

## Integracja z zewnętrznymi systemami

### Sposób utworzenia połączenia z systemem Optima

Aby umożliwić synchronizację z systemem Optima należy wykonać poniższe czynności.

- zmiana stałej w pliku config.inc
- sprawdzenie danych w tabeli wsdl\_config\_table w systemie eDokumenty
- utworzenie tabeli kolejki w systemie Optima
- założenie triggerów na tabele
  - dok\_Dokument
  - adr\_Ewid
  - kh\_Kontrahent
- założenie widoków (eDokumenty > 2.0 RC15)
  - subiekt\_export\_all\_contacts\_documents\_view.sql
  - subiekt\_export\_all\_contacts\_view.sql
  - subiekt\_export\_all\_queued\_contacts\_documents\_view.sql
  - subiekt\_export\_all\_queued\_contacts\_view.sql

### Zmiana stałej

W pliku config.inc należy zmienić stałą "SUBIEKT\_DATA\_SOURCE\_DTSCNM" na

```
define('OPTIMA_DATA_SOURCE_DTSCNM', 'nazwa');
```

gdzie nazwę należy zastąpić nazwą bazy danych systemu Optima

### Sprawdzenie danych w tabeli

Jeśli w systemie eDokumenty tabela wsdl\_config\_table jest pusta należy wykonać odpowiedni skrypt SQL dla Optima znajdujący się on w katalogu

```
apps\edokumenty\classes\BsConnect\install\data\optima_wsdl_ins.sql
```

następnie należy przejść do punktu "Konfiguracja tabeli wsdl\_config\_table" aby dokonać niezbędnych poprawek oraz ustawień według zaleceń.

### Ustawienia w tabeli wsdl\_config\_table

Następny krok to sprawdzenie tabeli wsdl\_config\_table oraz czy zawiera dane. W przypadku jeśli jest pusta proszę udać się na stronę z integracją wybranego systemu. Edycję ustawień połączeń możemy dokonać z interfejsu eDokumentów **Ustawienia -> Systemy zewnętrzne**

### Opis kolumn

Adres - adres pliku (serwera), jeśli w nazwie występuje słowo {host} należy ją zamienić na lokalizację systemu eDokumenty standardowo jest to localhost, w przypadku vhosta należy podać port (np.: localhost:8080), jeśli system znajduje się w katalogu różnym niż public należy dodać nazwę tego katalogu do nazwy (np.: localhost:8080/edokumenty), zmianę tą można dokonać z linii poleceń psql według ustawień instalacji

```
UPDATE wsdl_config_table SET wsdl__ = replace(wsdl__, '{host}', 'localhost:8080/edokumenty');
```

- System - wskazuje na nazwę systemu dla którego konfigurowana jest dana metoda (zostawiamy bez zmian)
- Warunek SQL - warunek po jakim będą synchronizowane dane
  - dla akcji "Podwiąż kontakt" z lewej strony znaku równości wskazujemy kolumnę z systemu zew. natomiast z prawej w wężach kolumnę z systemu eDokumenty

```
(np. dla OPT!MY: Knt_Nip='{nip___}' AND Knt_Nazwa1='{name_1}' co oznacza, że będzie wyszukiwanie kontaktu w systemie OPT!MA gdzie kolumna Knt_Nip będzie równa numerowi nip kontaktu z systemu eDokumenty (token {nip___} jest zamieniany na dane) itd)
```

- dla pozostałych akcji po lewej stronie jest nazwa kolumny z systemu eDokumenty a z prawej token z maski(zobacz w katalogu appsledokumenty\etc\sync) z jakimi zostanie zastąpiony.

Różnica między tymi metodami polega na tym iż w przypadku szukania kontaktu w systemie zew. należy wykonać zapytanie na zew. bazie dlatego z lewej strony są nazwy kolumn z systemu zew. a z prawej tokeny nazwy kolumn systemu eDokumenty, które zostaną zastąpione danymi wybranego kontaktu.

Reszta metod służy do porównania danych przychodzących z systemu zew. do systemu eDokumenty. Zapytanie jest wykonywane na bazie eDokumenty dlatego z lewej strony wstawiamy nazwę kolumny z tabeli z bazy eDokumenty natomiast z prawej dane z tablicy "mapy" z systemu zew.

Tablica (mapa kolumn) ułatwia sparsowanie danych i lepsze rozeznanie przykład

```
apps\edokumenty\etc\sync\OPTIMA_columns_map.ini
Przykładowa konfiguracja tablicy mapy dla kontaktu dla OPT!MY
[contacts]
Knt_KntID = contid
Knt_Nazwa1 = name_1
Knt_Nazwa2 = name_2
Knt_Nazwa3 = name_2
Knt_Kraj = countr
Knt_Wojewodztwo = woj___
Knt_Powiat = powiat
Knt_Ulica = street
Knt_NrDomu = bldnum
Knt_NrLokalu = fltnum
Knt_Miasto = city__
Knt_KodPocztowy = code__
Knt_Nip = nip___
```

Oznacza to, że do dyspozycji będą dane pod danymi kluczami np.: klucz "nip\_\_\_" będzie zawierał numer nip kontaktu z systemu zew. (w tym przypadku OPT!MA) dlatego warunek zapytania w bazie eDokumenty będzie miało postać

```
nip___='{nip___}'
```

Czyli szukamy kontaktu w bazie eDokumenty gdzie nip\_\_\_ (lewa strona) kontaktu z bazy eDokumenty jest równy numerowi nip z tabeli - mapy danych jakie otrzymamy z systemu zew. Należy pamiętać aby token w wąsach był dodatkowo w pojedynczych apostrofach. Ze względu na różny typ danych i sposób w jaki mogą być potraktowane przez SQL (cyfry, liczby nie wymagają apostrofów natomiast litery tak!!) lepiej jest dla każdego typu danych w wąsach stosować apostrofy.

