

Automatyczny import dokumentów do Symfonia Handel

Przykład

1	Wprowadzenie	2
2	Przykładowy przykład, czyli tzw. case	2
3	Koncepcja rozwiązania	2
3.1	Materiały	2
4	Praktyczny przykład rozwiązania	3
4.1	Przygotowanie importu danych	3
4.2	Uruchomienie importu z poziomu języka AmBasic	5
4.3	Cykliczne uruchamianie raportu z pomocą procedury OnTimer	6
5	Uwagi praktyczne	8
6	Przykładowe modyfikacje	9
7	Spis rysunków	9
8	Spis tabel	9

1 Wprowadzenie

Symfonia Handel wspomaga automatyzację procesów wprowadzania dokumentów. W szczególności umożliwia import dokumentów, np. w formacie 3.0, bezpośrednio do bufora odpowiedniej kartoteki. Domyślnie jednak operacja ta wymaga... operatora, czyli kogoś, kto ten import wykona. Tak jest domyślnie, ale można w prosty sposób skonfigurować program Symfonia Handel tak, by automatycznie importował dokumenty.

Dokument ten opisuje przykład konfiguracji programu Symfonia Handel do automatycznego importu dokumentów.

2 Przykładowy przykład, czyli tzw. case

Dosyć często mamy do czynienia z sytuacją, że regularnie (co pewien czas) spływają do Użytkownika dokumenty z innego systemu, które należy zaimportować. Użytkownik musi o tym pamiętać. I absorbuje to jego uwagę. A Użytkownik chciałby, żeby takie dokumenty same wczytywały się do systemu.

Przyjmujemy więc następujące założenia:

- Spływające dokumenty są w postaci plików tekstowych w formacie 3.0
- Trafiają w jakiś sposób do pliku o określonej i znanej nam lokalizacji i nazwie
- Znamy ich rodzaj (założmy, że są to np. dokumenty magazynowe)
- Chcemy je automatycznie zaimportować do bufora

Celem tego dokumentu jest pokazanie możliwości, jakie niesie ze sobą program Symfonia Handel. Przyjęte powyżej założenia (dość słabe) można więc zmienić/rozciągnąć/zlikwidować. Wpłynie to oczywiście na rozwiązanie tworząc je bardziej rozbudowanym, ale i bardziej uniwersalnym. Dla prostoty i przejrzystości dokumentu oprzemy się na nich.

3 Koncepcja rozwiązania

Przy tworzeniu rozwiązania oprzemy się o następujące funkcjonalności programu Symfonia Handel:

- a. Import danych w formacie 3.0
- b. Możliwość tworzenia własnych skryptów/programów w wewnętrznym skryptowym języku AmBasic
- c. Specjalna procedura OnTimer uruchamiana cyklicznie przez aplikację Symfonia Handel

Jednym zdaniem: cyklicznie (w procedurze OnTimer) będzie uruchamiany raport importujący z plików dokumenty.

3.1 Materiały

Szczegółowe informacje (dokumentacja) do funkcjonalności, o które opiera się koncepcja rozwiązania, zawarte są w dokumentacji instalowanej razem z aplikacją (patrz: Tabela 1).

Tabela 1 Zawartość istotnych plików dokumentacji dystrybuowanej z aplikacją Symfonia Handel

Plik	Zawartość
Ambashm.chm	Dokumentacja języka AmBasic
Amimpsp.chm	Opis języka szablonów importu specjalnego.
Handel - Dodatkowe polecenia języka raportów.pdf	Polecenia języka raportów nieudokumentowane dostatecznie w helpie.
Handel - Import specjalny.pdf	Opis importu specjalnego do Symfonia Handel.
Handel - Procedury języka raportów.pdf	Opis wybranych procedur języka raportów w Symfonia Handel.

