**Wikiprint Book**

**Title: Przewodnik wdrożeniowca > Integracja z Forte HM przy …**

**Subject: Archiwum - eDokumenty - elektroniczny system obiegu dokumentów, workflow i CRM - DeployerGuide/Customization/Forte/ForteOnTimer**

**Version: 19**

**Date: 07/21/26 22:27:35**

## Table of Contents

*Przewodnik wdrożeniowca > Integracja z Forte HM przy pomocy procedury OnTimer* 3
*Menu* 3
*Wprowadzenie* 3
*Koncepcja działania, czyli jak wykonać taki moduł* 3
*Szczegółowe omówienie integracji* 3
*Tworzenie skryptu VBA realizującego zadania importu* 3
*Umieszczanie skryptu VBA w dokumentach typu SBC* 6
*Włączanie działania OnTimer()* 7
*Informacje o możliwościach wykorzystania procedury OnTimer()* 7

# Przewodnik wdrożeniowca > Integracja z Forte HM przy pomocy procedury OnTimer

**Menu**


- Wprowadzenie
- Koncepcja działania Integracji
- Szczegółowe omówienie Integracji na przykładzie


Niniejszy artykuł jest częścią opisu dostępnego pod adresem Integracja z Forte

## Wprowadzenie

Dla celów integracji z systemem można wykorzystać mechanizm procedur dostępny w systemie Forte (HM). Procedura jest to pewnego rodzaju raport, który napisany jest w języku stworzony przez Sage Symfonię: *AmBasic*. Jedną z takich procedur jest raport **OnTimer()**, który może być uruchamiany w określonych interwałach czasu.

*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

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. Nastepnie 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.

Przykładowy skrypt:

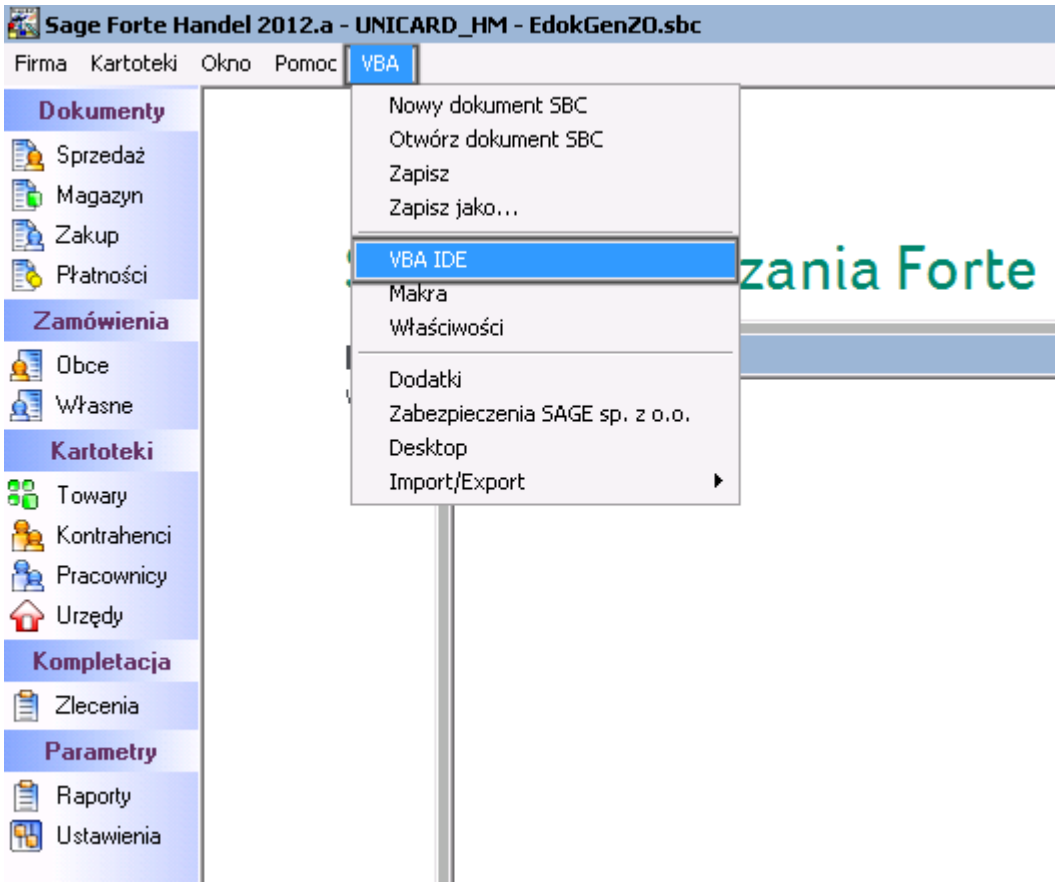
```
Private Sub GenerateZO()


' 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 ' Ilosc
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:\Users\ztestowy\dane\"
sBkpPath = "C:\Users\ztestowy\dane\done\"
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")
       Set oRec = oFactory.NewObject("BIORec") ' Rekord danych, który zostanie wstawiony do systemu

       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
```