## Utworzenie tabeli kolejki

W systemie Optima (w bazie) wykonujemy skrypt z pliku

```
apps\edokumenty\classes\BsConnect\install\sql\optima\optima_export_queue.sql
```

Po wykonaniu tego skryptu w systemie Optima powinna pojawić się dodatkowa tabela o nazwie export\_queue.

## Założenie triggerów

Triggery mają za zadanie dodawać do kolejki dokumenty i kontaktu które zostały zmodyfikowane bądź dodane do systemu Subiekt. Triggery wykonujemy z załączników bądź w folderze

```
apps\edokumenty\classes\BsConnect\install\sql\optima
```

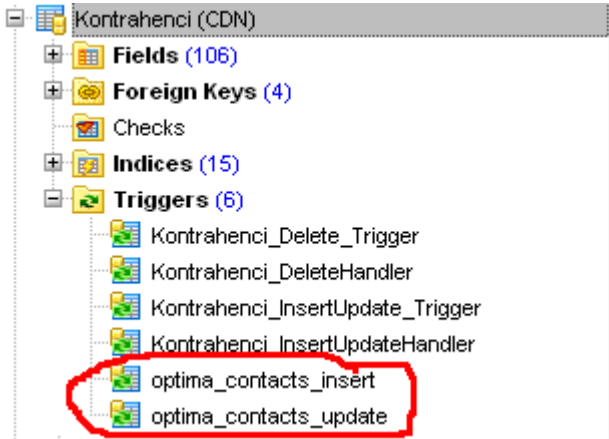
znajdują się pliki oryginalne. Kolejność ich wykonywania nie ma znaczenia. Po ich wykonaniu należy sprawdzić czy tabele dokDokument, adrEwid oraz Kontrahent posiadają dodatkowe triggery. Można tego dokonać za pomocą narzędzia SQL Manager Lite for SQL Server firmy EMS.

(<http://www.sqlmanager.net/>)

Dla tabeli dokumentów



Natomiast dla tabeli Kontrahenci



Możemy przetestować działanie mechanizmu poprzez dodanie do systemu Subiekt dokumentu lub kontrahenta i sprawdzeniu czy w tabeli export\_queue pojawiły się wpisy.

## Założenie widoków

Począwszy od wersji 2.0 RC16 można decydować jakie dane mają być pobierane z systemu zew. poprzez modyfikacje specjalnych widoków przeznaczonych do synchronizacji. Widoki te są wymagane w celu prawidłowego funkcjonowania. Można je znaleźć w katalogu

```
edokumenty\classes\BsConnect\install\sql\optima\views
```

Można też je pobrać z załączonego niżej katalogu optima.zip. Widoki należy wykonać na bazie Optima (MySQL).

### Opis widoków

- subiekt\_export\_all\_contacts\_documents\_view - wyświetla wszystkie dokumenty, które mają kontrahenta
- subiekt\_export\_all\_contacts\_view - wyświetla wszystkich kontrahentów
- subiekt\_export\_all\_queued\_contacts\_documents\_view - wyświetla wszystkie dokumenty, które mają kontrahenta oraz zostały zmodyfikowane(dodane do systemu) i znajdują się w tabeli export\_queue w systemie zew.
- subiekt\_export\_all\_queued\_contacts\_view - wyświetla wszystkich kontrahentów którzy zostali zmodyfikowani(dodani do systemu) i znajdują się w tabeli export\_queue w systemie zew.

kolejność wykonywania nie ma znaczenia.

Widoki te można modyfikować według potrzeb(dodawac kolumny itd) eliminując w ten sposób dodatkową pracę ze strony programisty i aktualizacji w postaci nowej wersji ze zmodyfikowanym widokiem. Widoki modyfikujemy jeśli klient zażyczy sobie aby z systemu zew. były pobierane dodatkowe dane, których podstawowa definicja nie uwzględniła. Następnie w pliku konfiguracyjnym

```
SUBIEKT_columns_map.ini
```

dodajemy kolejny wpis pod odpowiednim indeksem [contacts] - kontakty, [documents] - dokumenty z rzutowaniem danych z dodanej kolumny w widoku na kolumnę w systemie eDokumenty.

### Jaka metoda jaki widok wykorzystuje

- Importuj wszystkie kontakty - subiekt\_export\_all\_contacts\_view
- Aktualizuj dane - subiekt\_export\_all\_queued\_contacts\_view
- Aktualizuj dokumenty - subiekt\_export\_all\_queued\_contacts\_documents\_view
- Aktualizuj dane kontaktów - subiekt\_export\_all\_queued\_contacts\_view
- Aktualizuj dokumenty kontaktów - subiekt\_export\_all\_queued\_contacts\_documents\_view
- Pobierz wszystkie dokumenty - subiekt\_export\_all\_contacts\_documents\_view

## Reset

Na wypadek gdyby wprowadzone zmiany miały być z jakiegoś względu wycofane z bazy optymy, należy wykonać plik reset\_optima.sql który usuwa tabele export\_queue, triggerzy oraz widoki.

### **Raporty z parametrem klienta**

Aby raportować rozrachunki z klientem, sprzedaż czy cokolwiek z nim związane można do tego użyć raportów ze zdefiniowanym zewnętrznym źródłem danych patrz: [tworzenie raportów SQL](#)