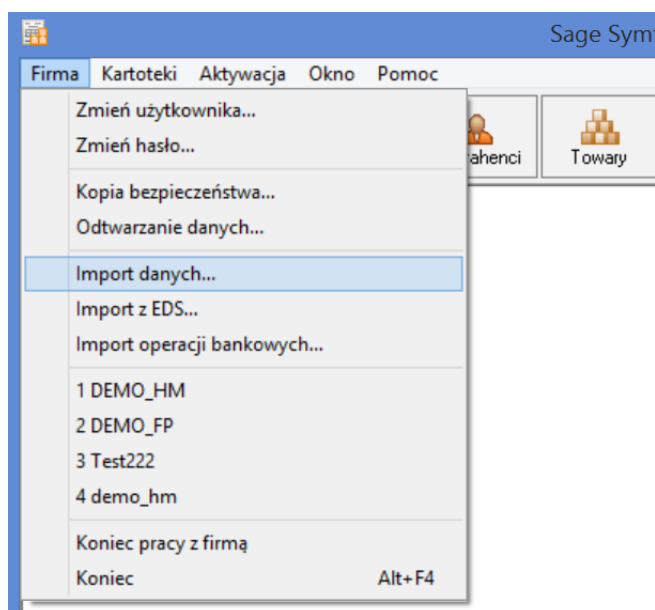
4 Praktyczny przykład rozwiązania

Mamy – dajmy na to – w chłodni wagę, która po każdym ważeniu przyjmowanego towaru generuje nam dokumenty (w formacie 3.0) i zapisuje je w pliku w określonej w lokalizacji (C:\WagaX\pzIn.txt). Chłodnia chce wiedzieć ile czego ma, więc generalnie dokumenty powinny mieć charakter 'Przyjęcie Zewnętrzne' (PZ). Nie wchodzimy tu w kwestię zakupu -> tym zajmuje się osobna komórka.

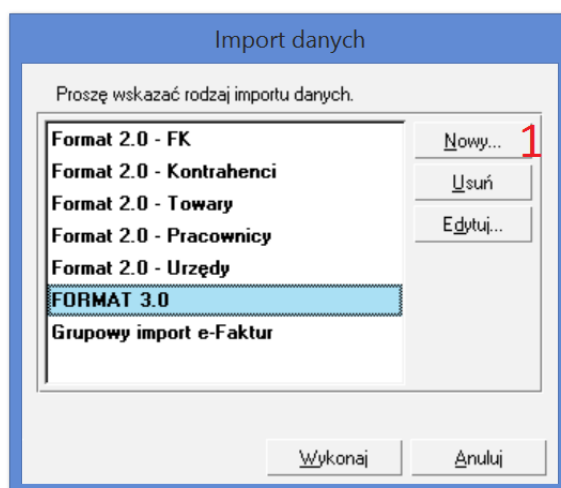
4.1 Przygotowanie importu danych

W pierwszym kroku zdefiniujemy sobie odpowiedni rodzaj importu przy pomocy którego będziemy importować do Symfonia Handel te dokumenty:

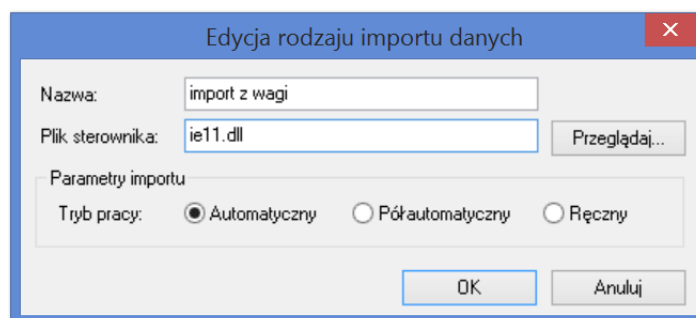
- Uruchamiamy konfigurację importu danych (Rysunek 1)
- Tworzymy nowy rodzaj importu danych (Rysunek 2)
- Określamy parametry importu danych (Rysunek 3)



