwie tabeli z dołączoną cyfrą '2' na końcu) **w bazie SQL**. W OPT!MIE jest to baza konfiguracyjna.

Dodatkowo dane dotyczące możliwych do zmiany (i zmienionych) przez Użytkownika ustawień wykresów standardowych są pamiętane w specjalnych tabelach (z dołączoną literą 'U' na końcu nazwy) w bazie SQL.

5.3 Wydruki wykresów

Okno operacyjne wykresów ma możliwość podłączania wydruków.

- Jeśli wydruk wykonamy z jednej z pierwszych dwóch zakładek – moduł wykresów stworzy plik .wmf na dysku w systemowym katalogu tymczasowym i o nazwie *TempFileName* zwracanej przez system. Ścieżka tego pliku zostanie przekazana w zmiennej dynamicznej *CDN_PlikWmfWykresu* do modułu wydruków, który parsując definicję wydruku przed jego wykonaniem wstawi ścieżkę do `IMAGE('...OPWYKRES.WMF')` w miejsce `...OPWYKRES.WMF`.

Bardzo ważne jest, aby definiując wydruk w kontrolce IMAGE wstawić plik, którego nazwa jest równa OPWYKRES.WMF (katalog dowolny) - w przeciwnym przypadku parser nie dokona wymiany. Po zakończeniu pracy z wydrukiem plik tymczasowy zostanie skasowany.

- Jeśli wydruk wykonamy z trzeciej lub czwartej zakładki – moduł wykresów wypełni globalną kolejkę źródłową wydruków danymi serii. Definicje wydruków podłączonych do tej zakładki powinny więc operować na danych z kolejki.

6 Przykład

Rozważmy następujący przykład.

Założmy, że bieżący wykres startowy jest podpięty do okna z rachunkami bankowymi i ma wyświetlić obroty na poszczególnych rachunkach odfiltrowanych tak jak na liście a dodatkowo takich, że obroty na nich przekraczają ustaloną na specjalnym spinie wartość. Założmy także, że chcemy osobno uwzględnić bądź nie raporty zamknięte i otwarte. Eksploracja punktu (odpowiadającego wybranemu rachunkowi) powinna móc spowodować albo otwarcie wykresu rozbicia w/w obrotów na obroty na poszczególnych raportach (*drill down*) rachunku albo wywołanie listy raportów rachunku (*drill through*).

Dane bieżącego wykresu startowego mogą być następujące:

Nazwa = 'Obroty na poszczególnych rachunkach'

TypAkcji = e_op_WykresyTypAkcjiSQL

Zapytanie = 'SELECT PunktID=1, RachunekID=Bra_BraID, RachunekSymbol=Bra_Symbol, BRa_SaldoBO, ObrotyNaRachunku=SUM(BRp_Przychody- BRp_Rozchody)

FROM cdn.BnkRachunki JOIN cdn.BnkRaporty ON Bra_BraID=BRp_BraID

WHERE ((BRp_Zamkniety=1 AND ?@X|RaportyZamkniete|Raporty & zamknięte:1@?=1) OR (BRp_Zamkniety=0 AND ?@X|RaportyOtwarte|Raporty & otwarte:1@?=1)) AND

BRa_Typ=?@O(Kasa:1|Konto:2|Karta:3)|RachunekTyp|&Typ rachunku:2@? AND

?@s255_NQ|FiltrApp|Filtr z aplikacji:@?

GROUP BY Bra_BraID, Bra_Symbol, Bra_SaldoBO HAVING SUM(BRp_Przychody- BRp_Rozchody) >= ?@n14.2|MinObroty|Obroty minimum:100.00@?'

