

Modelowanie procesów za pomocą C# – wprowadzenie

Informacje ogólne

Uwaga

W wersji 2020.0.0 Comarch DMS mechanizm modelowania procesów metodą C# jest wersją beta.

Od wersji 2021.1.0 modelowanie procesów w języku C# jest objęte licencją Premium.

Od wersji 2023.1.0 za pomocą C# można tworzyć skrypty do pracy automatycznej (zob. [Konfiguracja automatycznego trybu pracy \(robotyzacja procesów\)](#)).

Comarch DMS został wyposażony w mechanizm wspomagający modelowanie procesów oparty o kompilator C#. Wersja kompilatora jak również wersja języka C# jest uzależniona od Microsoft .NET Framework. Zalecana jest aktualizacja Microsoft .NET Framework do najnowszej dostępnej wersji.

Wskazówka

Kod C# ma zastosowanie w Comarch DMS m. in. przy **weryfikacji kontrahenta na białej liście podatników** (zob. [Weryfikacja kontrahenta na białej liście podatników](#))

Środowisko kompilacji C#

Wynikowymi bibliotekami kompilatora Comarch DMS są zawsze biblioteki **dll**.

Po skompilowaniu, biblioteki są automatycznie konfigurowane do pracy na serwerze IIS. Dzięki temu funkcjonalność jest

dostępna dla wszystkich platform (aplikacji stacjonarnej, webowej, mobilnych iOS i Android).

Miejsca przechowywania bibliotek:

1. **{Apl. www (serwerowa)}\bin\.scriptcs_cache** – katalog do którego są kopiowane biblioteki po kompilacji kodu C#.
2. **{Apl. www(serwerowa)}\bin\Scriptcs_bin** – miejsce na biblioteki dll używane przez kod C# jako referencje.

Przed kompilacją skryptów C# na komputerze z IIS należy **nadać użytkownikowi IIS_IUSRS prawo zapisu do katalogu .scriptcs_cache**

Okno kompilacji języka C# jest dostępne z poziomu właściwości wybranych kontroltek. Kontrolki obsługują poniższe zdarzenia:

1. Init (zdarzenie OnInit)
2. Obserwator (zdarzenie OnChange)
3. Wciśnięcie przycisku (zdarzenie OnClick)

Jeżeli zaistnieje potrzeba skorzystania z biblioteki referencyjnej w której znajdują się funkcje do wywołania w kodzie C#, bibliotekę należy wgrać do katalogu **bin\Scriptcs_bin**. Nazwę tej biblioteki należy podać w oknie „Referencje” (zob. poniżej punkt *Język C# na definicjach kontroltek*).

Istnieje możliwość skorzystania z referencji bibliotek com (.NET) zarejestrowanych na serwerze IIS. W takim przypadku, w oknie „Referencje” należy wpisać nazwę biblioteki bez rozszerzenia dll.

Kompilacja kodu C# odbywa się po naciśnięciu ikony  lub kliknięciu przycisku „Kompiluj i zapisz”. Elementy te znajdują

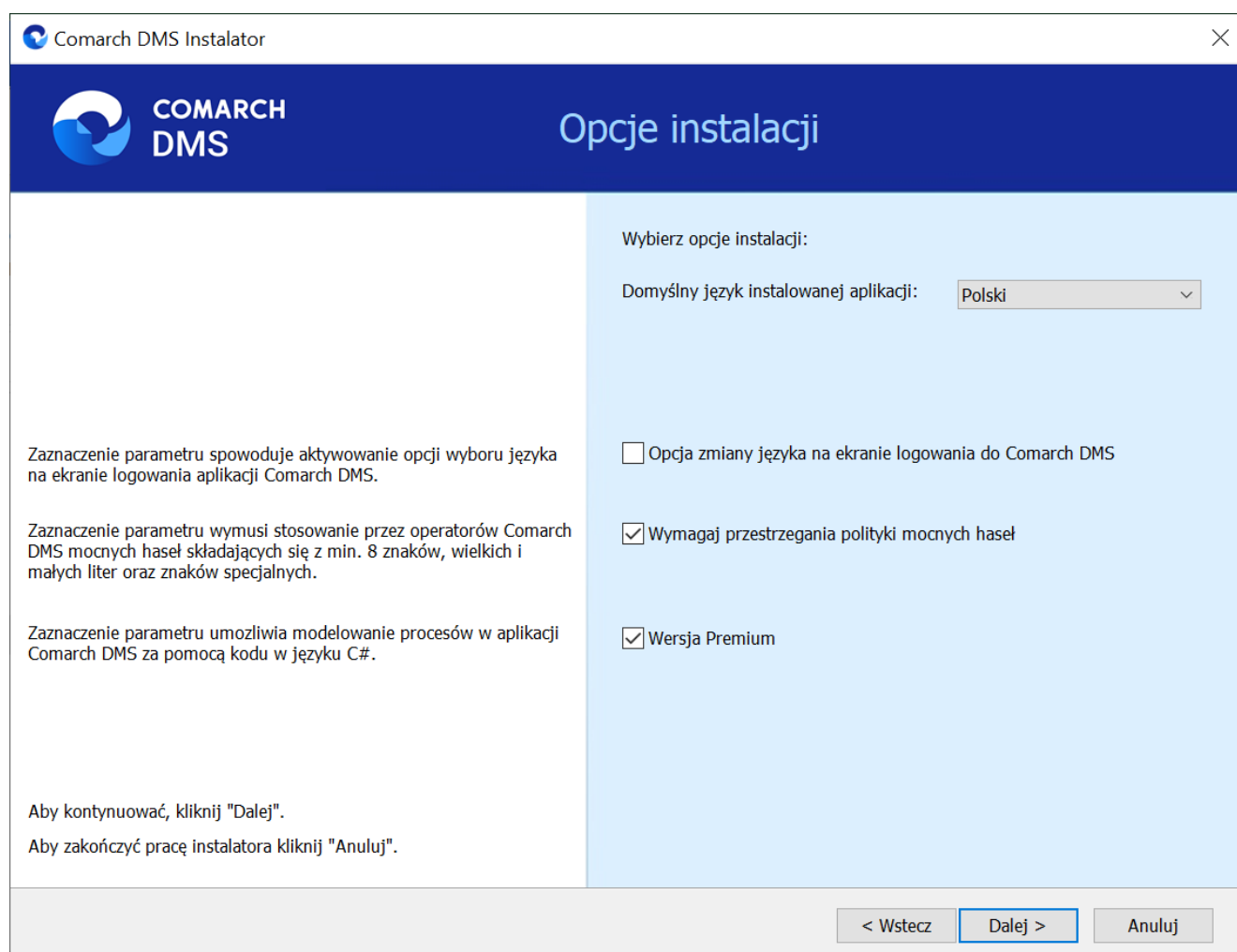
się w oknie edycji kodu C#. (zob. poniżej punkt *Język C# na definicjach kontrolek*)

Jeżeli skompilowane biblioteki dll zostaną usunięte z katalogu .scriptcs_cache, to przed pierwszym użyciem funkcjonalności (podniesienie dokumentu DMS) zamodelowanej metodą C# nastąpi automatyczne skompilowanie bibliotek.

Włączenie funkcjonalności

Aby **włączyć** funkcjonalność należy:

- przy instalacji lub aktualizacji systemu **zaznaczyć** parametr „**Wersja Premium**”



The screenshot shows the 'Comarch DMS Instalator' window with the title 'Opcje instalacji'. The window has a dark blue header with the Comarch DMS logo. The main area is divided into two columns. The left column contains three paragraphs of text explaining the effects of selecting certain options: 1. 'Zaznaczenie parametru spowoduje aktywowanie opcji wyboru języka na ekranie logowania aplikacji Comarch DMS.' 2. 'Zaznaczenie parametru wymusi stosowanie przez operatorów Comarch DMS mocnych haseł składających się z min. 8 znaków, wielkich i małych liter oraz znaków specjalnych.' 3. 'Zaznaczenie parametru umożliwi modelowanie procesów w aplikacji Comarch DMS za pomocą kodu w języku C#.' Below these paragraphs are instructions: 'Aby kontynuować, kliknij "Dalej".' and 'Aby zakończyć pracę instalatora kliknij "Anuluj".' The right column is titled 'Wybierz opcje instalacji:' and contains a dropdown menu for 'Domyślny język instalowanej aplikacji:' set to 'Polski'. Below this are three checkboxes: 'Opcja zmiany języka na ekranie logowania do Comarch DMS' (unchecked), 'Wymagaj przestrzegania polityki mocnych haseł' (checked), and 'Wersja Premium' (checked). At the bottom right are three buttons: '< Wstecz', 'Dalej >' (highlighted with a blue border), and 'Anuluj'.

Comarch DMS Instalator

COMARCH DMS

Opcje instalacji

Wybierz opcje instalacji:

Domyślny język instalowanej aplikacji: Polski

Zaznaczenie parametru spowoduje aktywowanie opcji wyboru języka na ekranie logowania aplikacji Comarch DMS.

Zaznaczenie parametru wymusi stosowanie przez operatorów Comarch DMS mocnych haseł składających się z min. 8 znaków, wielkich i małych liter oraz znaków specjalnych.

Zaznaczenie parametru umożliwi modelowanie procesów w aplikacji Comarch DMS za pomocą kodu w języku C#.

Aby kontynuować, kliknij "Dalej".

Aby zakończyć pracę instalatora kliknij "Anuluj".

☐ Opcja zmiany języka na ekranie logowania do Comarch DMS

☒ Wymagaj przestrzegania polityki mocnych haseł

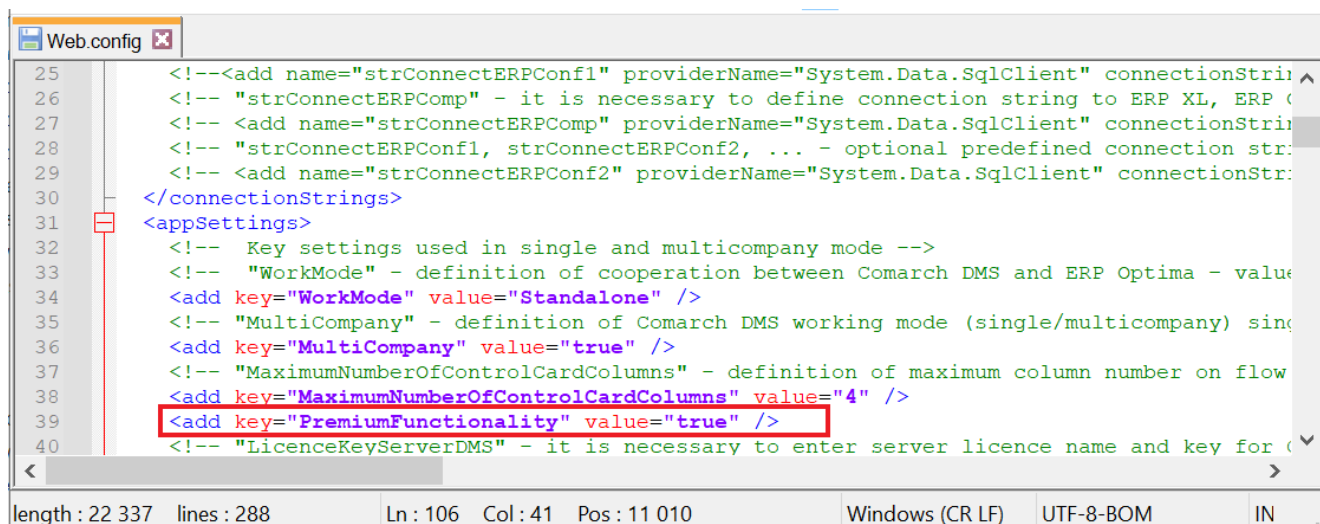
☒ Wersja Premium

< Wstecz Dalej > Anuluj

Wybór opcji „Wersja Premium” podczas instalacji Comarch DMS

lub

- w pliku **Web.config** wartość klucza **PremiumFunctionality** ustawić na „**True**” i wprowadzić odpowiedni klucz licencji.



```
25 <!--<add name="strConnectERPConf1" providerName="System.Data.SqlClient" connectionStrin
26 <!-- "strConnectERPComp" - it is necessary to define connection string to ERP XL, ERP (
27 <!-- <add name="strConnectERPComp" providerName="System.Data.SqlClient" connectionStrin
28 <!-- "strConnectERPConf1, strConnectERPConf2, ... - optional predefined connection str
29 <!-- <add name="strConnectERPConf2" providerName="System.Data.SqlClient" connectionStr
30 </connectionStrings>
31 <appSettings>
32 <!-- Key settings used in single and multicompany mode -->
33 <!-- "WorkMode" - definition of cooperation between Comarch DMS and ERP Optima - value
34 <add key="WorkMode" value="Standalone" />
35 <!-- "MultiCompany" - definition of Comarch DMS working mode (single/multicompany) sing
36 <add key="MultiCompany" value="true" />
37 <!-- "MaximumNumberOfControlCardColumns" - definition of maximum column number on flow
38 <add key="MaximumNumberOfControlCardColumns" value="4" />
39 <add key="PremiumFunctionality" value="true" />
40 <!-- "LicenceKeyServerDMS" - it is necessary to enter server licence name and key for (
```

Parametr „PremiumFunctionality”

Jeżeli operator **nie posiada odpowiedniej licencji**, a zaznaczył parametr „**PremiumFunctionality**”, wówczas **przy próbie logowania** w oknie logowania zostanie **wyświetlona na czerwono informacja** o braku licencji Premium, a dostęp do Comarch DMS **nie będzie możliwy**.