Rysunek 1 Import danych w aplikacji Symfonia Handel – uruchomienie konfiguracji



Rysunek 2 Tworzenie nowego rodzaju import danych – cz. 1



Rysunek 3 Tworzenie nowego rodzaju import danych – cz. 2

W drugim kroku przygotowujemy odpowiednio spreparowany szablon importu. Żeby nie zaczynać od zera oprzemy się o szablon importu `hmin.AMS`, który jest dystrybuowany razem z aplikacją. Kopiujemy więc ten szablon do osobnego pliku (np. `hminWAG.AMS`) i pracujemy już na tej kopii. Biorąc pod uwagę co chcemy importować (dokumenty PZ) dokonujemy modyfikacji: w sekcji „Dokument” zamieniamy wiersze:

```
if ( !Def("RodzajDokumentu"))
{
    @action = Ask("W pliku wejściowym istnieją dane dokumentów.",
        "Będą one zaimportowane do bufora jednej z kartotek.",
        "",
        "Proszę wybrać kartotekę, do której mają trafić importowane dokumenty.",
        "{{icon=? buttons=[Dokumenty &sprzedaży:p] (Dokumenty &zakupu:r) (Dokumenty &magazynowe:i)}}")

    switch( @action )
    {
    case 4 // :p
        Def("RodzajDokumentu", "ZK")
        info("Import dokumentów do bufora kartoteki dokumentów zakupu.")
        break
    case 5 // :r
        Def("RodzajDokumentu", "MG")
        info("Import dokumentów do bufora kartoteki dokumentów magazynowych.")
        break
    case else // :i
        Def("RodzajDokumentu", "SP")
        info("Import dokumentów do bufora kartoteki dokumentów sprzedaży.")
        break
    }
}
```

na następujące:

```
if ( !Def("RodzajDokumentu"))
{
    Def("RodzajDokumentu", "MG")
}
```

Dzięki temu aplikacja nie będzie pytać się użytkownika, z jakim typem dokumentu ma do czynienia, tylko importować go jako dokument magazynowy. Oczywiście:

- można uzyskać ten sam efekt dodając w pliku importu pole 'RodzajDokumentu' o wartości „MG”
- można w dalszy sposób modyfikować szablon importu, choćby usunąć niepotrzebne sekcje (tj. np. „Pracownik”, „Urząd”, „Waluta”, ...)

Ale wybraliśmy akurat ten sposób.

Tak zdefiniowany rodzaj importu uruchamiamy (Rysunek 2, klawisz 'Wykonaj'). Sprawdzamy, czy dokument zaimportuje się poprawnie. Wykonujemy odpowiednie uzgodnienia i ten krok mamy gotowy.

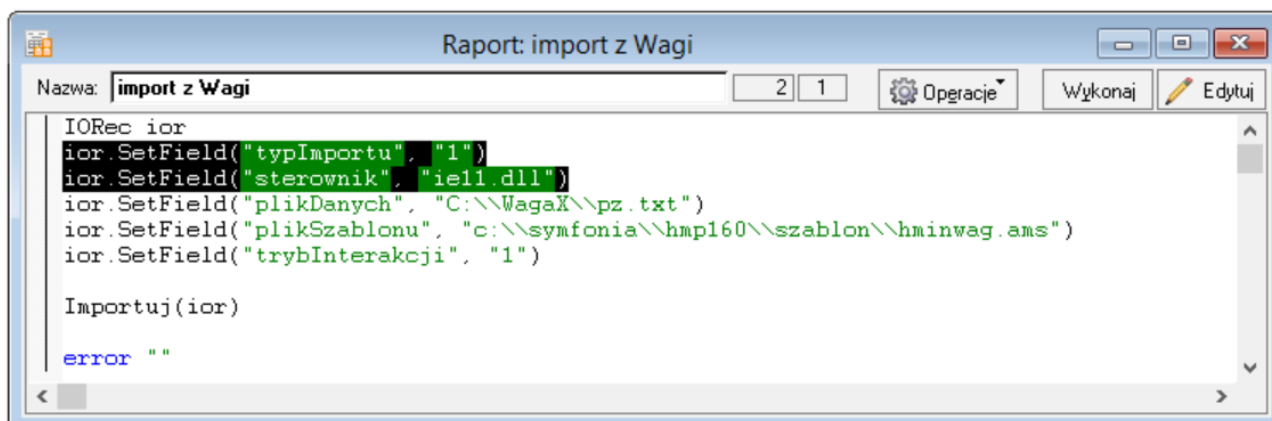
4.2 Uruchomienie importu z poziomu języka AmBasic

Aby uruchomić import z poziomu języka AmBasic skorzystamy z funkcji `long Importuj (IORec rec)`. Jest ona opisana w dokumencie „Handel – Dodatkowe polecenia języka raportów.pdf” w punkcie 2.3.5.