ParametryGraficzne= '@Type=1 /@Depth=100 /@HeaderName=Obroty na rachunkach /@XAxisName=Rachunek /@YAxisName=Obroty /@Series=5,4'

Dane pierwszego 'podwykresu':

Nazwa = 'Obroty raportów w ramach rachunku'

TypAkcji = e_op_WykresyTypAkcjiSQL

Zapytanie = 'SELECT Brp_BrpID, RaportID=Brp_BrpID, NumerRaportu=Brp_NumerPelny, ObrotyRaportu=BRp_Przychody-BRp_Rozchody

FROM cdn.BnkRaporty

WHERE

BRp_BraID=?@n3|RachunekID|Id raportu:1@? AND

((BRp_Zamkniety=1 AND ?@X|RaportyZamkniete|Raporty & zamknięte:1@?=1) OR (BRp_Zamkniety=0 AND ?@X|RaportyOtwarte|Raporty & otwarte:1@?=1))

ORDER BY Brp_BrpID'

ParametryGraficzne = '@Type=1 /@Depth=100 /@HeaderName={'Obroty na raportach rachunku ' & ?@Hs20_Q|RachunekID|:1@?} /@XAxisName=Raport /@YAxisName=Obroty /@Series=5,4'

Dane drugiego 'podwykresu':

Nazwa = 'Lista raportów rachunku'

TypAkcji = e_op_WykresyTypAkcjiProc

Zapytanie = OP_KASAI.DLL|CALLBRPLISTA('BRA',?@s6|RachunekID|Rachunek:0@?)

ParametryGraficzne = ''

Programista w danym wywołaniu przekazuje do obiektu wykresu związanego z tym wywołaniem pewne parametry wynikające z bieżących ustawień zmiennych w aplikacji, np. filtr zastosowany na liście rachunków bankowych (zmienna dynamiczna *FiltrApp*). Przed wykonaniem bieżącego wykresu startowego obiekt wykresu poszukuje w zapytaniu parametrów. Znajduje, że należy podać wartości parametrów *RaportyZamkniete*, *RaportyOtwarte*, *RachunekTyp*, *FiltrApp* i *ObrotyMin*. Poszukuje parametrów o takich nazwach w kolejce zmiennych dynamicznych przekazanych z aplikacji i znajduje tylko jeden z nich: *FiltrApp*. Wpisuje jego wartość do kolejki a wartości innych, nie znalezionych parametrów pobiera z ich defaultów. Po podstawieniu wartości parametrów do zapytania jest wykonywany wykres i tworzone są kontrolki ze wszystkimi parametrami dynamicznymi. Teraz można zmienić wartości w tych kontrolkach i ponownie wykonać wykres (przycisk *Odśwież*).

Eksploracja punktu bieżącego wykresu startowego do pierwszego podwykresu:

Mamy unikalne Id klikniętego punktu; odczytujemy rekord z danymi tego punktu. Z zapytania w pierwszym podwykresie odczytujemy, że potrzebna jest wartość parametrów *RachunekID*, *RaportyZamkniete* i *RaportyOtwarte*. Przeszukujemy nazwy kolumn zapytania wykresu nadrzędnego aby sprawdzić, w których kolumnach znajdują się szukane wartości parametrów: poszukujemy kolumn o nazwach *RachunekID*, *RaportyZamkniete* i *RaportyOtwarte*. Znajdujemy tylko pierwszą z nich (kolumna 5.). Wartości dwóch pozostałych parametrów 'dziedziczymy' z ustawionych parametrów

wykresu nadrzędnego (zgodność nazw parametrów w obu zapytaniach). Podstawiamy odczytane wartości wszystkich parametrów i wykonujemy zapytanie podwykresu.

Eksploracja punktu bieżącego wykresu startowego do drugiego podwykresu:

Mamy unikalne Id klikniętego punktu; odczytujemy rekord z danymi tego punktu. Z zapytania w drugim podwykresie odczytujemy, że potrzebna jest wartość parametru *RachunekID*. Przeszukujemy nazwy kolumn zapytania wykresu nadrzędnego aby sprawdzić, w której kolumnie znajduje się szukana wartość parametru: poszukujemy kolumny o nazwie *RachunekID*. Znajdujemy – jest to 5. kolumna. Podstawiamy odczytaną wartość parametru i wykonujemy zapytanie podwykresu (wołamy funkcję).