**COMARCH
DMS**

×

Spółka

Dom

Login

Administrator

Hasło

Brak wymaganej licencji Comarch DMS Premium.

Zmień język

Zmień hasło

☐ zapamiętaj logowanie



Okno logowania do aplikacji desktop z informacją o braku licencji Premium

Włączenie edytora C# skutkuje pojawieniem się **dodatkowych opcji** na definicjach kontrolek.

Zasady pisania kodu C#

Kod C# dzięki któremu można programować inicjowanie wartości w poszczególnych kontrolkach programuje się w udostępnianych przez kontrolki edytorach na zasadzie skryptu. Można przyjąć, że obszar edytora to „wnętrze” metody obsługującej określone zdarzenie.

W zależności od edytora kontrolki (wybranego zdarzenia), administrator ma dostęp do listy właściwości, które można przywołać wpisując nazwę „**Global**”.

Obiekt „**Global**” skupia w sobie wszystkie inne obiekty, które można wykorzystać podczas pisania obsługi zdarzenia. Poprzez ten obiekt użytkownik ma również **dostęp do wartości kontrolek znajdujących się na karcie obiegu**.

W przypadku zdarzeń „**Inicjowanie**”, każda kontrolka posiada obiekt „**Common**” z następującymi właściwościami:

- **Globals.Common.DocumentCompanyId** – Id spółki powiązanej z kontrolką, np. DocumentCompanyId (typu int)
- **Globals.Common.DocumentId** – Id dokumentu DMS, np. DocumentId (typu int?)
- **Globals.Common.ParentDocumentId** – Id właściciela dokumentu, np. ParentDocumentId (typu int?)
- **Globals.Common.StageId** – Id etapu w którym znajduje się dokument, StageId (typu int?)
- **Globals.Common.UserId** – Id zalogowanego operatora, np. UserId (typu int)
- **Globals.Common.WorkflowId** – Id typu obiegu, np. WorkflowId (typu int)

Typ zakończony „?” (int?) oznacza, że oprócz wartości numerycznych zmienna może przechowywać **również wartość NULL (pustą)**. Przykładowa sytuacja to **brak wpisanej wartości** w kontrolce typu liczba całkowita

W przypadku zdarzeń „**Obserwacja**”, każda z kontrolek posiada

dodatkowo właściwość:

- **Globals.Common**.SenderControlName – Identyfikator kontrolki inicjującej zdarzenie

Uwaga

Wszystkie właściwości udostępniane przez obiekt „**Common**” są tylko do odczytu.

Obiekt „**Global**” posiada również obiekt „**MainFrame**”.

Po nazwie „**MainFrame**” występują **nazwy identyfikatorów kontrolek, następnie właściwości lub metody**.

Elementy udostępniane przez obiekt „**MainFrame**” **zależą od kontrolki**, dla której pisze się obsługę zdarzenia, **zdarzeń** oraz od **innych zdefiniowanych na karcie obiegu kontrolek**.

W przypadku obiektu „**MainFrame**”, dostęp do właściwości odbywa się w oparciu o następującą ścieżkę:

Globals.MainFrame.Identyfikator kontrolki.Nazwa właściwości lub nazwa metody

Przykład

Przykładowa linia kontrolki o identyfikatorze „Pole1” pozwalająca na wprowadzenie lub zmianę wartości kontrolki za pomocą właściwości „Text”:

```
Globals.MainFrame.Pole1.Text = „NR/123/2020”;
```

Język C# na definicjach kontrolek

Oprogramowano **funkcjonalność tworzenia własnych zdarzeń w języku C#**. W aktualnej wersji możliwość modelowania zdarzeń w języku C# dostępna jest w kontrolkach:

- Dane tabelaryczne,
- Dane tabelaryczne, kolumna typu lista,
- Data i godzina,
- Dokument elektroniczny,
- Komunikat,
- Kontrahent,
- Liczba całkowita,
- Liczba rzeczywista,
- Liczba stałoprzecinkowa,
- Lista,
- Tekst,
- Towar,
- Własna akcja.

W większości kontrolek możliwe jest **tworzenie skryptów** służących zarówno **inicjowaniu wartości w kontrolkach**, jak i **obserwowaniu wartości innych kontrolek**. Obserwowanie wartości umożliwia „reagowanie” na zmiany. Dla kontrolek typu dane tabelaryczne oraz własna akcja możliwa jest również „obserwacja samej siebie”.

Tekst

Nazwa wyświetlana	Numer dokumentu
Nazwa (identyfikator)	Numer_dokumentu
Prezentuj na całej szerokości	☐
Ustaw kontrolkę na początku wiersza	☐
Max długość	900
Tekst wielolinijkowy	1  
Pokaż na liście	☐

Inicjowanie wartości

Inicjowanie	C# Script ▼ Brak SQLOD C# Script
Inicjuj przy każdym otwarciu dokumentu	

Zmiana wartości (Obserwator)

Kontrolki powiązane	Dodaj
Wzór na wartość	Wyrażenie ▼

[Ustaw](#)

Pozycja C# Script w opcjach inicjowania wartości kontrolki typu tekst

Własna akcja

Nazwa wyświetlana	Własna akcja
Nazwa (identyfikator)	Wlasna_akcja
Prezentuj na całej szerokości	☐
Ustaw kontrolkę na początku wiersza	☐
Plik wykonywalny	Ustaw
IIS	Ustaw
Procedura	Ustaw
Procedura ERP	Ustaw
C# Script	Ustaw
Nazwa spółki	
Potwierdzenie wykonania akcji	☐
Pytaj przed uruchomieniem	☐
Zapisz przed uruchomieniem	☐

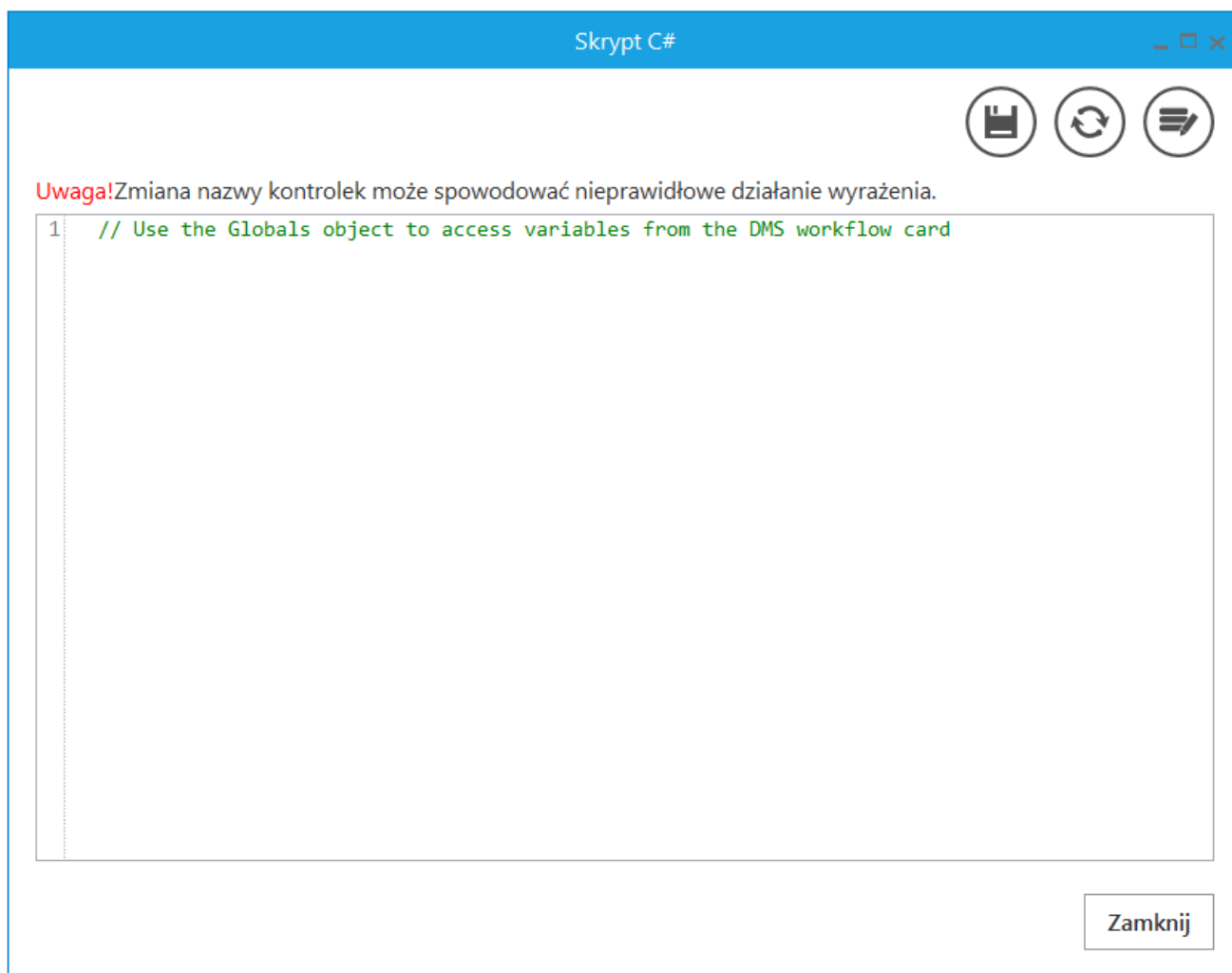
Pozycja C# Script w kontrolce typu Własna akcja

Po **wskazaniu** inicjowania wartości za pomocą **C# Script** i

kliknięciu w

[Ustaw](#)

uruchamiana jest formatka, w której należy **wpisać** kod **C#**.



Okno Skrypt C#

Okno **Skrypt C#** składa się z pola, w którym należy **wpisać kod**,

a następnie **skompilować i zapisać** przyciskiem  **[Kompiluj i zapisz]**.

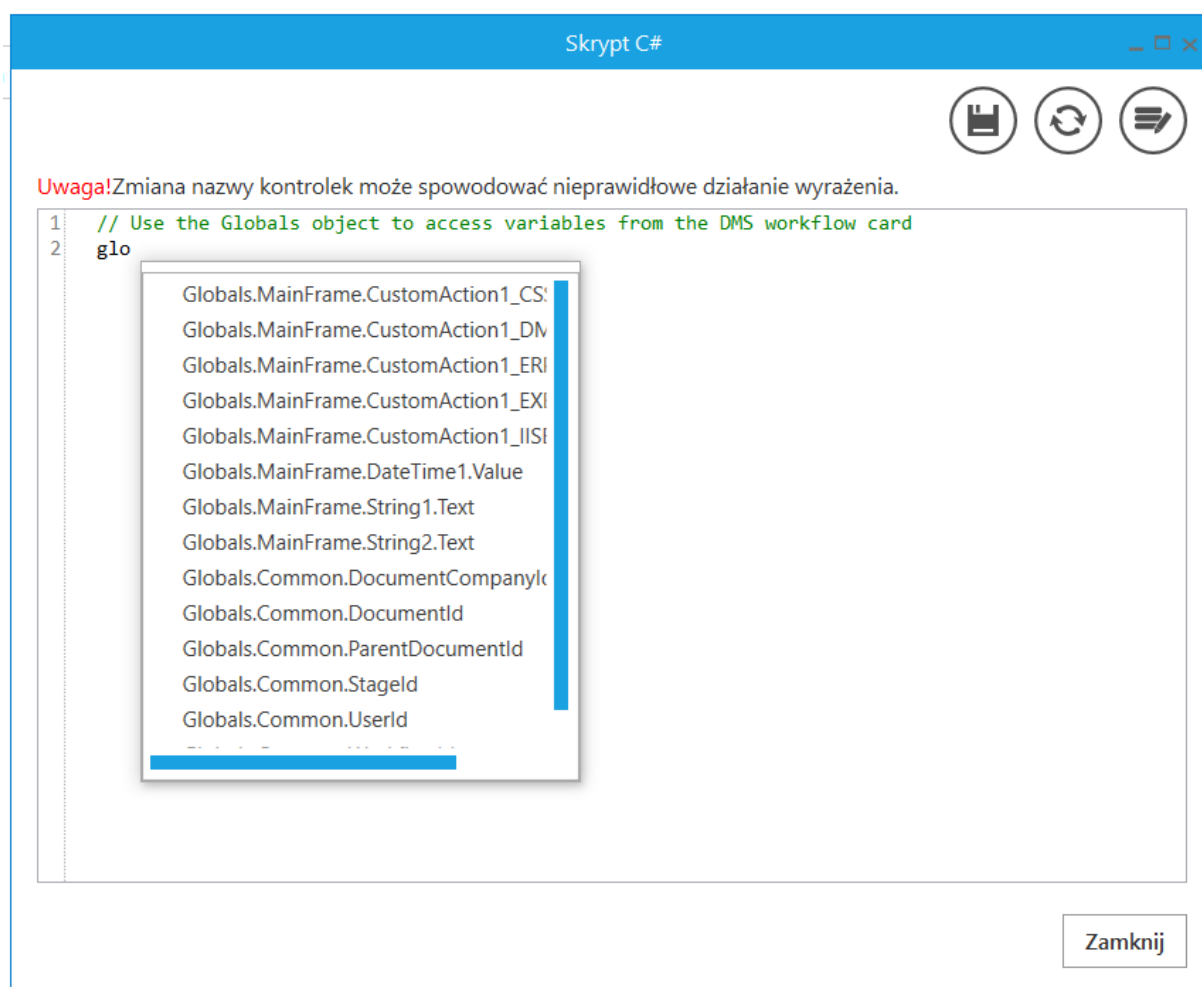
Kod po skompilowaniu **zapisywany jest w formie biblioteki** w lokalizacji **...\bin\.scriptsc_cache** w katalogu, w którym zainstalowano aplikację serwerową (dawna web) Comarch DMS.