Umożliwia ona import na podstawie zdefiniowanego rodzaju importu (Rysunek 4), lub też na dynamiczne definiowanie parametrów importu (Rysunek 5).



Rysunek 4 Uruchomienie importu z poziomu AmBasic – na podstawie określonego rodzaju importu



Rysunek 5 Uruchomienie importu z poziomu AmBasic – bez tworzenia rodzaju importu

Skorzystamy z tej pierwszej możliwości. Kod poniżej:

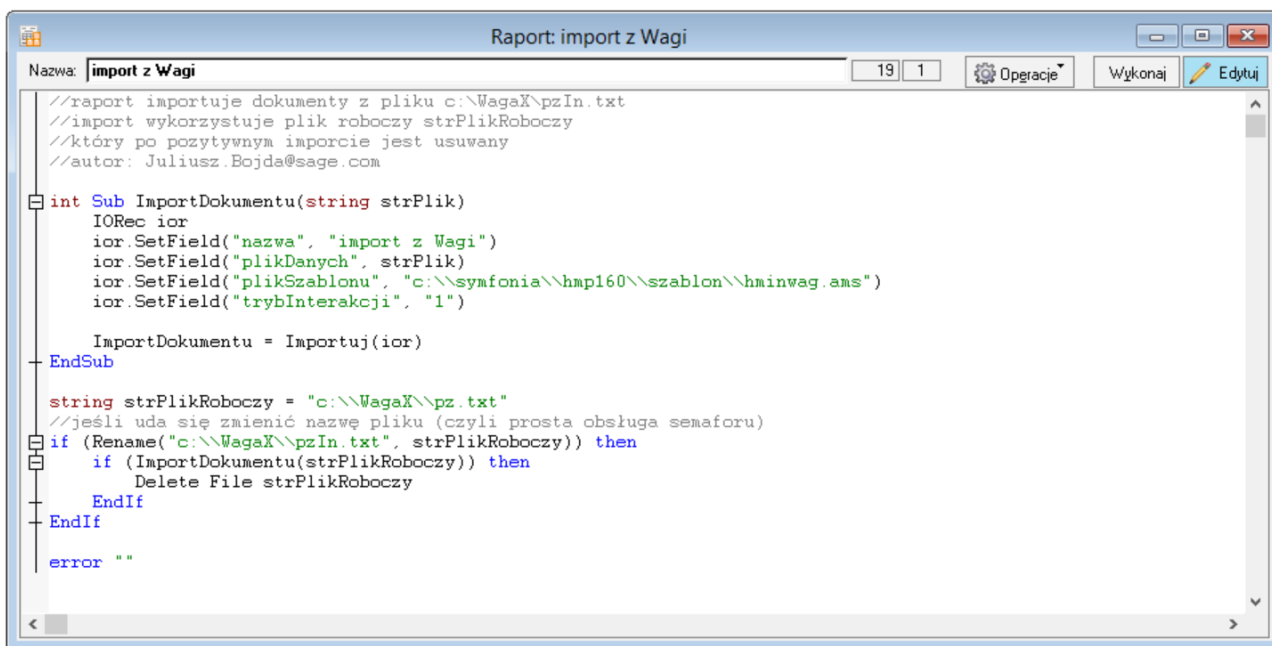
```
IORec ior
ior.SetField("nazwa", "import z Wagi")
ior.SetField("plikDanych", "c:\\WagaX\\pz.txt")
ior.SetField("plikSzablonu", "c:\\symfonia\\hmp160\\szablon\\hminwag.ams")
ior.SetField("trybInterakcji", "1")

Importuj(ior)

Error ""
```

Uruchamiamy raport => dokument z określonej lokalizacji powinien zostać zaimportowany do bufora kartoteki dokumentów magazynowych.

