

[Przewodnik wdrożeniowca](#) > Integracja z Forte HM przy pomocy procedury OnTimer

Menu

- [Wprowadzenie](#)
- [Koncepcja działania Integracji](#)
- [Szczegółowe omówienie Integracji na przykładzie](#)
 - [Tworzenie skryptu VBA realizującego zadania importu](#)
 - [Umieszczanie skryptu VBA w dokumentach typu SBC](#)
 - [Przygotowanie procedury "Generowanie zamówienia obcego z eDokumentów"](#)
 - [Konfiguracja procedury OnTimer\(\)](#)

Niniejszy artykuł jest częścią opisu dostępnego pod adresem [Integracja z Forte](#)

Wprowadzenie

System Forte HM oferuje wiele możliwości wykonania procedur, które można wykorzystać w procesie integracji. Są to:

- język raportów **AmBasic** - język skryptowy własny dostępny w Handlu Forte, który umieszczamy bezpośrednio w raporcie, jest kombinacją języka VB oraz C, nie posiada dobrej dokumentacji (ta co jest, to załączam do niniejszej instrukcji).
- język **VBA**, czyli Visual Basic For Applications - produkt wykorzystywany przez Microsoft m.in. w produktach pakietu Office. Jest dość okrojona wersja Visual Basic-a, ale do tych potrzeb jest po prostu wystarczająca.

Przejdź do [Menu](#)

Koncepcja działania, czyli jak wykonać taki moduł

Zasada działania mechanizmu integracji polega na:

- Stworzeniu skryptu w AmBasic-u lub VBA realizującego zadania importu
- Stworzenie Dokumentu SBC
- Konfiguracji raportu *Procedura OnTimer*

Przejdź do [Menu](#)

Szczegółowe omówienie integracji na przykładzie generowania dokumentu Zamówień Obcych

Poniższa część jest szczegółowym opisem **importu** do Forte HM dokumentu typu Zamówienie Obce ZO. Polega to na tym, że wygenerowany dokument przez akcję procedury workflow w systemie eDokumenty jest wczytywany przez opisywany mechanizm procedury Forte HM z katalogu na dysku. Po wczytaniu pliku odpowiednie skrypty generują dokument w Forte Handel. Dokument ten jest w buforze, do wystawienia.

Tworzenie skryptu VBA realizującego zadania importu

Poniżej przedstawiono skrypt VBA, który:

- Przechodzi do zadeklarowanego katalogu na dysku, przeszukuje go w poszukiwaniu plików do importu
- Po znalezieniu generuje sobie listę plików tekstowych do zaimportowania.
- Skrypt przechodzi po każdym pliku, otwierając go. Następnie linia po linii odczytuje go i generuje schemat dokumentu Zapotrzebowania Obcego.
- Po utworzeniu schematu metodą obiektu *!BIORec* generuje dokument w Forte HM (ImzportZO)
- Końcowymi działaniami skryptu jest zamknięcie pliku i przesunięcie go do katalogu kopii bezpieczeństwa zaimportowanych plików.

Format pliku exportu (zapisany jako .txt)

```
<KOD DOSTAWCY>
<OPIS DOKUMENTU>
<SYMBOL TOWARU Z POZ.1>
<ILOŚĆ TOWARU Z POZ.1>
<CENA NETTO TOWARU Z POZ.1>
<SYMBOL TOWARU Z POZ.2>
```

```

<ILOŚĆ TOWARU Z POZ.2>
<CENA NETTO TOWARU Z POZ.2>
...
<SYMBOL TOWARU Z POZ.n>
<ILOŚĆ TOWARU Z POZ.n>
<CENA NETTO TOWARU Z POZ.n>