W polu **Referencje**  należy wpisać **nazwy bibliotek**, z których będzie korzystał kod C#. **Domyślną lokalizacją**, w której należy umieścić biblioteki jest **...\bin\scripts_bin** w

katalogu, w którym zainstalowano aplikację serwerową (dawna web)Comarch DMS.

Warunkiem koniecznym do poprawnego działania jest **nadanie operatorowi IIS_USER uprawnień pełnej kontroli do katalogu .scriptscache**, pozwoli to na **dynamiczny zapis** skompilowanych plików.

W edytorze skryptów C# wyświetlane są **podpowiedzi** zawierające obiekty kontrolek karty obiegu (**Globals.MainFrame.***) oraz zmienne (**Globals.Common.***).



Podpowiedzi wyświetlane w edytorze skryptów C#

W kontrolce typu Dane tabelaryczne dostępne są następujące

funkcje i właściwości:

- **MainFrame.<nazwaDT>.Items[Index]...** – odwołanie do wartości kontrolki
 - Funkcje
 - **MainFrame.<nazwaDT>.AddRow()** – **dodaj** kolejny wiersz.
 - **MainFrame.<nazwaDT>.AddRows(count)** – **dodaj** wiersze, gdzie „**count**” oznacza ile wierszy ma zostać dodane.
 - **MainFrame.<nazwaDT>.RemoveRow(index)** – **usuń** wiersz, gdzie „**index**” to wiersz, który ma zostać usunięty.
 - **MainFrame.<nazwaDT>.Clear()** – **usuwa** wszystkie wiersze.
 - Właściwości
 - **MainFrame.<nazwaDT>.Column** – informacja o modyfikowanej komórce – **numer kolumny**.
 - **MainFrame.<nazwaDT>.Row** – informacja o modyfikowanej komórce – **numer wiersza**.
 - **MainFrame.<nazwaDT>.RowCount** – informacja o **ilości wierszy w DT**.

Wskazówka

W tablicach C# – **pierwszy wiersz to 0**

Uwaga

Podczas pierwszego uruchomienia operacji zamodelowanej mechanizmem C# może nastąpić **opóźnienie działania mechanizmu**. Wynika to z konieczności załadowania biblioteki dll do pamięci komputera.

Ze względu na konieczność zachowania nomenklatury nazewnicznej języka C# **nie jest możliwe** używanie polskich znaków diakrytycznych oraz spacji w identyfikatorach kontrolek.

Użytkownik **może modyfikować wartości kontrolek** poprzez modyfikację właściwości „**Value**” lub „**Text**”. Np.
Globals.MainFrame.Liczba.Value = 10;

Możliwe jest to jednak **tylko** dla kontrolek, dla których został wprowadzony kod C#, czyli zostało wywołane zdarzenie **OnInit** lub **OnChange** (obserwacja). Pozostałe właściwości są **tylko do odczytu**.

Przykład

Przepisanie wartości z kontrolki typu liczba całkowita o nazwie (identyfikatorze) *Liczba* do kontrolki typu tekst o nazwie (identyfikatorze) *Tekst*:

```
Globals.MainFrame.Tekst.Text  
= Globals.MainFrame.Liczba.Value.ToString();
```

Przykład

Pobranie wartości z kontrolki składnik i zapisanie tej wartości w kontrolce suma powiększonej o 1.

```
var a = Globals.MainFrame.skladnik.Value ?? 0; // jeżeli pusta  
wartość zapisz 0  
a = a + 1;  
Globals.MainFrame.suma.Value = a;
```

Przykład

Przepisanie wartości z kontrolki typu tekst do kontrolki typu liczba całkowita, pod warunkiem, że wpisana wartość jest liczbą.

```
if (IsNumeric (Globals.MainFrame.Wartosc_tekst.Text))
{
    Globals.MainFrame.Liczba.Value =
    Int32.Parse(Globals.MainFrame.Wartosc_tekst.Text);
}

public static bool IsNumeric(string value)
{
    return value.All(char.IsNumber);
}
```

Przykład

Przykład pobrania informacji o identyfikatorze operatora i wyświetlenia jej w kontrolce typu tekst.

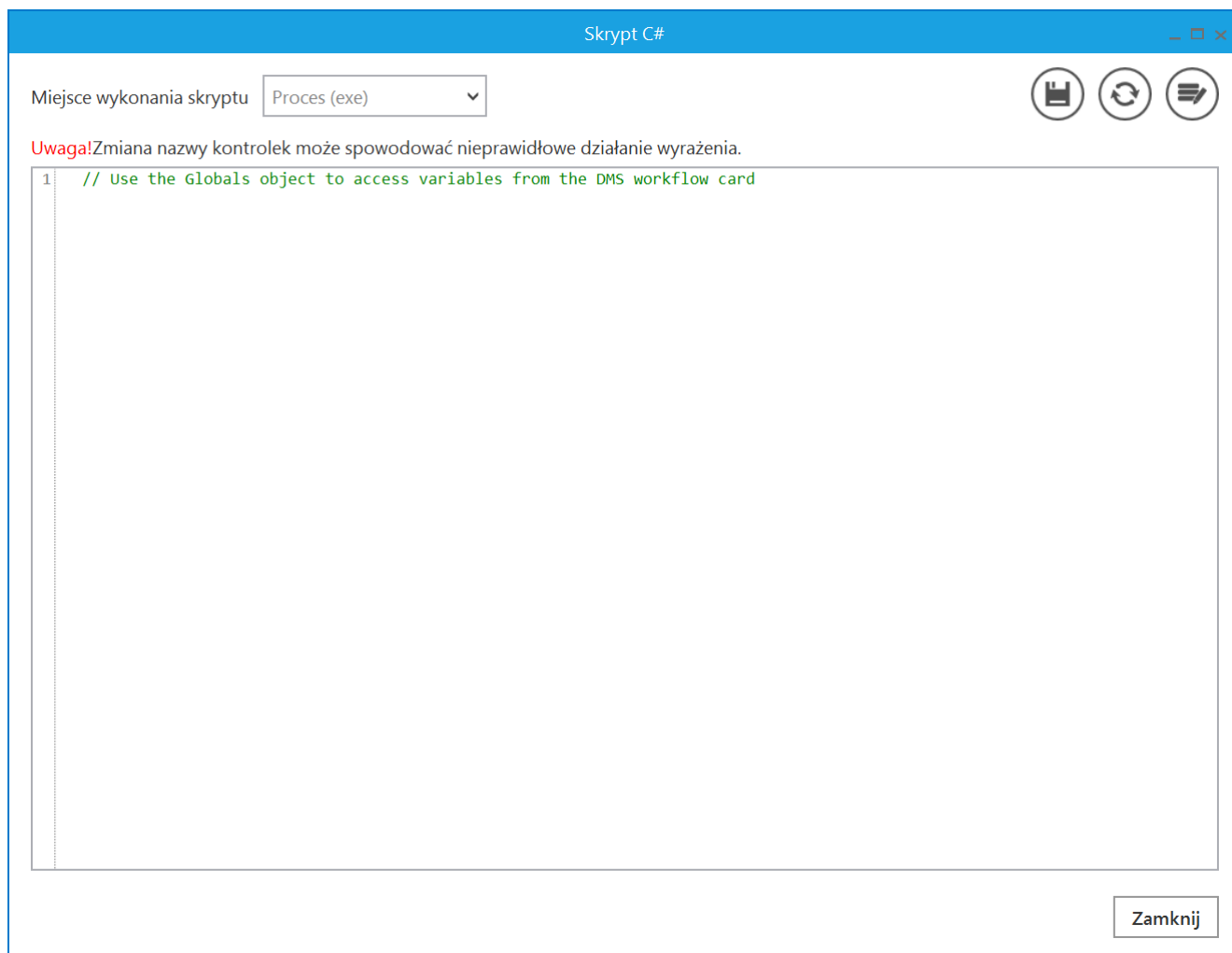
```
var info = „Identyfikator operatora:
” + Globals.Common.UserId;
Globals.MainFrame.Informacja.Text = info;
```

Od wersji 2024.3.0 zablokowano możliwość zmiany miejsca wykonywania skryptów C# – dostępna jest jedynie opcja Proces (exe).

Uwaga

W przypadku wcześniejszych wersji uruchomienie skryptu na niezależnym procesie exe jest zalecane w przypadku obsługi

API, które nie powinno działać na procesach wielowątkowych takich jak proces IIS.



Okno definiowania procedury wywołania kontrolki typu Własna akcja w języku C#

[Powrót do początku artykułu](#)

Weryfikacja kontrahenta na białej liście podatników

Od wersji 2022.0 w Comarch DMS Premium dostępna jest funkcjonalność weryfikacji kontrahentów na białej liście podatników.

Uwaga

Funkcjonalność dostępna jest **tylko** w przypadku, jeśli Comarch DMS jest na gwarancji. Okres gwarancji jest sprawdzany za każdym razem podczas weryfikacji kontrahenta na białej liście.

Kontrahent jest **weryfikowany według numeru NIP**. Wedle tej informacji można także sprawdzić, **czy kontrahent był zarejestrowany na białej liście podatników w danym dniu**.

Aby zweryfikować kontrahenta, w konfiguracji kontrolki używa się metody C#:

- **GetStatusVAT(string nip)** – aby sprawdzić, **czy kontrahent jest na białej liście podatników**;
- **GetStatusVATForDate(string nip, DateTime date)** – aby sprawdzić, **czy kontrahent w danym dniu był na białej liście podatników**

Przykład

Metoda **GetStatusVAT(string nip)** – przykład

1. Użytkownik w ramach obiegu dokumentów **tworzy dwie kontrolki typu tekst**.

Są to:

- Kontrolka *Czy kontrahent zweryfikowany?* (identyfikator

kontrolki: *weryfikacja_kontrolka*)

- Kontrolka *NIP* (identyfikator kontrolki: *nip_kontrolka*)

Tekst

Nazwa wyświetlana

Czy kontrahent zweryfikowany?

Nazwa (identyfikator)

weryfikacja_kontrolka

Prezentuj na całej szerokości ☐

Ustaw kontrolkę na początku wiersza ☐

Max długość

900

Tekst wielolinijkowy

1

Pokaż na liście ☐

Inicjowanie wartości

Inicjowanie

Brak

Zmiana wartości (Obserwator)

Kontrolki powiązane

NIP (*nip_kontrolka*)



[Dodaj](#)

Wzór na wartość

C# Script

[Ustaw](#)

Kontrolka „Czy kontrahent zweryfikowany?”