Jeśli wszystko zadziało się zgodnie z naszymi oczekiwaniami, przygotowujemy pełną wersję raportu, który nie tylko importuje dokument z określonego pliku, ale również robi dodatkowe rzeczy (np. usuwa ten plik, żeby w kółko nie importować tego samego dokumentu ☺). Przykładowy trochę bardziej rozbudowany raport przedstawia Rysunek 6. W tym przykładzie po pomyślnym zakończeniu importu usuwamy plik z danymi.



Rysunek 6 Import z pliku – przykład raportu z bardziej zaawansowaną logiką

```
//raport importuje dokumenty z pliku c:\WagaX\pzIn.txt
//import wykorzystuje plik roboczy strPlikRoboczy
//który po pozytywnym imporcie jest usuwany
//autor: Juliusz.Bojda@sage.com

int Sub ImportDokumentu(string strPlik)
    IORec ior
    ior.SetField("nazwa", "import z Wagi")
    ior.SetField("plikDanych", strPlik)
    ior.SetField("plikSzablonu", "c:\\symfonia\\hmp160\\szablon\\hminwag.ams")
    ior.SetField("trybInterakcji", "1")

    ImportDokumentu = Importuj(ior)
EndSub

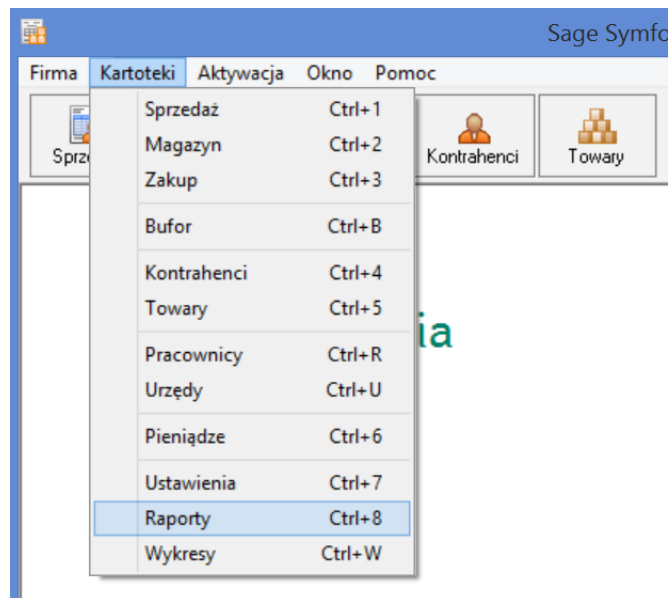
string strPlikRoboczy = "c:\\WagaX\\pz.txt"
//jeśli uda się zmienić nazwę pliku (czyli prosta obsługa semaforu)
if (Rename("c:\\WagaX\\pzIn.txt", strPlikRoboczy)) then
    if (ImportDokumentu(strPlikRoboczy)) then
        Delete File strPlikRoboczy
    EndIf
EndIf

error ""
```