```

Przykładowy skrypt, który importuje pliki tekstowe zawierające dane z powyższego schematu:

```

Private Sub Document_Close(ByVal EvType As AmEventState, Cancel As Boolean)
    Element.CloseDoc (True)
End Sub

Private Sub CloseDocument()
    Element.CloseDoc (True)
End Sub

Private Sub Document_Open()
    Element.Visible = False
    GenerateZO
    CloseDocument
End Sub

Private Sub GenerateZO()
    ' Deklaracje zmiennych
    Dim oFactory As Object
    Dim oRec As Object 'Rekord danych
    Dim nError

    Dim sDocumentType As String ' Typ dokumentu
    Dim sDesc As String ' Opis dokumentu
    Dim sClientCode As String 'Nazwa kontrahenta
    Dim sProductCode As String 'Kod produktu
    Dim sQuantity As String ' Ilość
    Dim sPrice As String ' Cena netto

    Dim sPath As String ' ścieżka katalogu z plikami wyeksportowanymi z eDokumentów
    Dim sBkpPath As String ' ścieżka katalogu, do którego odczytane i przetworzone pliki
        ' zostaną przeniesione po zakończonym procesie
    Dim sName As String ' nazwa pliku
    Dim sFileName As String ' kompletna ścieżka do pliku
    Dim iFileNum As Integer ' znacznik pliku do operacji Input, Line Input
    Dim sBuf As String ' Bofor stringu odczytanego z pliku
    Dim iCounter As Integer ' Licznik przebiegu po liniach pliku
    Dim iNumber As Integer ' Licznik przebiegu po liniach pozycji produktu w pliku
    Dim oFileSystem As Object ' Obiekt do operacji na plikach

    Set oFileSystem = CreateObject("Scripting.FileSystemObject")

    sPath = "C:\eDokumentyXML\export\ZO\"
    sBkpPath = "C:\eDokumentyXML\export\ZO\sent\"
    sName = Dir$(sPath & "*.txt") ' Pobranie listy plików w katalogu

    Do While sName <> "" ' Jeżeli/dopóki istnieją pliki w katalogu to wykonaj dalsze operacje
        sFileName = sPath & sName ' Ustawienie pełnej ścieżki dokumentu
        iFileNum = FreeFile()

        iCounter = 1 ' Licznik przebiegu po liniach pliku
        iNumber = 0 ' Licznik przebiegu po pozycjach dokumentu.
            ' Dopuszczalne wartości dla jednej pozycji dokumentu to 0-2

```

```

        ' 0 - symbol produktu
        ' 1 - ilość
        ' 2 - cena netto
Open sFileName For Input As iFileNum

' Tworzenie obiektów deklaracji danych
Set oFactory = Application.PropertyAp("HFactory")
' Rekord danych, który zostanie wstawiony do systemu
Set oRec = oFactory.NewObject("BIORec")

sDocumentType = "ZMO" ' Typ dokumentu: Zamówienie obce
nError = oRec.SetField("typDk", Trim(sDocumentType)) ' Ustawienie typu dokumentu

Do While Not EOF(iFileNum)
    Line Input #iFileNum, sBuf

    Select Case iCounter
        Case Is = 1 ' Wywołanie identyfikatora kontrahenta

            sClientCode = "" & Trim(sBuf)
            nError = oRec.BeginSection("daneKh")
            nError = oRec.SetField("KhKod", Trim(sClientCode))
            nError = oRec.EndSection()
        Case Is = 2 ' Opis dokumentu
            sDesc = Trim(sBuf)
            nError = oRec.SetField("opis", Trim(sDesc))
        Case Is > 2 ' Pozycje dokumentu

            Select Case iNumber
                Case Is = 0 ' Kod produktu
                    sProductCode = Trim(sBuf)
                    nError = oRec.BeginSection("Pozycja dokumentu")
                    nError = oRec.SetField("kod", Trim(sProductCode))

                Case Is = 1 ' Ilość
                    sQuantity = Trim(sBuf)
                    nError = oRec.SetField("ilosc", Trim(sQuantity))

                Case Is = 2 ' Cena netto
                    sPrice = Trim(sBuf)
                    nError = oRec.SetField("cena", Trim(sPrice))
                    nError = oRec.EndSection()

            End Select

            iNumber = iNumber + 1 ' Inkrementacja licznika
            If iNumber > 2 Then
                iNumber = 0 ' Resetowanie wartości licznika
            End If

        End Select

        iCounter = iCounter + 1

Loop

Dim oFunction As Object
Set oFunction = oFactory.NewObject("BFunkcja")