Tekst

Nazwa wyświetlana	NIP
Nazwa (identyfikator)	nip_kontrolka
Prezentuj na całej szerokości	☐
Ustaw kontrolkę na początku wiersza	☐
Max długość	900
Tekst wielolinijkowy	1
Pokaż na liście	☐

Inicjowanie wartości	
Inicjowanie	Brak

Zmiana wartości (Obserwator)

Kontrolki powiązane	Dodaj
Wzór na wartość	Wyrażenie

[Ustaw](#)

Kontrolka „NIP”

2. Następnie we właściwościach kontrolki *Czy kontrahent zweryfikowany?* użytkownik **dodaje** w ramach pola *Kontrolki powiązane* kontrolkę *NIP*, a jako *Wzór na wartość* **wybiera** *C# Script* i **klika** *Ustaw*. Następnie wpisuje następujący wzór:

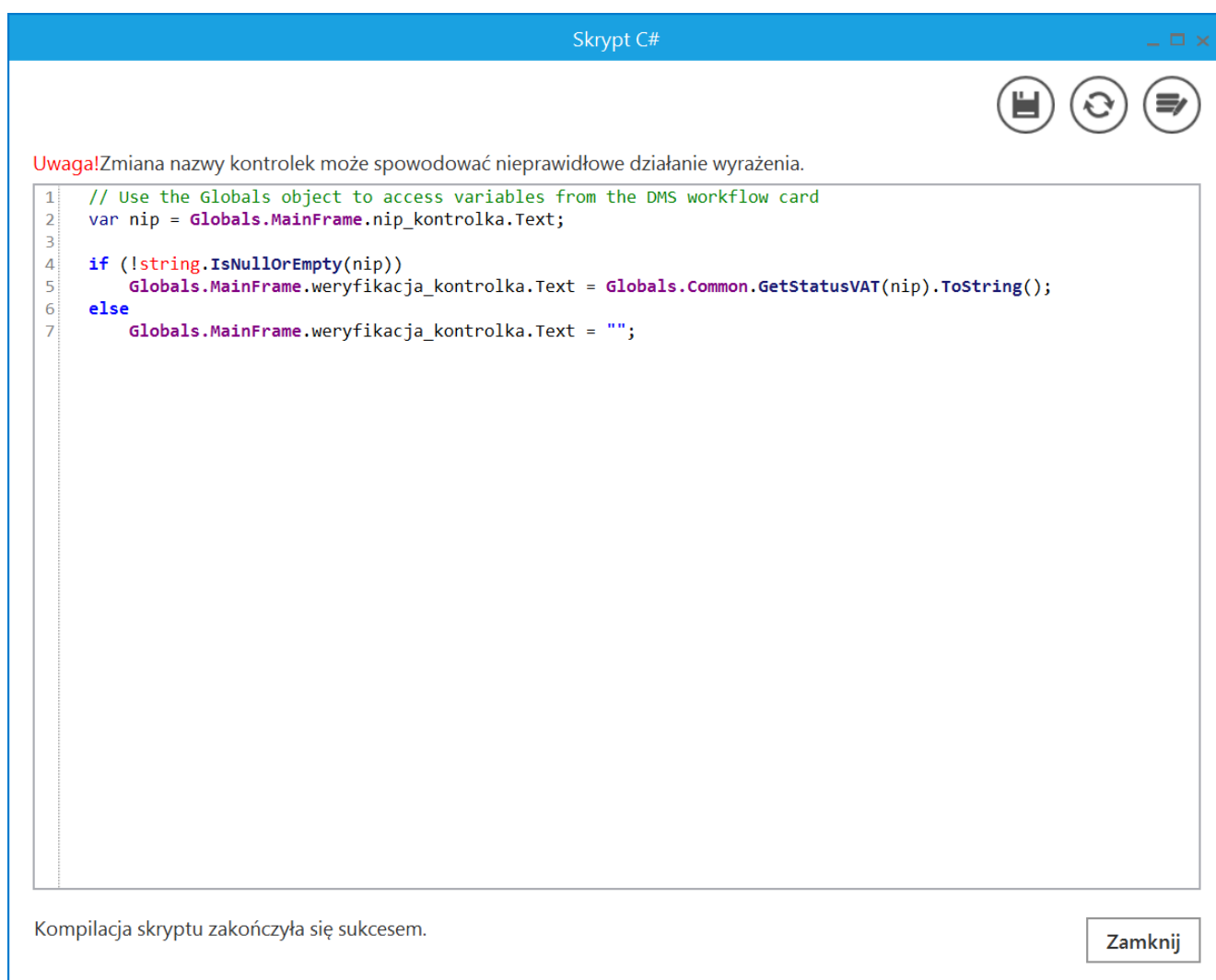
```
var nip = Globals.MainFrame.nip_kontrolka.Text;

if (!string.IsNullOrEmpty(nip))
    Globals.MainFrame.weryfikacja_kontrolka.Text =
    Globals.Common.GetStatusVAT(nip).ToString();
else
    Globals.MainFrame.weryfikacja_kontrolka.Text = „”;
```



i wybiera przycisk , aby skompilować wzór i zapisać.

3. Następnie zamyka okno *Skrypt C#*, zapisuje obieg i sprawdza działanie kontrolek.



Wprowadzanie kodu C# do weryfikacji kontrahenta

Możliwe wartości zwrócone po weryfikacji do kontrolki „Czy kontrahent zweryfikowany?” to:

- **No** – podatnik **nie jest** VATowcem
- **Yes** – podatnik **jest czynnym** VATowcem
- **Free** – podatnik **jest zwolniony** z VAT
- **None** – wystąpił **błąd podczas weryfikacji**
- **NIPError** – **niepoprawny numer NIP**

Karta obiegu

Czy kontrahent zweryfikowany?

Yes

NIP

6770065406

Weryfikacja kontrahenta – kontrahent jest czynnym VATowcem

Karta obiegu

Czy kontrahent zweryfikowany?

NIPError

NIP

567896789

Weryfikacja kontrahenta – niepoprawny numer NIP

Przykład

Metoda `GetStatusVATForDate(string nip, DateTime date)` – przykład

1. Użytkownik w ramach obiegu dokumentów **tworzy dwie kontrolki typu tekst i jedną kontrolkę typu data**. Są to:

- Kontrolka typu tekst *Czy kontrahent był danego dnia na białej liście?* (identyfikator kontrolki: *kontrahent_dzien*)
- Kontrolka typu tekst *NIP* (identyfikator kontrolki: *NIP*)
- Kontrolka typu data *Data* (identyfikator kontrolki: *Data*)

Tekst

Nazwa wyświetlana	Czy kontrahent był danego dnia na białej liście?	
Nazwa (identyfikator)	kontrahent_dzien	
Prezentuj na całej szerokości	☐	
Ustaw kontrolkę na początku wiersza	☐	
Max długość	900	
Tekst wielolinijkowy	1	
Pokaż na liście	☐	

Inicjowanie wartości

Inicjowanie	Brak	
-------------	------	--

Zmiana wartości (Obserwator)

Kontrolki powiązane	NIP (NIP)	
	Data (Data)	
	Dodaj	
Wzór na wartość	C# Script	

[Ustaw](#)

Kontrolka „Czy kontrahent był danego dnia na białej liście?”

Tekst

Nazwa wyświetlana

NIP

Nazwa (identyfikator)

NIP

Prezentuj na całej szerokości

☐

Ustaw kontrolkę na początku wiersza

☐

Max długość

900

Tekst wielolinijkowy

1

Pokaż na liście

☐

Inicjowanie wartości

Inicjowanie

Brak



Zmiana wartości (Obserwator)

Kontrolki powiązane

[Dodaj](#)

Wzór na wartość

Wyrażenie



[Ustaw](#)

Kontrolka „NIP”

Data i godzina

Nazwa wyświetlana	Data
Nazwa (identyfikator)	Data
Prezentuj na całej szerokości	☐
Ustaw kontrolkę na początku wiersza	☐
Czas	Nie
Pokaż na liście	☐
Inicjowanie wartości	
Inicjowanie	Brak
Zmiana wartości (Obserwator)	
Kontrolki powiązane	Dodaj
Wzór na wartość	Wyrażenie
	Ustaw

Kontrolka „Data”

2. Następnie we właściwościach kontrolki *Czy kontrahent był danego dnia na białej liście?* użytkownik **dodaje** w ramach pola *Kontrolki powiązane* kontrolki: *NIP* i *Data*, a jako *Wzór na wartość* **wybiera** *C# Script* i **klika** *Ustaw*. Następnie **wpisuje** następujący wzór:

```
var nip = Globals.MainFrame.NIP.Text;
var data1 = Convert.ToDateTime(Globals.MainFrame.Data.Value);

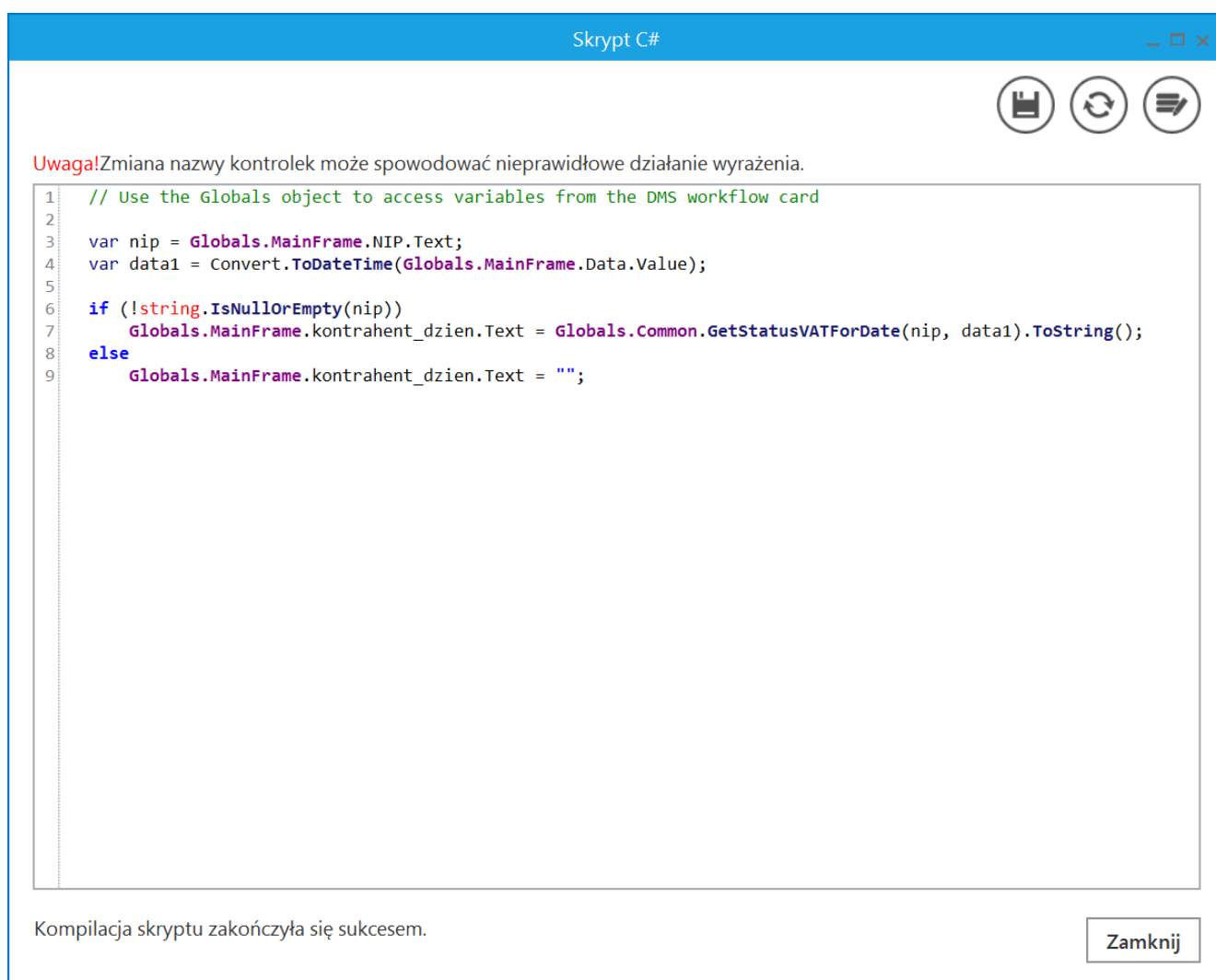
if (!string.IsNullOrEmpty(nip))
    Globals.MainFrame.kontrahent_dzien.Text =
    Globals.Common.GetStatusVATForDate(nip, data1).ToString();
else
    Globals.MainFrame.kontrahent_dzien.Text = „”;
```

i **wybiera** przycisk



, aby **skompilować wzór i zapisać**.

3. Następnie użytkownik **zamyka** okno *Skrypt C#*, **zapisuje** obieg i **sprawdza** działanie kontrollek.



Wprowadzanie kodu C# do weryfikacji kontrahenta w danym dniu

Możliwe wartości zwrócone po weryfikacji do kontrolki „Czy kontrahent był danego dnia na białej liście?” to:

- **No** – podatnik **nie był w danym dniu** VATowcem
- **Yes** – podatnik **był w danym dniu** czynnym VATowcem
- **Free** – podatnik **był w danym dniu zwolniony** z VAT
- **None** – wystąpił **błąd podczas weryfikacji**
- **NIPError** – **niepoprawny numer NIP**

Spółka : Default company

Data utworzenia : 30-05-2022



Karta obiegu



NIP

6770065406

Data

07.04.2022



Czy kontrahent był danego dnia na białej liście?

Yes

Weryfikacja kontrahenta według stanu na 7 kwietnia 2022 – kontrahent był wtedy czynnym VATowcem

[Powrót do początku artykułu](#)

Konfiguracja automatycznego

trybu pracy (robotyzacja procesów) – przykład zastosowania kodu C#

Spis treści artykułu

- [Informacje ogólne](#)
- [Zakładka „Konfiguracja automatycznego trybu pracy” – nadawanie uprawnień do zakładki i elementy okna](#)
- [Zakładka „Edytor skryptów”](#)
- [Zakładka „Uprawnienia”](#)
- [Uruchamianie i zatrzymywanie programu](#)
- [Konfiguracja i uruchamianie automatycznego trybu pracy – przykład](#)

Informacje ogólne

W wersji 2023.1.0 w aplikacji desktop Comarch DMS udostępniono funkcjonalność robotyzacji procesów.

Uwaga

Robotyzacja procesów jest dostępna jedynie w aplikacji desktop Comarch DMS .