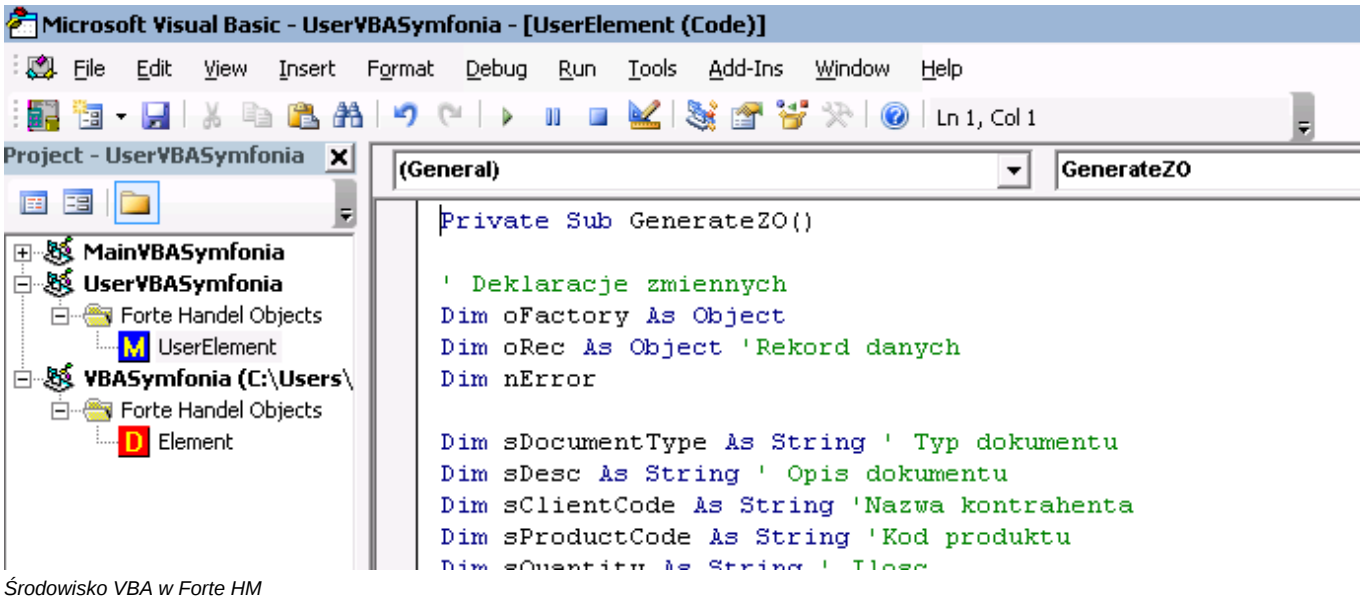
Jak przygotować kod skryptu?

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 umieścić, 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 jakiekolwiek elementy znajdowały się na formatce dokumentu. Najistotniejsza jest jego zawartość.

W mnu VBA wybieramy nowy dokument SBC. Otwiera się puste okienko, które będzie stanowić dla nas bazę do dalszych działań.

Należy pamiętać o uprawnieniach do wykonywania skryptów SBC

*Przejdź do Menu*

### Włączanie działania 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.

### Informacje o możliwościach wykorzystania procedury OnTimer()

Mając już włączony mechanizm wykonywania raportu OnTimer musimy wykonać raport. Raport jest opcjonalny, gdyż wszystkie niezbędne instrukcje można oczywiście ująć w OnTimerze.

HMF ofertuje wiele możliwości wykonania procedur. 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 rewelacyjna.

Na potrzeby integracji sugeruje się wykorzystać język VBA. Import danych do Forte HM (na przykładzie generowania dokumentu **Zamówienia Obcego** przedstawiam poniżej.

1. Aby wywołać powyższy skrypt dodajemy w tym samym module metodę: