



## **Dodawanie operacji dodatkowych w WAPRO Mag.**

obowiązuje od wersji 8.21.0

**Opracował i wykonał:  
Grzegorz Lenarczyk**

**Asseco Business Solutions SA  
Oddział w Warszawie**  
Warszawa, ul. Branickiego 13  
02-972 Warszawa

[www.wapro.pl](http://www.wapro.pl)

Copyright © Asseco Business Solutions SA

## 1. Dodawanie operacji dodatkowych (modułów rozszerzeń) do WAPRO Mag.

W WAPRO Mag od wariantu BIZES istnieje możliwość dołączania dodatkowych, tworzonych na życzenie użytkownika, modułów rozszerzających możliwości programu. Moduły rozszerzeń muszą mieć postać wykonywalnych programów/procedur tzn. EXE lub BAT. Dołączane moduły integrują się z interfejsem WAPRO Maga jako operacje dodatkowe dostępne ze standardowego menu wywoływanego pod przyciskiem ekranowym „Operacje” (skrót Ctrl+O) lub pod prawym przyciskiem myszy w sekcji operacje dodatkowe.

Dodanie nowego modułu rozszerzeń jako operacji dodatkowej może być wykonane jedynie przez użytkownika zalogowanego do programu jako administrator systemowy czyli predefiniowany użytkownik admin zakładany w momencie instalacji bazy danych programu. Możliwości dodawania nowych modułów rozszerzeń włączane są na oknie operacji dodatkowych (wywoływanym pod przyciskiem ekranowym „Operacje”) poprzez naciśnięcie klawiszy Ctrl+Shift+Alt+?. Pojawiają się wówczas standardowe przyciski „Dodaj”, „Popraw” i „Usuń”. Wybranie przycisku „Dodaj” prowadzi do ekranu jak na rys.1

Rys.1 Ekran dodawania operacji dodatkowej (modułu rozszerzeń)

1. *Kontekst wywołania funkcji* – oznacza miejsce w interfejsie WAPRO Maga gdzie operacja dodatkowa będzie dostępna. Jest to również miejsce skąd wywołano ekran dodawania operacji dodatkowej. Pole nie podlega edycji i jest domyślnie inicjowane.
2. *Nazwa funkcji* – oznacza nazwę operacji dodatkowej, która będzie widoczna dla użytkownika w oknie wyboru operacji oraz pod prawym klawiszem myszy

3. *Skrót klawiszowy* – oznacza kombinację klawiszy, którą wywoływana będzie z interfejsu dodawana operacja dodatkowa. Kombinacje klawiszy dostępne są z listy możliwych skrótów klawiszowych. Na liście pokazane są wszystkie możliwe – niektóre jednak mogą być blokowane są przez interfejs Windows (np. Alt+<litera>) a niektóre mogą być już wykorzystywane. Proszę upewnić się po zdefiniowaniu, że skrót dla danej operacji rzeczywiście działa.
4. *Kod operacji* – to wewnętrzny identyfikator operacji dodatkowej, po której będzie ona identyfikowana w programie WAPRO Mag.
5. *Funkcja dostępna w wariantach* – oznacza dostępność operacji dodatkowej w wybranych wariantach. Obecnie przyjmuje się, że każda ze zdefiniowanej operacji będzie dostępna w wariantach PRESTIŻ i Prestiż PLUS
6. *Zmienne przesyłane do modułu* – opcja umożliwia przesłanie do modułu rozszerzeń wartości ośmiu predefiniowanych w WAPRO Magu zmiennych, które mogą być dowolnie wykorzystywane np. przy konstruowaniu kontekstowych zapytań do bazy danych WAPRO Maga. Wartości tych zmiennych są takie jakie zostały zainicjowane w WAPRO Magu w momencie wywołania operacji dodatkowej przez użytkownika. Zmienne „id\_firmy”, „id\_magazynu” i „id\_użytkownika” przechowują wartości identyfikatorów Firmy, Magazynu i Użytkownika z tabel bazy danych. Identyfikator „id\_obiektu” ma takie samo znaczenie jak przy definiowaniu interfejsu przy podłączaniu raportów w CR i zależy od kontekstu wywołania operacji np. na dokumentach handlowych będzie to „id\_dokumentu\_handlowego” podświetlonego na liście dokumentów w momencie wywołania operacji. Zmienne „zakres daty od”, „zakres daty do” oraz „opis zakresu” przechowują aktualny w momencie wywołania globalny filtr zakresu dat. Oczywiście wartości zmiennych „zakres daty od”, „zakres daty do” są zapisane w standardzie Clariona czyli w postaci liczby dni od 28.12.1800 roku. Aby dokonać konwersji na postać akceptowaną przez SQL Server (typ datetime) należy wykonać na serwerze np. funkcję cast(„zakres daty od” -36163 as datetime). Zmienna „Kod kontekstu (użycie)” zwraca wartość tzw. użycia czyli liczbę całkowitą zależną od miejsca gdzie została wykonana operacja. Użycie jest parametrem wykorzystywanym przy zaznaczaniu dokumentów, artykułów czy kontrahentów i odpowiada wartości kolumny „użycie” w tabeli „ZAZNACZONE” przechowującej identyfikatory (w kolumnie „Id”) zaznaczonych obiektów (dokumentów, kontrahentów lub artykułów).
7. *Ścieżka do modułu* – oznacza lokalizację pliku modułu rozszerzeń wraz z jego nazwą widoczną dla wszystkich końcówek WAPRO Maga, które powinny korzystać z tego modułu rozszerzeń. W szczególnym przypadku może to być katalog instalacyjny programu (nie zalecane!).
8. *Zatrzymanie aplikacji do czasu ukończenia modułu* – zaznaczenie tej opcji powoduje, że WAPRO Mag blokuje interfejs dla użytkownika wywołującego operację dodatkową do czasu aż moduł zakończy swoją pracę i przekaże sterowanie do systemu operacyjnego. Nie zaznaczenie tej opcji powoduje, że moduł i WAPRO Mag mogą wykonywać się niezależnie.
9. *Sposób wywołania* – prezentuje sposób wywołania modułu rozszerzeń z linii komend systemu operacyjnego. Przy wywołaniu operacji dodatkowej po ścieżce do modułu (lokalizacji pliku modułu wraz z nazwą pliku) WAPRO Mag zawsze umieści w linii komend dwa parametry oddzielone spacjami. Parametr nr 1 to tzw. ConnectString czyli parametry połączenia z bazą danych w postaci nazwy serwera, nazwy bazy danych, nazwy użytkownika na serwerze bazy

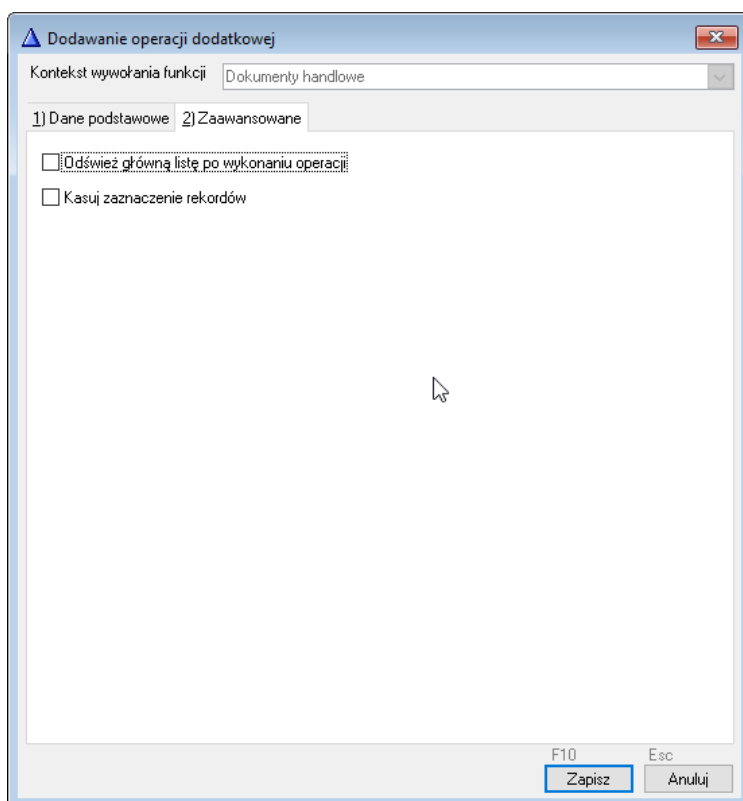
danych (użytkownik serwera a nie programu!!!) oraz hasła tego użytkownika oddzielonych przecinkami. Jest to najprostsza forma przekazania parametrów połączenia z bazą danych w standardzie ODBC. Parametr nr 2 zawiera uporządkowaną listę wartości zmiennych przekazywanych do modułu (patrz pkt.6) oddzieloną przecinkami. Kolejność umieszczania tych wartości obrazowo przedstawiona jest na białym polu u dołu okna dodawania operacji. Oczywiście w przypadku braku konieczności przesyłania wartości tych zmiennych parametr nr 2 będzie ciągiem pustym. UWAGA! Parametry nr 1 i nr 2 są przekazywane w linii komend systemu w postaci ciągu tekstowego w cudzysłowach podwójnych („”).

10. *Zakładka2 – odśwież główną listę po wykonaniu operacji*

Znacznik ten pozwala odświeżyć listę po wykonaniu operacji dodatkowej z której jej wykonywana operacja. (Rys. 2)

11. *Zakładka2 – Kasuj zaznaczenie rekordów*

Znacznik ten pozwala usunąć zaznaczenia na liście po wykonaniu operacji dodatkowej. (Rys. 2)



Rys.2 Zakładka zaawansowane

Sposób przekazania parametrów oraz wywołania jest analogiczny dla skryptów Visual Basic Scripts (VBS).