W ramach mechanizmu robotyzacji użytkownik korzysta ze skryptów C#, z zastosowaniem predefiniowanych klas. W tym celu został udostępniony lokalny kompilator C#.

Skrypty mogą korzystać z referencji.

Skrypty są aktywowane ręcznie

Uwaga

Aby możliwe było uruchomienie skryptów, operator musi być zalogowany do aplikacji desktop Comarch DMS.

Zakładka „Konfiguracja automatycznego trybu pracy” – nadawanie uprawnień do zakładki i elementy okna



W wersji 2023.1.0 została dodana nowa zakładka [Konfiguracja automatycznego trybu pracy] , dostępna w menu głównym aplikacji desktop Comarch DMS nad ikoną panelu



użytkownika .





Zakładka [Konfiguracja automatycznego trybu pracy] jest dostępna w menu głównym aplikacji desktop Comarch DMS jedynie wtedy, gdy danemu operatorowi nadano uprawnienie do tej zakładki na karcie pracownika.



Aby przyznać uprawnienie do zakładki [Konfiguracja automatycznego trybu pracy] pracownikowi należy przejść na



zakładkę [Struktura organizacyjna], następnie kliknąć na sekcje „Pracownicy”, na danego pracownika i na zakładkę „Uprawnienia”. Następnie należy wybrać sekcję „Funkcjonalności” i w ramach pola „Konfiguracja automatycznego trybu pracy” wybrać wartość „Tak”, a następnie zapisać zmiany znakiem dyskietki. Wówczas pracownik będzie mógł tworzyć skrypty do pracy automatycznej.

Jeżeli w polu „Konfiguracja automatycznego trybu pracy”



ustawiona jest wartość „Nie”, wówczas zakładka nie jest widoczna dla pracownika i nie ma on uprawnień do tworzenia skryptów do pracy automatycznej.

Uwaga

Jeżeli operator nadaje uprawnienie „Konfiguracja automatycznego trybu pracy” samemu sobie, wówczas zakładka będzie widoczna dopiero po ponownym zalogowaniu .

Struktura organizacyjna

Comarch DMS 2023.1.0 Administrator2

Dział: **Pracownicy** Stanowiska

Szukaj

Firma X

Adam Nowak (AdamNowak)

Administrator (Administrator)

Administrator2 (Administrator2)

Administrator3 (Administrator3)

Anna Maj (AnnaMaj)

Barbara Bar (BarbaraBar)

Cecylia Kowal (CecyliaKowal)

Dorota Nowak (DorotaNowak)

MaciejMan (MaciejMan)

MarekKról (MarekKrol)

Marlena Kawa (MarlenaKawa)

☐ Pokaż archiwalne

Pracownik

OGÓLNE **UPRAWNIENIA** TYPY OBIEGU DANE OSOBOWE API

Uprawnienia ogólne

Praca z dokumentem

Funkcjonalności

Inicjowanie obiegu na podstawie skanu dok. Nie

Automatyczne generowanie dokumentów Nie

Wydruk książki nadawczej Nie

Eksport listy dokumentów do arkusza Nie

Kreator raportów i dashboardów Tak

Konfiguracja automatycznego trybu pracy Tak

Praca z modulem Obieg Dokumentów i Repozytorium

Nadawanie pracownikowi uprawnień do zakładki „Konfiguracja automatycznego trybu pracy”



Po pierwszym przejściu na zakładkę [Konfiguracja automatycznego trybu pracy] widoczne jest puste okno ze zwiniętym panelem „Skrypty”. W górnej części okna dostępne są następujące przyciski:



- [Dodaj] – po naciśnięciu przycisku zostaje dodany nowy skrypt



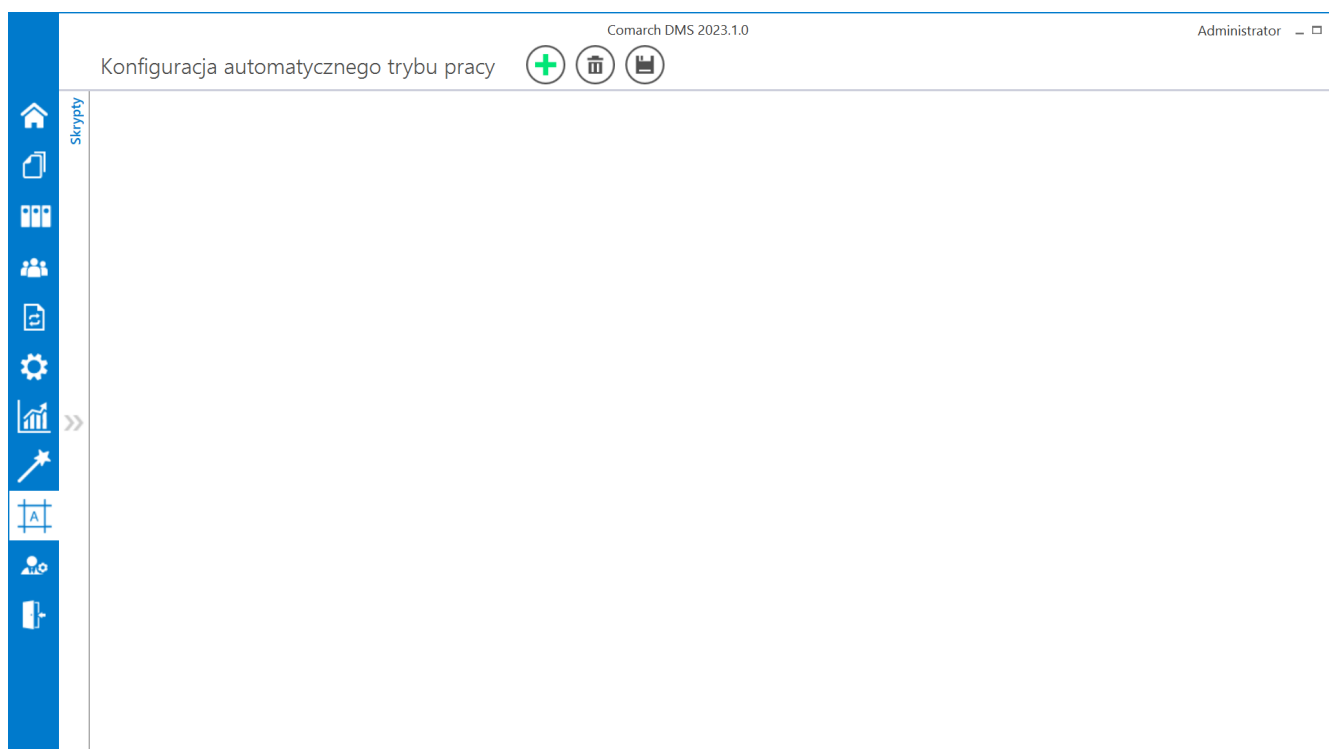
- [Usuń] – przycisk pozwala na usunięcie programu z listy programów w oknie „Konfiguracja automatycznego trybu pracy” oraz z listy programów, które mogą być uruchomione przez użytkownika



- [Zapisz] – przycisk służy do zapisywania zmian w programie



- [Uruchom] – przycisk otwiera okno „Automatyczny tryb pracy”; przycisk widoczny jedynie dla użytkowników, którym nadano uprawnienie do danego programu na zakładce „Uprawnienia” w ramach okna „Konfiguracja automatycznego trybu pracy” (zob. [Uruchamianie i zatrzymywanie programu](#)).



Okno „Konfiguracja automatycznego trybu pracy” po pierwszym otwarciu zakładki

Aby dodać nową pozycję, należy nacisnąć przycisk



[Dodaj]. Wówczas zostanie dodany nowy program z predefiniowaną nazwą „Nazwa programu 1”. W ramach okna widoczne są dwie zakładki:

- Edytor skryptów
- Uprawnienia

Zakładka „Edytor skryptów”



W wersji 2023.1.0 została dodana nowa zakładka [Konfiguracja automatycznego trybu pracy] , dostępna w menu głównym aplikacji desktop Comarch DMS nad ikoną panelu

użytkownika



.





Zakładka [Konfiguracja automatycznego trybu pracy] jest dostępna w menu głównym aplikacji desktop Comarch DMS jedynie wtedy, gdy danemu operatorowi nadano uprawnienie do tej zakładki na karcie pracownika.



Aby przyznać uprawnienie do zakładki [Konfiguracja automatycznego trybu pracy] pracownikowi należy przejść na



zakładkę [Struktura organizacyjna], następnie kliknąć na sekcje „Pracownicy”, na danego pracownika i na zakładkę „Uprawnienia”. Następnie należy wybrać sekcję „Funkcjonalności” i w ramach pola „Konfiguracja automatycznego trybu pracy” wybrać wartość „Tak”, a następnie zapisać zmiany znakiem dyskietki. Wówczas pracownik będzie mógł tworzyć skrypty do pracy automatycznej.

Jeżeli w polu „Konfiguracja automatycznego trybu pracy”



ustawiona jest wartość „Nie”, wówczas zakładka nie jest widoczna dla pracownika i nie ma on uprawnień do tworzenia skryptów do pracy automatycznej.

Uwaga

Jeżeli operator nadaje uprawnienie „Konfiguracja automatycznego trybu pracy” samemu sobie, wówczas zakładka będzie widoczna dopiero po ponownym zalogowaniu .

Struktura organizacyjna

Comarch DMS 2023.1.0 Administrator2

Dział: **Pracownicy** Stanowiska

Szukaj

Firma X

Adam Nowak (AdamNowak)

Administrator (Administrator)

Administrator2 (Administrator2)

Administrator3 (Administrator3)

Anna Maj (AnnaMaj)

Barbara Bar (BarbaraBar)

Cecylia Kowal (CecyliaKowal)

Dorota Nowak (DorotaNowak)

MaciejMan (MaciejMan)

MarekKról (MarekKrol)

Marlena Kawa (MarlenaKawa)

☐ Pokaż archiwalne

Pracownik

OGÓLNE **UPRAWNIENIA** TYPY OBIEGU DANE OSOBOWE API

Uprawnienia ogólne

Praca z dokumentem

Funkcjonalności

Inicjowanie obiegu na podstawie skanu dok. Nie

Automatyczne generowanie dokumentów Nie

Wydruk książki nadawczej Nie

Eksport listy dokumentów do arkusza Nie

Kreator raportów i dashboardów Tak

Konfiguracja automatycznego trybu pracy Tak

Praca z modulem Obieg Dokumentów i Repozytorium

Nadawanie pracownikowi uprawnień do zakładki „Konfiguracja automatycznego trybu pracy”



Po pierwszym przejściu na zakładkę [Konfiguracja automatycznego trybu pracy] widoczne jest puste okno ze zwiniętym panelem „Skrypty”. W górnej części okna dostępne są następujące przyciski:



- [Dodaj] – po naciśnięciu przycisku zostaje dodany nowy skrypt



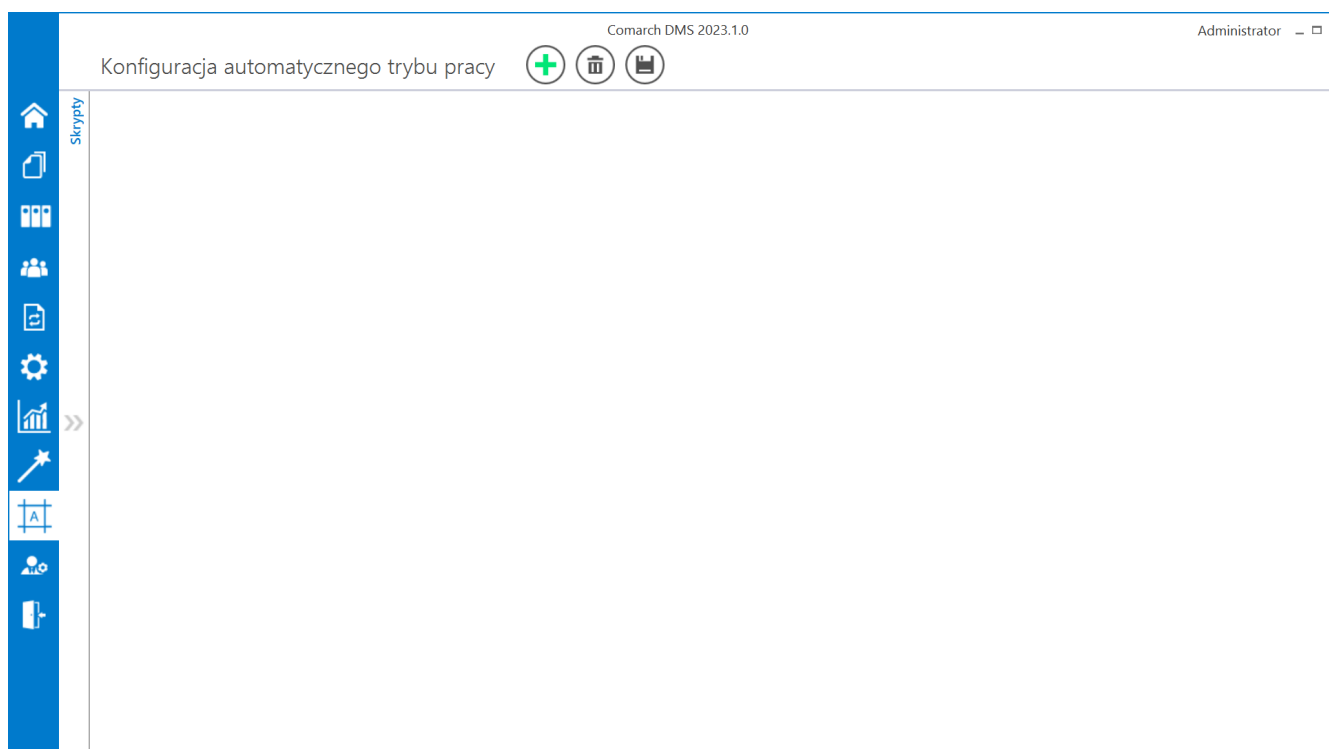
- [Usuń] – przycisk pozwala na usunięcie programu z listy programów w oknie „Konfiguracja automatycznego trybu pracy” oraz z listy programów, które mogą być uruchomione przez użytkownika



- [Zapisz] – przycisk służy do zapisywania zmian w programie



- [Uruchom] – przycisk otwiera okno „Automatyczny tryb pracy”; przycisk widoczny jedynie dla użytkowników, którym nadano uprawnienie do danego programu na zakładce „Uprawnienia” w ramach okna „Konfiguracja automatycznego trybu pracy” (zob. [Uruchamianie i zatrzymywanie programu](#)).