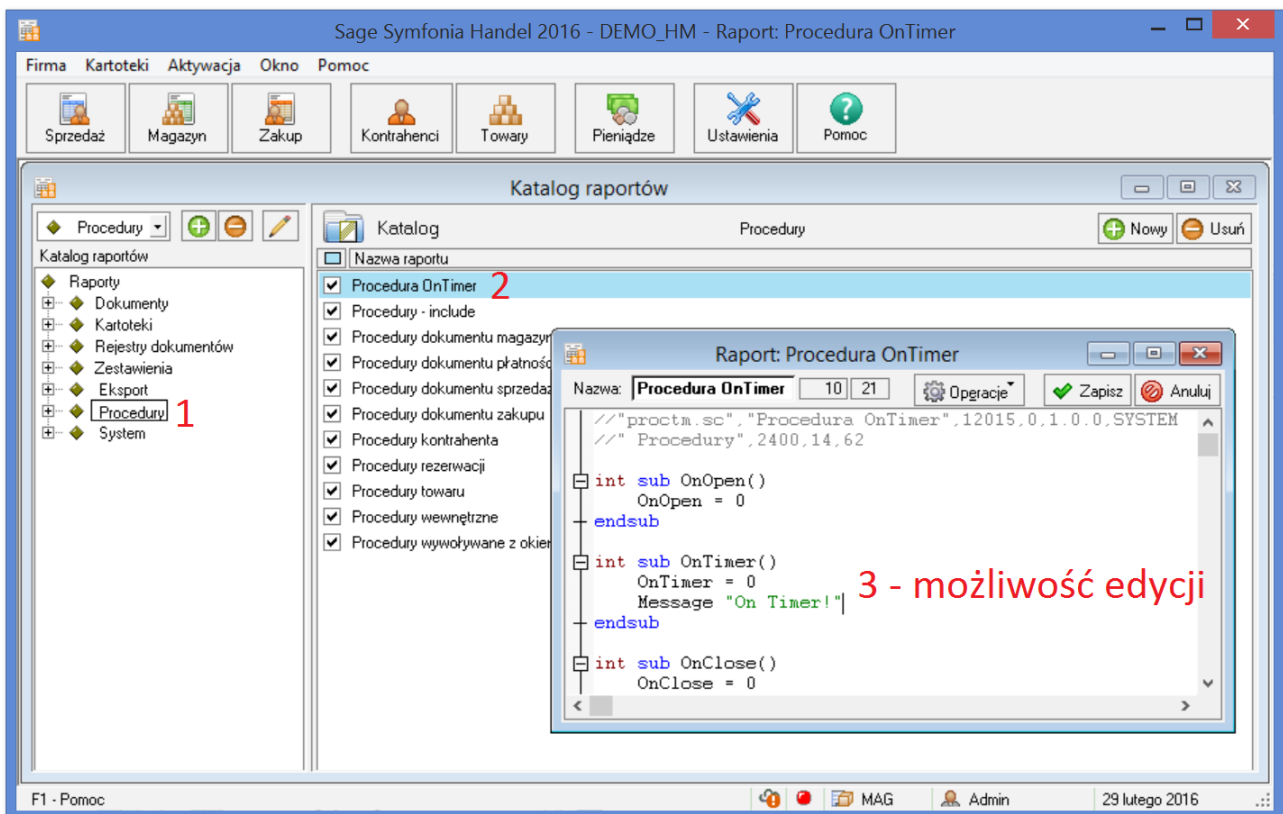
4.3 Cykliczne uruchamianie raportu z pomocą procedury OnTimer

Odpowiednio przygotowany raport (pkt 4.2, Rysunek 6) wystarczy teraz tylko cyklicznie uruchamiać a dane z określonej lokalizacji będą automatycznie łądownać w odpowiedniej kartotece dokumentów. Cykliczne uruchamianie można w prosty sposób osiągnąć korzystając z procedury OnTimer (opisana w dokumencie „Handel – Procedury języka raportów.pdf”, pkt 4.).

Zaczynamy od dostosowania/modyfikacji procedury OnTimer. Znajdziemy ją w kartotece raportów (Rysunek 7) w gałęzi Procedury w raporcie ‘Procedura OnTimer’ (Rysunek 8).