```

```

' Wywołanie generowania dokumentu
Dim DocumentID As Long
DocumentID = oFunction.ImportZO(oRec) ' GENEROWANIE DOKUMENTU ZAMÓWIENIA OBCEGO
Close iFileNum ' Zamykanie pliku

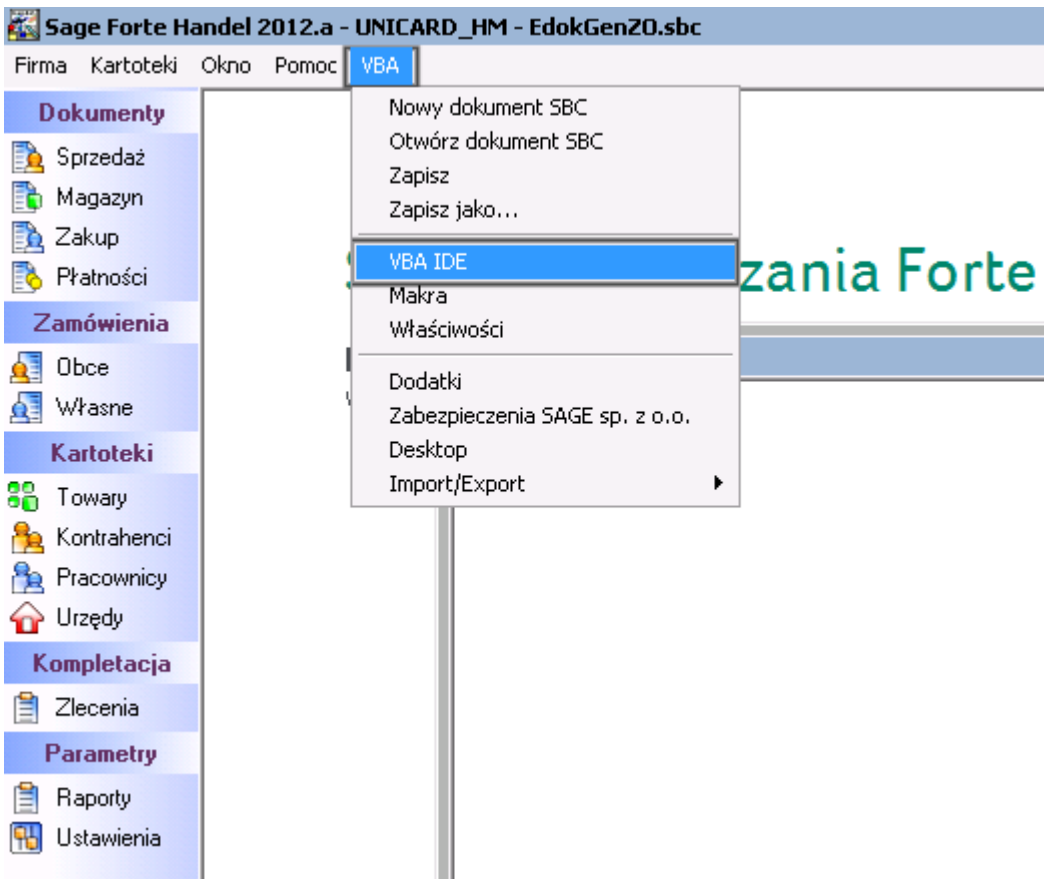
oFileSystem.MoveFile sFileName, sBkpPath & sName ' Przeniesienie pliku do katalogu bkp

sName = Dir()
Loop
End Sub