Okno „Konfiguracja automatycznego trybu pracy” po pierwszym otwarciu zakładki

Aby dodać nową pozycję, należy nacisnąć przycisk



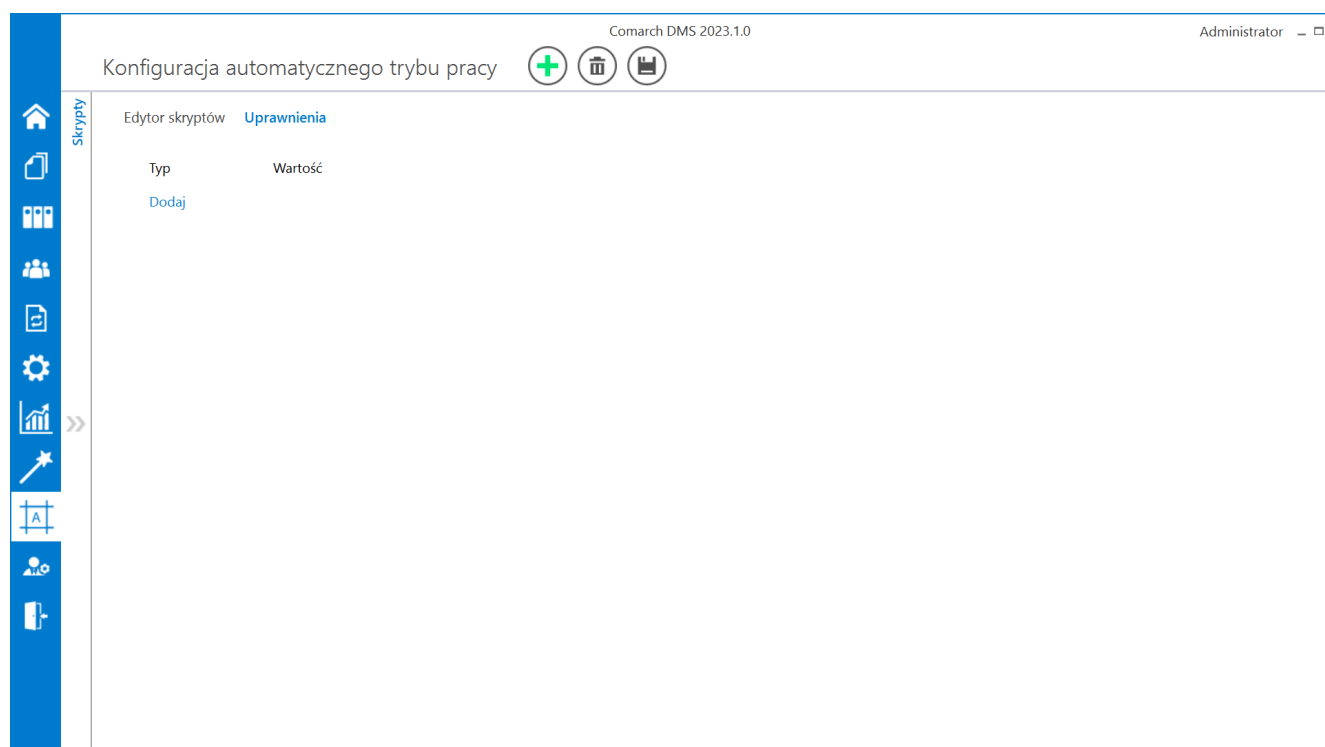
[Dodaj]. Wówczas zostanie dodany nowy program z predefiniowaną nazwą „Nazwa programu 1”. W ramach okna widoczne są dwie zakładki:

- Edytor skryptów
- Uprawnienia

Zakładka „Uprawnienia”

W ramach zakładki „Uprawnienia” zostają zdefiniowane uprawnienia operatorów do uruchamiania skryptów.

Uprawnienia dla każdego utworzonego programu są nadawane oddzielnie.



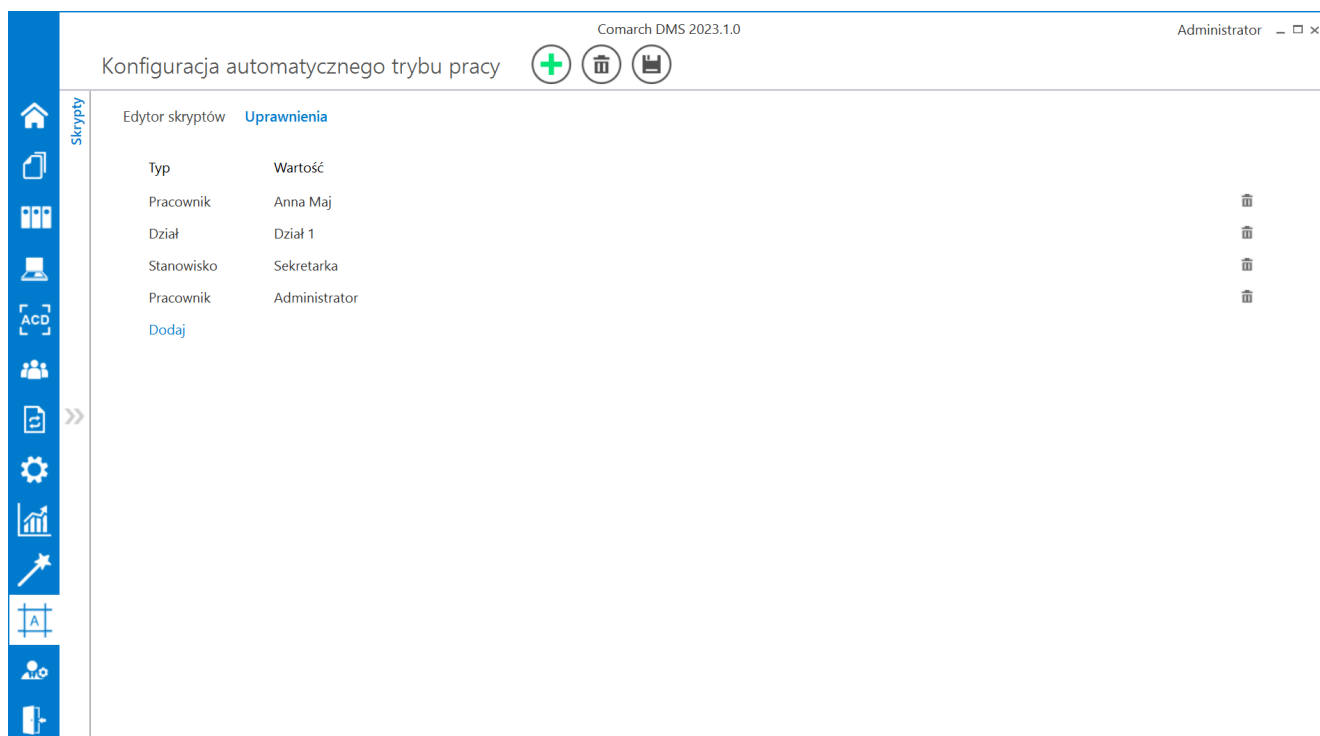
Zakładka „Uprawnienia” w ramach okna „Konfiguracja automatycznego trybu pracy” przed dodaniem uprawnień

Na zakładce dostępne są następujące pola:

- Nazwa spółki – pole dostępne jedynie w trybie wielospółkowym Comarch DMS; należy w nim dokonać wyboru spółki, z której pochodzi operator, któremu mają zostać nadane uprawnienia
- Typ – w ramach tego pola możliwy jest wybór pracownika, działu lub stanowiska, dla którego mają zostać przyznane uprawnienia do danego programu
- Wartość – w zależności od wyboru dokonanego w polu „Typ” w ramach tego pola można wybrać danego pracownika/dział/stanowisko, któremu zostaną przyznane uprawnienia do danego programu
- Po nadaniu uprawnień należy zapisać zmiany, klikając na

przycisk  [Zapisz]. Wówczas operator zostaje przeniesiony na zakładkę „Edytor skryptów”, a w prawym górnym rogu okna „Konfiguracja automatycznego trybu

pracy” pojawia się przycisk  [Uruchom], służący do uruchomienia programu.



Zakładka „Uprawnienia” w ramach okna „Konfiguracja automatycznego trybu pracy” z dodanymi uprawnieniami – przed zapisaniem zmian

Uruchamianie i zatrzymywanie programu

Okno „Uruchom program”



Aby uruchomić program, należy kliknąć w ikonę [Uruchom], która jest widoczna:

- na zakładce [Dokumenty] – dla każdego użytkownika, któremu przyznano uprawnienia do co najmniej jednego programu

Comarch DMS 2023.1.0 Administrator

Dokumenty

Szukaj

Wszystkie (59)

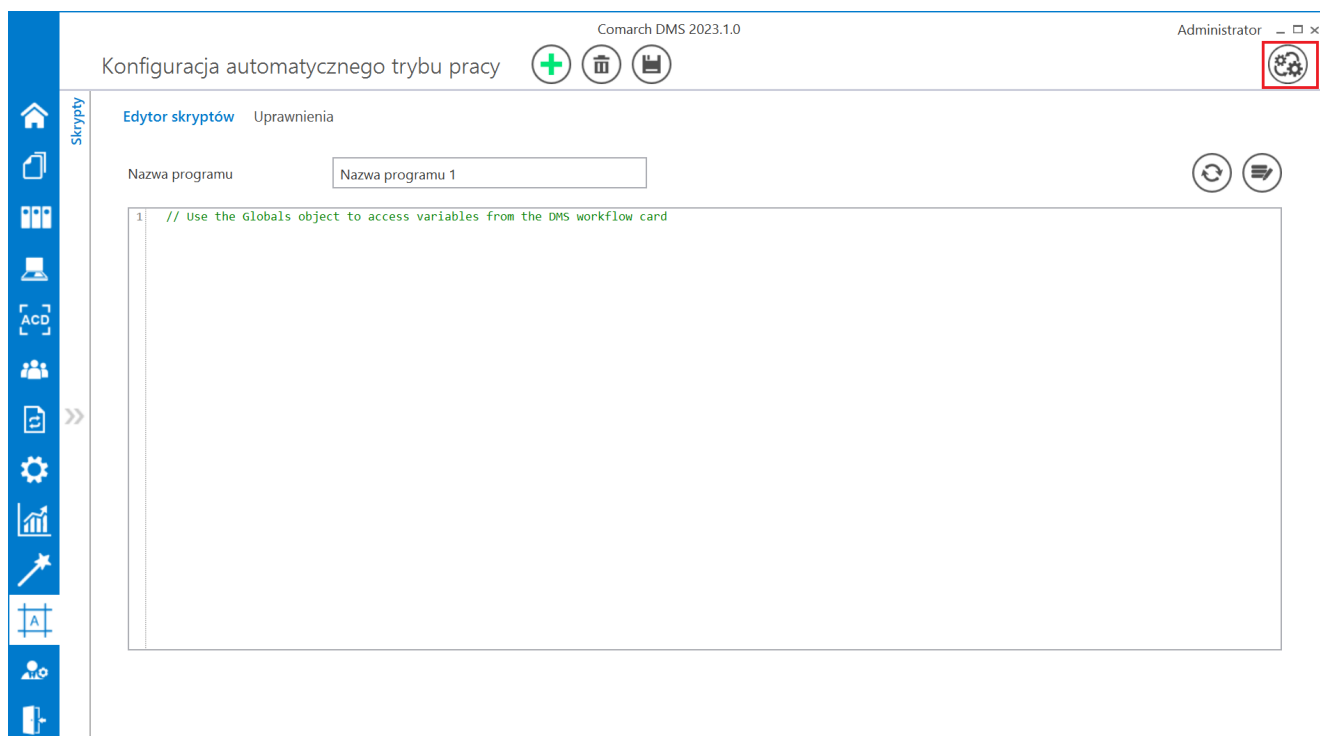
- Faktury - firma A (16)
- Faktury - firma B (16)
- Faktury - firma C (27)