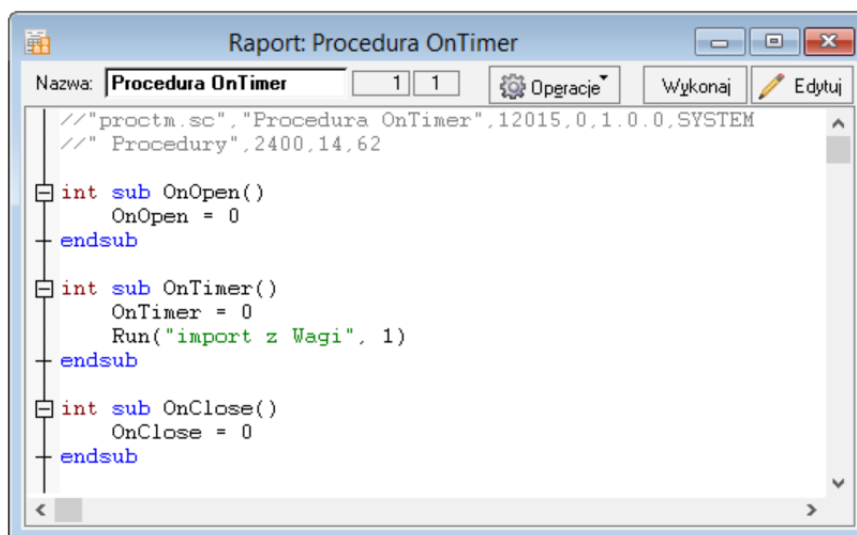


Rysunek 7 Uruchomienie kartoteki raportów



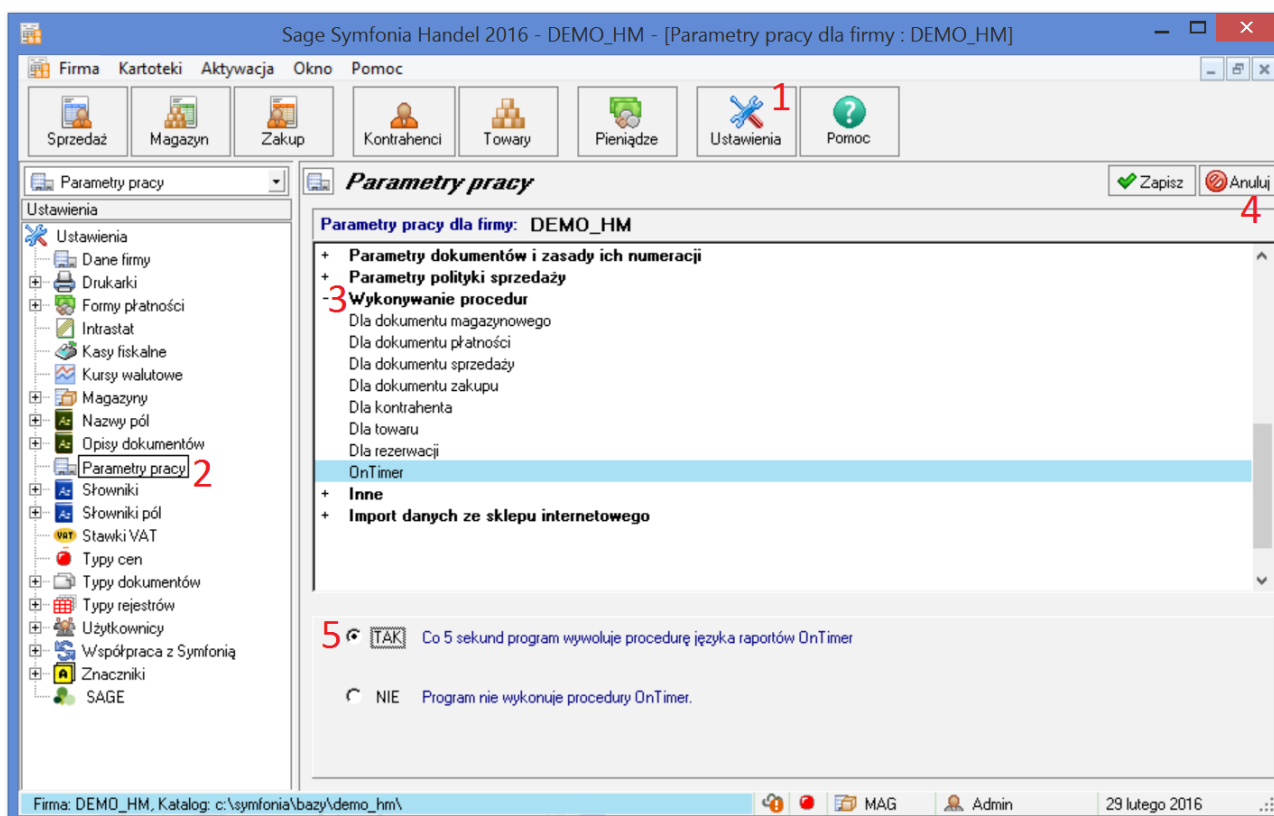
Rysunek 8 Lokalizacja raportu 'Procedura OnTimer'

W raporcie 'Procedura OnTimer' znajdujemy odpowiednią procedurę oraz odpowiednio modyfikujemy ją (dla powyższego przykładu wystarczy dopisanie linijki kodu, jak pokazuje Rysunek 9).