```

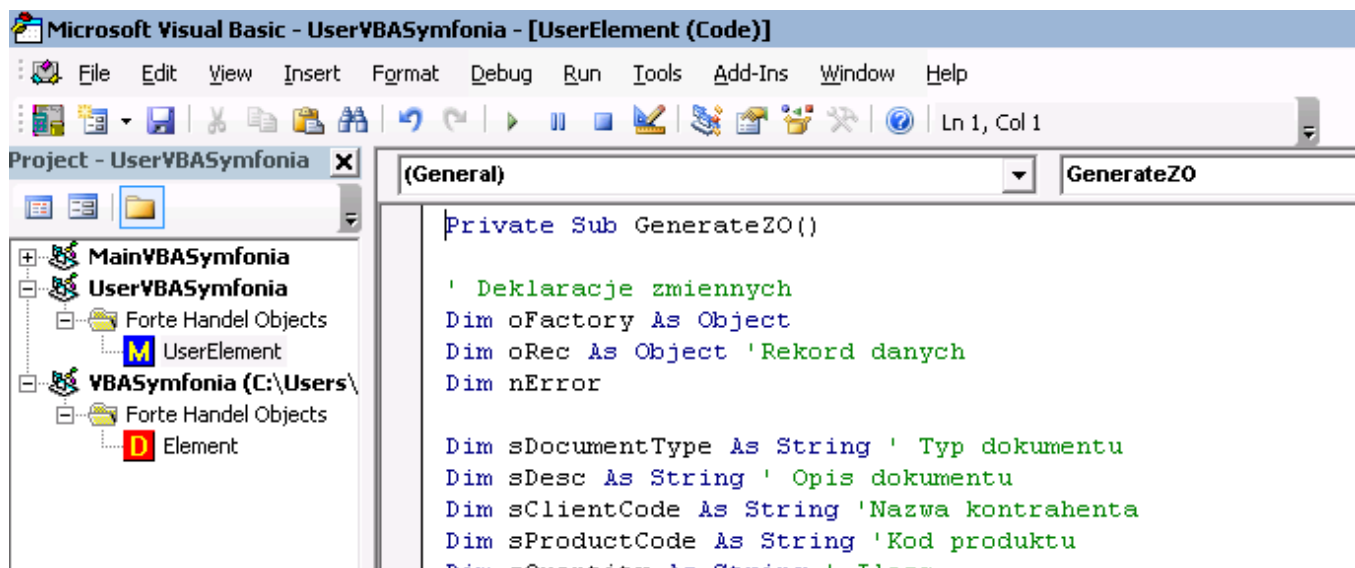
Jak przygotować kod skryptu?

Bardzo prosto. Należy wykorzystać środowisko VBA dostępne w Forte HM (sugerowane do testowania) lub w ostateczności takie samo środowisko pakietu Microsoft Office. W tym przykładzie skorzystamy ze środowiska zawartego w Forte HM. Aby je uruchomić należy w menu głównym (z odpowiednimi uprawnieniami) przejść do menu VBA:



Uruchamianie środowiska VBA w Forte HM

Po pojawieniu się okna wprowadzany kod możemy testować w module UserElement. Uwaga, obiekt ten nie przechowuje trwale zawartości, która kasuje się po zamknięciu systemu Forte HM.