NUMER	NAZWA OBIEGU	NAZWA ETAPU	DATA PRZEKAZANIA	STATUS	ETAP
FAFC/27/6/2023	Faktury - firma C	1	2023-06-22	Niezakończony	Przypisany do Ciebie
FAFC/26/6/2023	Faktury - firma C	1	2023-06-22	Niezakończony	Przypisany do Ciebie
FAFC/25/6/2023	Faktury - firma C	1	2023-06-22	Niezakończony	Przypisany do Ciebie
FAFC/24/6/2023	Faktury - firma C	1	2023-06-22	Niezakończony	Przypisany do Ciebie
FAFC/23/6/2023	Faktury - firma C	1	2023-06-22	Niezakończony	Przypisany do Ciebie
FAFC/22/6/2023	Faktury - firma C	1	2023-06-22	Niezakończony	Przypisany do Ciebie
FAFC/21/6/2023	Faktury - firma C	1	2023-06-22	Niezakończony	Przypisany do Ciebie
FAFC/20/6/2023	Faktury - firma C	1	2023-06-22	Niezakończony	Przypisany do Ciebie
FAFC/19/6/2023	Faktury - firma C	1	2023-06-22	Niezakończony	Przypisany do Ciebie
FAFC/18/6/2023	Faktury - firma C	1	2023-06-22	Niezakończony	Przypisany do Ciebie
FAFC/17/6/2023	Faktury - firma C	1	2023-06-22	Niezakończony	Przypisany do Ciebie
FAFC/16/6/2023	Faktury - firma C	1	2023-06-22	Niezakończony	Przypisany do Ciebie
FAFC/15/6/2023	Faktury - firma C	1	2023-06-22	Niezakończony	Przypisany do Ciebie
FAFC/14/6/2023	Faktury - firma C	1	2023-06-22	Niezakończony	Przypisany do Ciebie
FAFC/13/6/2023	Faktury - firma C	1	2023-06-22	Niezakończony	Przypisany do Ciebie
FAFC/12/6/2023	Faktury - firma C	1	2023-06-22	Niezakończony	Przypisany do Ciebie

1 >>

Widok listy dokumentów dla użytkownika, któremu nadano uprawnienia do programu – w prawym górnym rogu widoczna jest ikona „Uruchom”

- W ramach okna „Konfiguracja automatycznego trybu pracy”, dla operatora, któremu przyznano uprawnienia do tego okna oraz do co najmniej jednego programu



Widok okna „Konfiguracja automatycznego trybu pracy” dla użytkownika, któremu nadano uprawnienia do programu – w prawym górnym rogu widoczna jest ikona „Uruchom”



Po naciśnięciu ikony [Uruchom] zostaje otwarte okno „Uruchom program”. W ramach tego okna operator może uruchomić program. W oknie znajdują się dwa pola:

- Uruchom program – w ramach tego pola użytkownik może wybrać program, który chce uruchomić, z rozwijanej listy

Uwaga

W ramach pola „Uruchom program” widoczne są tylko te programy, do których przyznano danemu operatorowi uprawnienia na zakładce „Uprawnienia” w oknie „Konfiguracja automatycznego trybu pracy” (zob. [Zakładka „Uprawnienia”](#)).

The screenshot shows a dialog box titled "Uruchom program". It contains a dropdown menu labeled "Uruchom program" with the text "Faktury z firm A,B,C" and a downward arrow. The dropdown menu is open, showing a list of options: "Faktury z firm A,B,C" (highlighted in blue), "Instrukcje obsługi", and "Potwierdzenia odbioru". To the left of the dropdown is a checkbox labeled "Zabezpiecz hasłem" which is currently unchecked. At the bottom right of the dialog are two buttons: "Uruchom" and "Anuluj".

Wybór z listy programu, który ma zostać uruchomiony

- Zabezpiecz hasłem – pole domyślnie odznaczone; jeżeli zostanie zaznaczone, wówczas zatrzymanie pracy programu będzie wymagało podania hasła zalogowanego operatora w oknie „Automatyczny tryb pracy”.

The screenshot shows the same "Uruchom program" dialog box. The dropdown menu is now closed and shows "Faktury z firm A,B,C". The checkbox labeled "Zabezpiecz hasłem" is now checked. The "Uruchom" and "Anuluj" buttons remain at the bottom right.

Okno „Uruchom program” z zaznaczonym checkboxem „Zabezpiecz hasłem”

- Po wyborze programu operator powinien nacisnąć przycisk

Uruchom

[Uruchom] – wówczas okno „Uruchom program” zostanie zamknięte, a otworzy się okno „Automatyczny tryb pracy” i wybrany program rozpocznie działanie.

- Natomiast jeżeli operator nie chce jednak uruchomić

Anuluj

programu, powinien kliknąć przycisk [Anuluj] – wtedy okno „Uruchom program” zostanie zamknięte, a program się nie uruchomi.

Okno „Automatyczny tryb pracy”

Po naciśnięciu w oknie „Uruchom program” przycisku

Uruchom

[Uruchom] zostaje otwarte okno „Automatyczny tryb pracy”.

W ramach okna „Automatyczny tryb pracy” znajduje się nazwa programu, który został uruchomiony, jest również podawany czas, który zajęło dotychczas wykonywanie programu.

Aby zakończyć działania programu, należy kliknąć przycisk

Zatrzymaj program

[Zatrzymaj program], znajdujący się w dolnej części okna.

Uwaga

Jeśli okno „Automatyczny tryb pracy” jest włączone, użytkownik nie może pracować z Comarch DMS.

Automatyczny tryb pracy

Uruchomiono program: Faktury z firm A,B,C

Czas wykonywania: 00.00:00:02

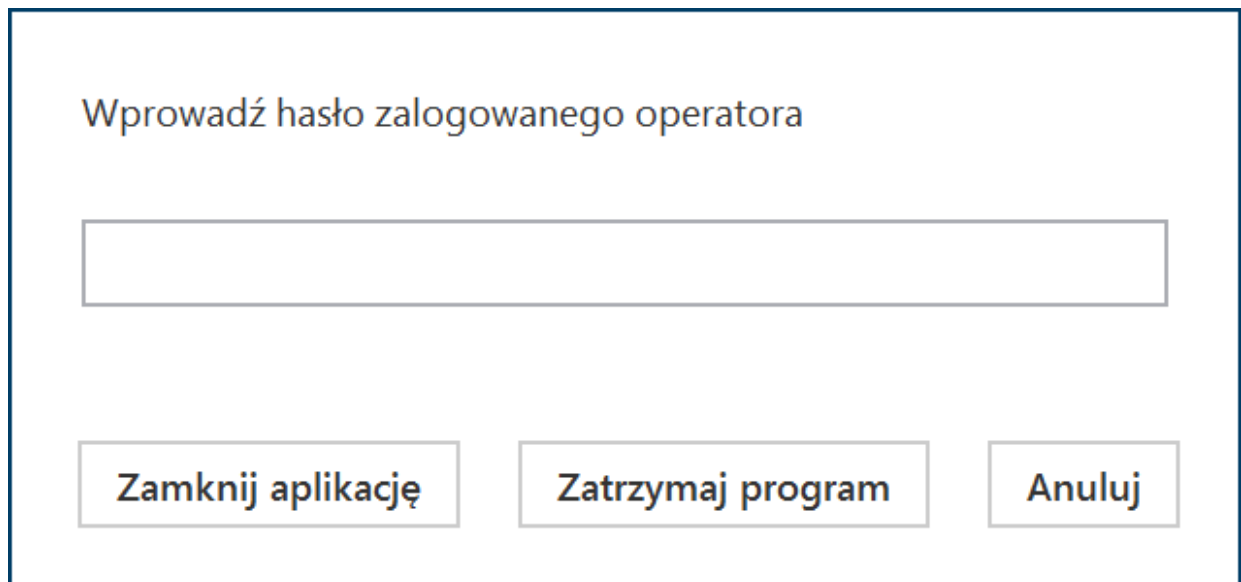
Zatrzymaj program

Okno „Automatyczny tryb pracy”

Zatrzymaj program

Kiedy operator naciśnie przycisk
[Zatrzymaj program]:

- jeśli w oknie „Uruchom program” nie zaznaczono checkboxa „Zabezpiecz hasłem” – wykonywanie skryptu zostanie przerwane, okno „Automatyczny tryb pracy” – zamknięte, a operator może znowu pracować z Comarch DMS;
- jeśli w oknie „Uruchom program” zaznaczono checkbox „Zabezpiecz hasłem” – wyświetlone zostaje okno, w ramach którego należy wprowadzić hasło zalogowanego operatora.



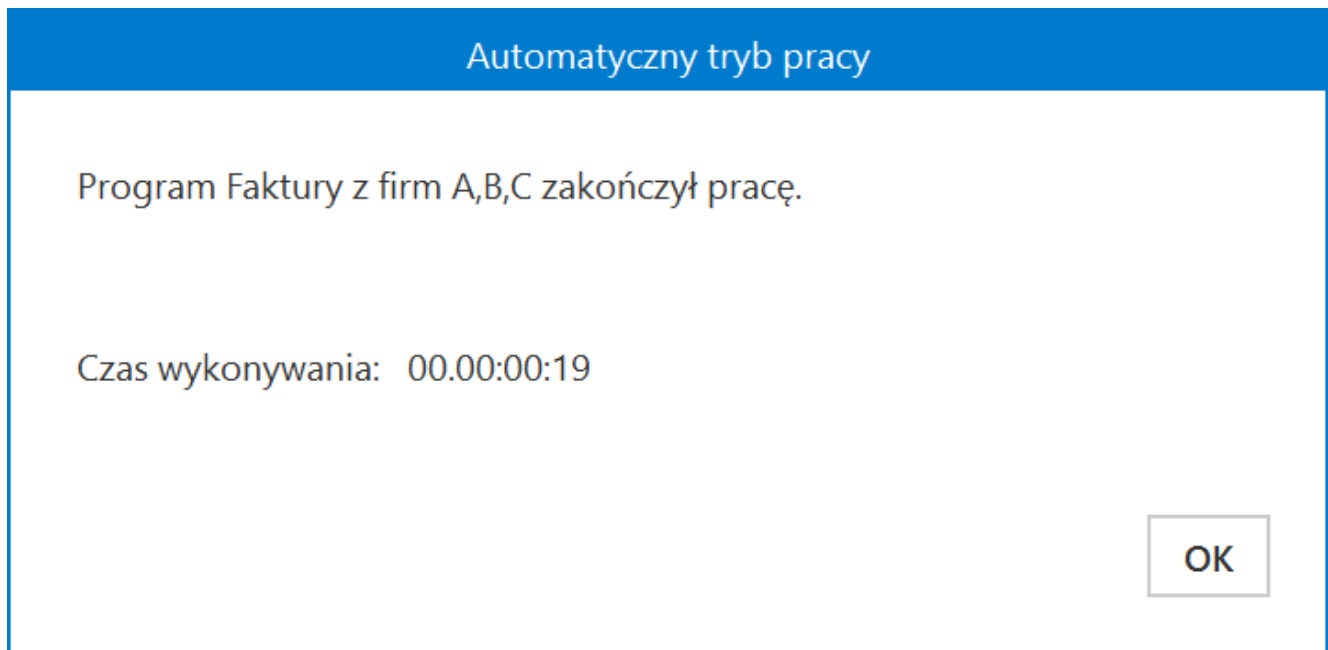
Wprowadź hasło zalogowanego operatora

Zamknij aplikację Zatrzymaj program Anuluj

Okno, w którym należy wpisać hasło zalogowanego operatora, aby zatrzymać program

W oknie dostępne są następujące przyciski:

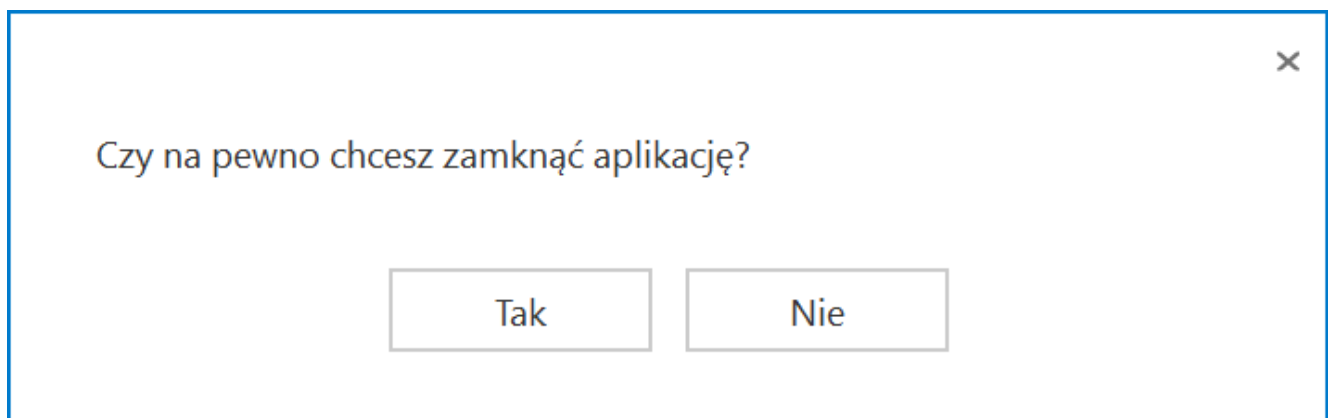
-  [Zatrzymaj program] – kiedy operator wybierze ten przycisk, wówczas wykonywanie skryptu zostanie zatrzymane, okno „Automatyczny tryb pracy” – zamknięte, zostanie również wyświetlone okno informujące o zakończeniu działania programu, w którym należy kliknąć przycisk „Ok”. Operator będzie mógł znowu pracować z Comarch DMS;