Nieco inaczej wygląda definiowanie operacji dodatkowej dla procedury SQL. W polu *Nazwa procedury* należy podać część nazwy procedury – w celu ujednoczenia rejestracji rozszerzeń w bazie danych program wymaga aby nazwa procedury poprzedzona była prefiksem `MAGEXT_`. Dodatkowo mamy możliwość

określenia czy program będzie wyświetlał okienko pytania przed wykonaniem procedury oraz komunikat potwierdzający zakończenie wykonania.

Jeśli zdefiniujemy na liście parametrów przekazywanych do procedury `id_obiektu` oraz `kod_kontekstu(użycie)` pojawi się opcja *Wykonaj w pętli dla zaznaczonych* (rys.3). Umożliwia ona sekwencyjne wywołanie zdefiniowanej procedury dla wszystkich obiektów zaznaczonych na ekranie przez użytkownika w danym kontekście wywołania. W trakcie sekwencyjnego wykonania wyświetlany jest na ekranie pasek postępu. Sekwencyjne wykonanie jest szczególnie zalecane jeśli przewidywany czas wykonania operacji jest długi. Operację wykonywaną w pętli użytkownik może przerwać klawiszem Esc. Jego naciśnięcie spowoduje wyświetlenie prośby o potwierdzenie przerwania wykonywanej operacji. Jeśli nastąpiło to w czasie komunikacji z serwerem (wykonania procedury SQL) program będzie czekał na zakończenie wykonania procedury wyświetlając stosowny komunikat. Dla ułatwienia implementacji procedury SQL przygotowano również jej schemat dostępny pod przyciskiem *Pokaż schemat procedury SQL*. Przykład wygenerowanej procedury jest na Rys. 4

**Dodawanie operacji dodatkowej**

Kontekst wywołania funkcji: Dokumenty handlowe

1) Dane podstawowe 2) Zaawansowane

Nazwa funkcji: \_\_\_\_\_

Skrót klawiszowy: \_\_\_\_\_

Kod operacji: 1111\_000\_ \_\_\_\_\_ Poziom grupowania: 0

Moduł wykonywalny  Skrypt VBS  **Procedura SQL**

Zmienne przesyłane do modułu

|   |   |
|---|---|
| <input type="checkbox"/> 1. Id_firmy                  | <input type="checkbox"/> 6. Zakres daty do                    |
| <input type="checkbox"/> 2. Id_magazynu               | <input type="checkbox"/> 7. Opis zakresu                      |
| <input checked="" type="checkbox"/> 3. Id_obiektu     | <input checked="" type="checkbox"/> 8. Kod kontekstu (użycie) |
| <input checked="" type="checkbox"/> 4. Id_użytkownika |   |
| <input type="checkbox"/> 5. Zakres daty od            |   |

Funkcja dostępna w wariantach

|   |
|---|
| <input checked="" type="checkbox"/> START   |
| <input checked="" type="checkbox"/> BIZNES  |
| <input checked="" type="checkbox"/> PRESTIŻ |
| <input checked="" type="checkbox"/> MAX     |

Procedura SQL

Nazwa procedury: MAGEXT\_ PrzykładowaProcedura

Wykonaj w pętli dla zaznaczonych  Pytanie przed wykonaniem  Potwierdzenie wykonania

Procedurę składaną SQL należy umieścić w bazie danych programu. Zalecamy stosowanie transakcji poziomu REPEATABLE READ dla kodu pisanego wewnątrz procedury. Dla ułatwienia implementacji został przygotowany ogólny schemat procedury SQL.

id\_obiektu,id\_użytkownika,kod kontekstu (użycie),

F10 Zapisz Esc Anuluj

Rys.3 Ekran dodawania operacji dodatkowej (procedura SQL)

```
Schemat procedury SQL MAGEXT_PrzykladowaProcedura
if exists (select 1 from sysobjects where name = 'MAGEXT_PrzykladowaProcedura' and type = 'P')
  drop procedure MAGEXT_PrzykladowaProcedura
go
create procedure MAGEXT_PrzykladowaProcedura
  @id_obiektu numeric,
  @id_uzytkownika numeric,
  @kod_kontekstu int
as
declare @errmsg varchar(255)
begin
  set xact_abort on
  set transaction isolation level REPEATABLE READ
  begin transaction

  select @errmsg = 'Operacja dodatkowa <NazwaOperacji> nie została jeszcze zdefiniowana!'
  goto Error

  if @@trancount>0 commit transaction
  goto Koniec
Error:
  raiserror (@errmsg,16,1)
  if @@trancount>0 rollback tran
  goto Koniec
Koniec:
  set transaction isolation level READ COMMITTED
  return
end
go
```

Ctrl+C      Ctrl+W      Esc  
Kopiuł do schowka zaznaczony fragment      Wykonaj SQL      Zamknij

Rys.4 Przykład wygenerowanej procedury.