Rysunek 9 Przykładowa modyfikacja procedury OnTimer

Pozostaje już tylko włączenie mechanizmu cyklicznego wykonywania procedury OnTimer (domyślnie jest wyłączona). Sposób włączenia przedstawia Rysunek 10.



Rysunek 10 Włączenie cyklicznego wykonywania raportu OnTimer

5 Uwagi praktyczne

Dobrze jest:

- Pierwszy import przeprowadzić ręcznie
Dzięki temu w razie potrzeby utworzą się wszystkie potrzebne uzgodnienia itp.. Będzie też można zweryfikować w praktyce, czy raport wykonuje się prawidłowo.
- Sprawdzić, czy raport jest uruchamiany:
 - przez uprawnionego użytkownika
 - na odpowiedniej maszynie

- Dorobić obsługę błędów i sytuacji wyjątkowych.
W szczególności co najmniej logować operacje i ich wynik.
- Dobrać częstotliwość wykonywania raportu do potrzeb.
Można to zrobić zarówno poprzez odpowiedni wpis w pliku .ini, jak również poprzez odpowiednią konstrukcję samej procedury OnTimer (procedura może być wykorzystywana do wielu celów!).
- Dostosować szablon importu do potrzeb.
Szablony domyślnie dystrybuowane z aplikacją są ogólnego przeznaczenia, w przypadku więc zastosowania specjalistycznego będą zawierały nadmiarowe informacje. Nie powinny na pewno zawierać elementów mogących prowadzić do interakcji z użytkownikiem, zaś na ogół niektóre wartości można w nich określić na stałe.

6 Przykładowe modyfikacje

W/w rozwiązanie, to tylko przykład. W rzeczywistości należy uwzględnić dodatkowe uwarunkowania, które wpłyną na rozwiązanie, np.:

- zmiana przyjętych założeń
 - inny format plików wejściowych
 - dynamicznie określana lokalizacja, niekoniecznie plikowa
 - nieznany lub niedookreślony rodzaj dokumentu
 - potrzeba ręcznych modyfikacji lub wręcz przeciwnie: potrzeba automatycznego wystawienia
- Uprawnienia użytkowników
- Określony wyzwalacz procesu importu (np. konkretna godzina lub zdarzenie)
- ...

Potrzeby użytkowników są szerokie, natomiast najważniejsze, że możliwości realizacji są...

7 Spis rysunków

Rysunek 1 Import danych w aplikacji Symfonia Handel – uruchomienie konfiguracji.....	3
Rysunek 2 Tworzenie nowego rodzaju import danych – cz. 1	3
Rysunek 3 Tworzenie nowego rodzaju import danych – cz. 2	4
Rysunek 4 Uruchomienie importu z poziomu AmBasic – na podstawie określonego rodzaju importu	5
Rysunek 5 Uruchomienie importu z poziomu AmBasic – bez tworzenia rodzaju importu	5
Rysunek 6 Import z pliku – przykład raportu z bardziej zaawansowaną logiką	6
Rysunek 7 Uruchomienie kartoteki raportów.....	7
Rysunek 8 Lokalizacja raportu 'Procedura OnTimer'	7
Rysunek 9 Przykładowa modyfikacja procedury OnTimer	8
Rysunek 10 Włączenie cyklicznego wykonywania raportu OnTimer	8

8 Spis tabel

Tabela 1 Zawartość istotnych plików dokumentacji dystrybuowanej z aplikacją Symfonia Handel.....	2
--	---