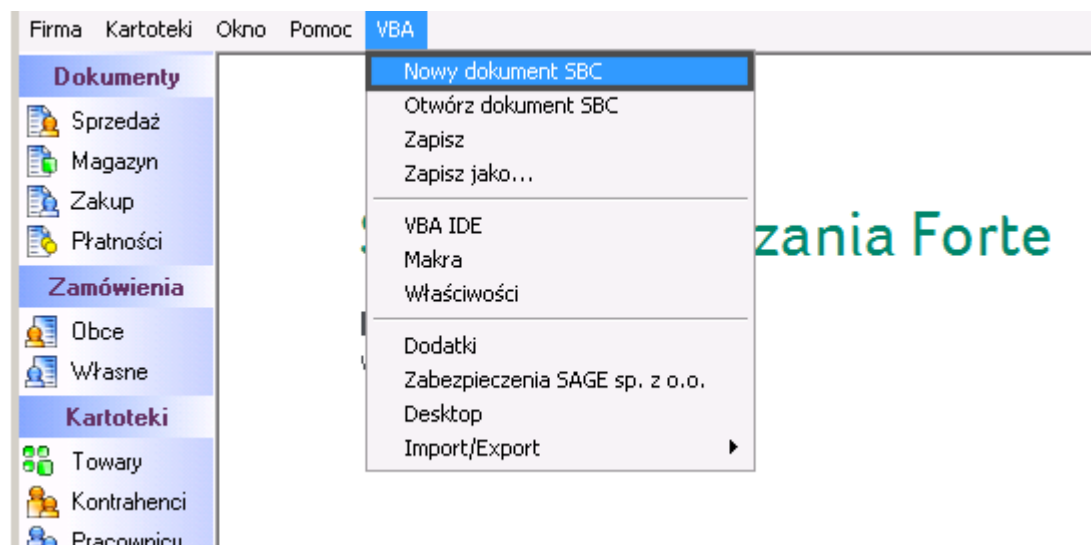
Środowisko VBA w Forte HM

Taki skrypt musi być umieszczony, aby był widoczny przez Forte HM. Do tego celu wykorzystamy inny mechanizm, który nazywa się Dokument SBC.

Umieszczanie skryptu VBA w dokumentach typu SBC

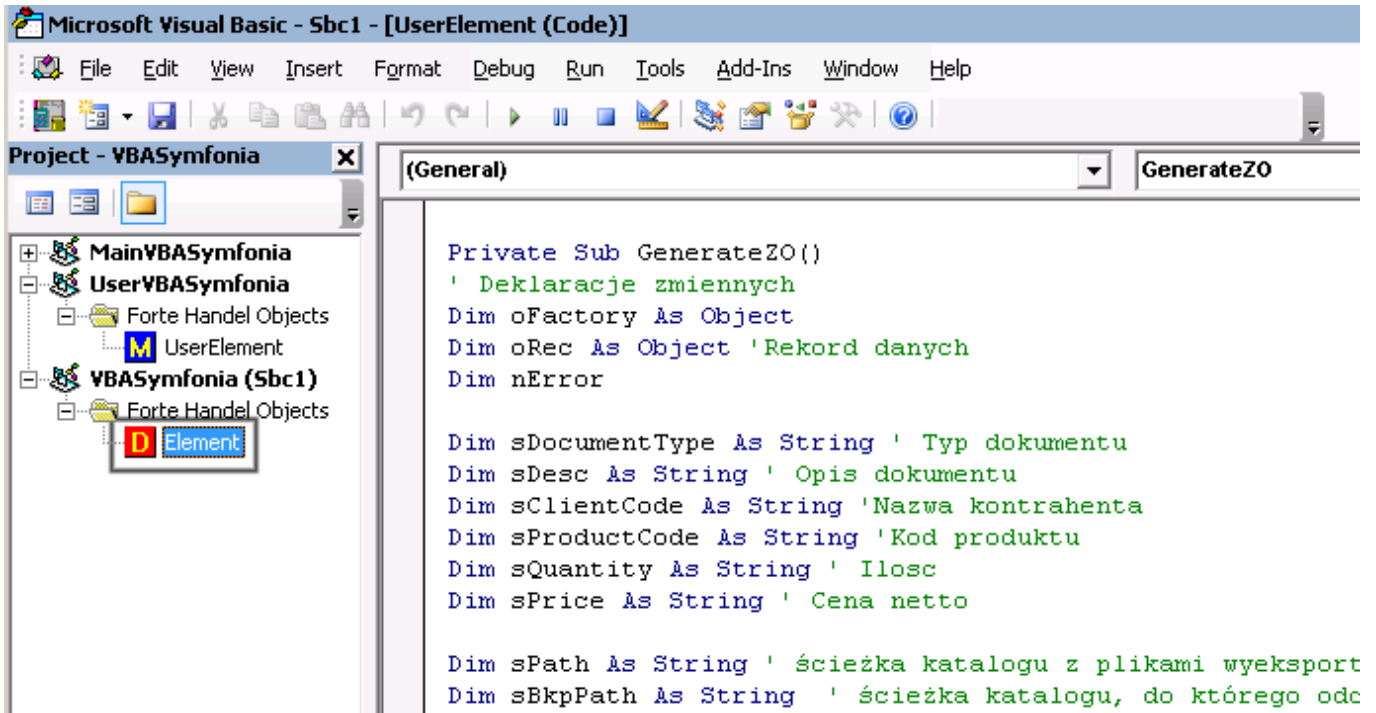
Dokument SBC stanowi on dodatkowy moduł/okno/dokument tworzony pod potrzeby użytkownika, w którym można zamieścić dostępne kontrolki (przyciski, pola, raporty, itp.). Dla potrzeb integracji nie ma wymagań, aby jakiegokolwiek elementy znajdowały się na formatce dokumentu. Najistotniejsza jest jego zawartość.

W menu VBA wybieramy nowy dokument SBC. Najpierw otwiera się okno układu formatki, co dla nas jest nie istotne. Najlepiej jest wybrać pierwszą opcję, czyli puste okienko. Dokument ten będzie głównym obiektem uruchamianym z poziomu raportu. Mając otwartą formatkę dokumentu otwieramy z menu VBA > VBA IDE,



Tworzenie nowego dokumentu SBC

a następnie na liście eksploratora projektu wybieramy element z projektu !VBASymfonia (Sbc1) i wklejamy utworzony kod.



Edycja dokumentu SBC

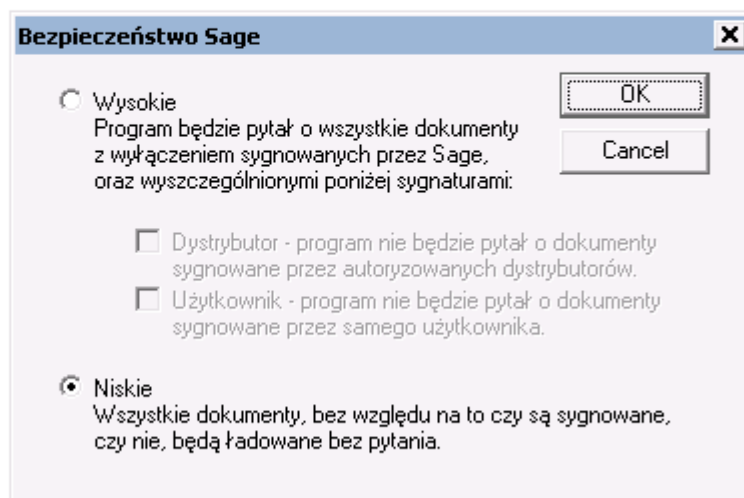
Po wprowadzeniu kodu przechodzimy z powrotem do okna Forte HM i w menu VBA klikamy *Zapisz* lub *Zapisz jako...* Otworzy się okno wyboru miejsca zapisu pliku. Nadajemy nazwę i klikamy zapisz. Okienko można zamknąć. Tworzy się dokument SBC, który następnie wywołamy w raporcie.

W tym momencie nie jest jeszcze możliwym wykonanie kodu zawartego w utworzonym dokumencie. Do tego celu należy zmienić **uprawnienia** do wykonywania skryptów. Osiągniemy to wchodząc do menu *VBA > Zabezpieczenia Sage Sp. z o.o.* . Otworzy się okienko, w którym ustawiamy poziom zabezpieczeń na niskie:

System Zarządzania Forte

Handel

Wersja 2012.a



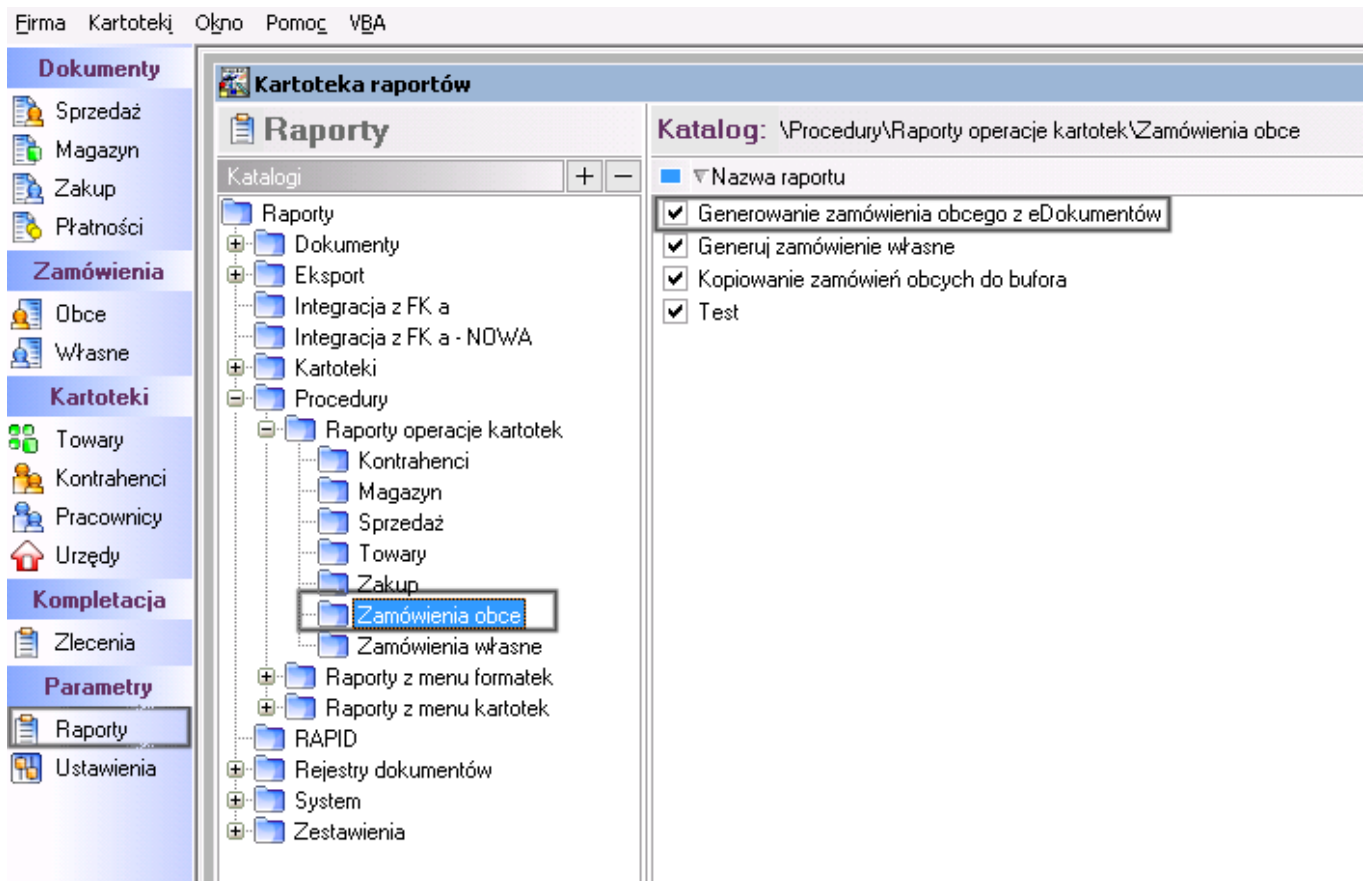
Ustawianie zabezpieczeń do wykonywania skryptów SBC

Jednym z ostatnich etapów importu danych z plików jest utworzenie raportu, który będzie uruchamiany przez OnTimer, i który będzie wywoływał program zapisany w dokumencie SBC.

Przejdź do [Menu](#)

Przygotowanie procedury "Generowanie zamówienia obcego z eDokumentów"

Tworzenie odpowiedniej procedury należy rozpocząć w oknie Forte HM wchodząc w menu raporty, następnie w otwartym oknie *Raporty* przejść do katalogu *Raporty > Procedury > Raporty operacje kartotek > Zamówienia Obce*. W prawym panelu klikamy przycisk z lewego górnego rogu *Nowy raport*.



Ustawianie zabezpieczeń do wykonywania skryptów SBC

Tworzy się formatka nowego raportu, w który prowadzimy kod:

```
// Definicja z raporcie Generowanie zamówienia obcego z eDokumentów (język amBasic)
// Raport generujący dokument Zamówienia Obcego
// Dane pochodzą z pliku wygenerowanego w eDokumentach
// Autor: Jacek Achtelik
// Copyright: BetaSoft 2012

Dispatch App, Doc // deklarowanie obiektów

App = GetApplication() //ustawienie stanu obiektu
// wywołanie uruchomienia dokumentu SBC
Doc = App.Documents.OpenDocument("C:\\Users\\ztestowy\\documents\\Sage\\EdokGenZ0.sbc")
```

Po zapisaniu takiego raportu (możemy dla testów wykonać) należy skonfigurować procedurę OnTimer.

Przejdź do [Menu](#)

Konfiguracja procedury OnTimer()

Procedura OnTimer() jest specyficznego rodzaju raportem, który systemowo posiada możliwość jego wywołania w zadanym interwale czasu. Domyślną wartością jest 5 sek, lecz może ona być dostosowana w plikach konfiguracyjnych.

Konfiguracja OnTimer()

- Mając odpowiednie uprawnienia administratora w systemie Forte HM Z lewego panelu wybieramy menu Ustawienia.
- Po otwarciu się w prawym panelu okna z danymi Ustawień przechodzimy do *Ustawienia > Firma > Parametry pracy*.
- W prawym panelu przechodzimy do *Wykonywanie procedur > OnTimer*.
- Po zaznaczeniu OnTimer wybieramy edytuj i wybieramy TAK, a następnie zapisujemy.

Mając już włączony mechanizm wykonywania raportu OnTimer musimy wykonać raport. Raport jest opcjonalny, gdyż wszystkie niezbędne instrukcje można oczywiście ująć w OnTimerze. Aby skonfigurowana procedura OnTimer() wykonała dla nas konkretną czynność należy zmienić jej kod. W tym celu odszukujemy w drzewku *Raporty > Procedury* raport *Procedura OnTimer*. Edytując ją wprowadzamy kod:

```
#text/x-vba
int sub OnOpen()

    OnOpen = 0
endsub

int sub OnTimer()

    If xFactory.user == "b2b" then // użytkownik, który stale ma uruchomioną instancję Forte HM.

        //Wywołanie raportu generującego dokumenty zamówień obcych z eDokumentów
        Run ("Generowanie zamówienia obcego z eDokumentów")
        //      Message "OnTimer()"
    EndIf

    OnTimer = 0

endsub

int sub OnClose()

    OnClose = 0
endsub
```

Uwaga! Jak już wspomniano powyżej aby OnTimer zadziałał, musi być uruchomiona instancja Forte HM na wybranym użytkowniku. Użytkownik b2b jest systemowym użytkownikiem b2b, który jest przeznaczony do tych zadań.