Okno wyświetlane po podaniu hasła operatora i zatrzymaniu programu

Zamknij aplikację

- [Zamknij aplikację] – po naciśnięciu tego przycisku zostaje wyświetlone następujące okno:



Okno wyświetlone po wyborze przycisku „Zamknij aplikację”

W ramach okna znajdują się dwa przyciski:

- Tak – aplikacja Comarch DMS zostanie zamknięta
- Nie – jeżeli operator wybierze tę opcję, wówczas aplikacja Comarch DMS pozostaje otwarta, a skrypt nadal jest wykonywany

Anuluj

- [Anuluj] – jeśli użytkownik kliknie ten przycisk, wówczas okno do wprowadzenia hasła zostanie zamknięte, a skrypt będzie nadal wykonywany;
- Jeżeli operator nie wprowadzi hasła lub wprowadzi błędne hasło, a następnie kliknie w przycisk

Zatrzymaj program

[Zatrzymaj program], wówczas zostanie wyświetlona informacja o nieprawidłowym hasle, a skrypt nadal będzie wykonywany.

Wprowadź hasło zalogowanego operatora

Wprowadzone hasło jest niepoprawne

Zamknij aplikację

Zatrzymaj program

Anuluj

Informacja wyświetlana, jeśli nie wprowadzono hasła lub wprowadzono błędne hasło

Klasa ACD

W obiekcie Globals dodano klasę ACD, która określa grupę metod wykonywanych jedynie w kontekście okna ACD. W klasie ACD

udostępniono następujące metody:

GenerateDMSDocuments (int) – metoda umożliwia wciśnięcie przycisku „Generuj” na zakładce „Automatyczne generowanie dokumentów”.

- zwraca wyniki:
 - 0 – operacja została wykonana poprawnie
 - -1 – kiedy wystąpił błąd

Int GenerateDMSDocuments()

Return: 0 – ok, -1 – error

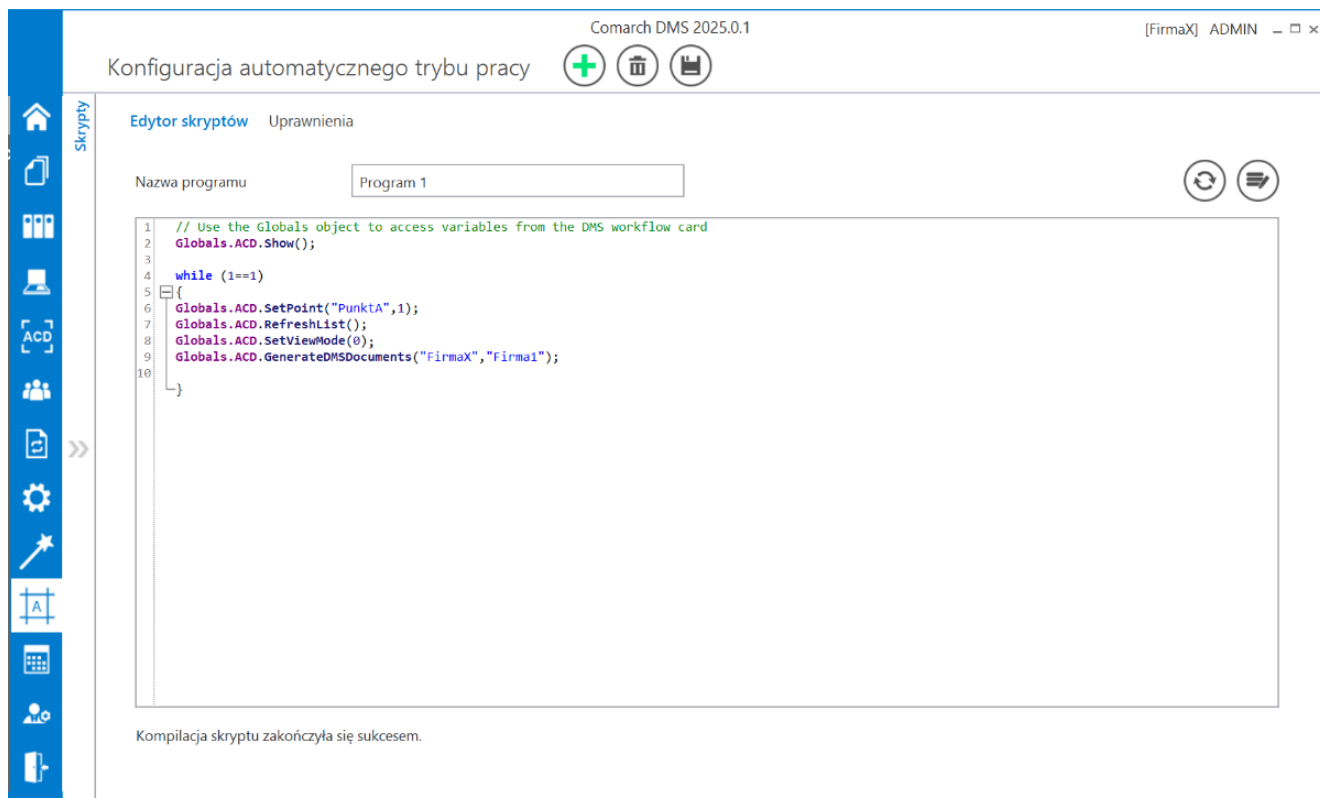
Uwaga

Od wersji 2024.0.1 na zakładce „Edytor skryptów” na zakładce



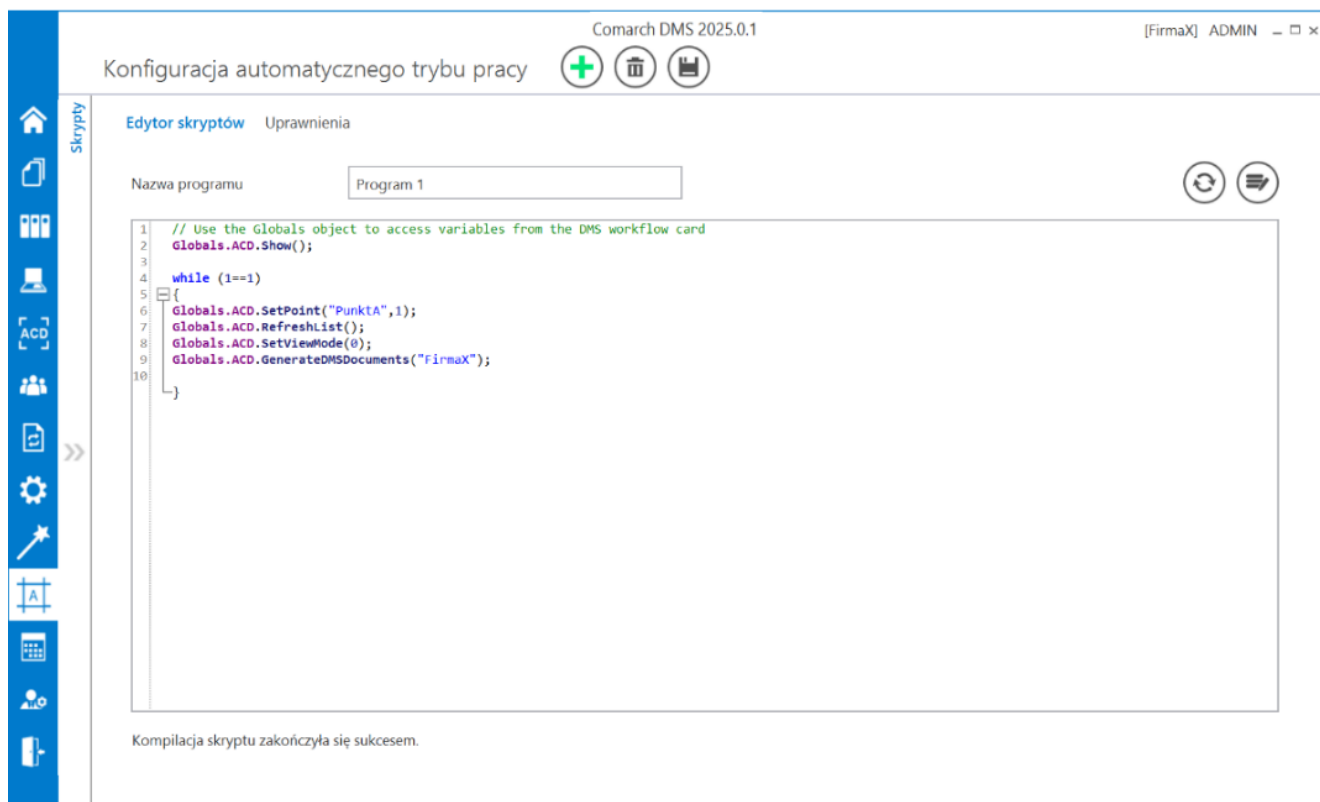
[Konfiguracja automatycznego trybu pracy] można korzystać z następujących wariantów metody GenerateDMSDocuments():

- GenerateDMSDocuments(string company, string department)
 - W ramach parametru string company należy wprowadzić nazwę spółki, a w ramach parametru string department – nazwę działu danej spółki



Przykład skryptu z metodą `GenerateDMSDocuments(string company, string department)`

- `GenerateDMSDocuments(string company)`
 - W ramach parametru `string company` należy wprowadzić nazwę spółki



Przykład skryptu z metodą `GenerateDMSDocuments(string company)`

`[/alert]`

GetDocumentList – metoda umożliwia pobranie informacji o zaczytanych dokumentach

- Zwraca wyniki:
 - 0 – operacja została wykonana poprawnie
 - -1 – wystąpił błąd

```
Int GetDocumentList(ref ACDDocument[])
```

```
Struct ACDDocument
```

```
(
```

```
string Name,
```

```
byte Status
```

```
)
```

Name: document name

Status: 0 - correct, 1 – incorrect

Return: 0 – ok, -1 - error

GetNumberItems (ref long) – metoda, która sprawdza liczbę pobranych dokumentów w punkcie ACD

- brak parametrów wejściowych
- zwraca wyniki:
 - Liczba pobranych dokumentów
 - 0 – operacja została wykonana poprawnie
 - -1 – kiedy metoda została wywołana w niewłaściwym kontekście

int GetNumberItems(ref long)

long: number of documents

return: 0 – ok, -1 - error

RefreshList() – metoda umożliwia wciśnięcie przycisku „Odśwież” na zakładce „Automatyczne generowanie dokumentów”

- brak parametrów wejściowych
- zwraca wyniki:
 - 0 – operacja została wykonana poprawnie
 - -1 – kiedy wystąpił błąd

Int RefreshList()

return: 0 – ok, -1 - error

SelectDocument (int) – metoda, która umożliwia podświetlenie dokumentu na liście (jeżeli lista prezentuje dokumenty z

checkboxami – zaznaczenie checkboxa).

- Parametry wejściowe:
 - Identyfikator dokumentu
- Zwraca wyniki:
 - 0 – operacja została wykonana poprawnie
 - -1 – wystąpił błąd

int SelectDocument(int)

int: document index

return: 0 – ok, -1 - error

SetPoint (string,byte) – metoda, która odpowiada za wskazanie punktu ACD i włączenie lub wyłączenie automatycznego wysyłania dokumentów do Comarch OCR

- Parametry wejściowe:
 - Pierwszy parametr jest wymagany, należy wprowadzić w nim nazwę punktu ACD, w ramach którego będzie się odbywało przetwarzanie dokumentów
 - Drugi parametr jest opcjonalny, dotyczy kwestii włączenia procesowania dokumentów w punkcie ACD, przyjmuje następujące wartości:

0 – procesowanie dokumentów jest wyłączone (dotyczy punktów ACD, które nie obsługują OCR)

1 – procesowanie dokumentów jest włączone

- Zwraca wyniki:
 - 0 – operacja została wykonana poprawnie
 - -1 – kiedy wystąpił błąd
 - -2 – kiedy punkt ACD o podanej nazwie nie istnieje

int SetPoint(string,byte)

string: ACD point name

byte:1 -turn on ACD proces, 0 – turn off ACD proces

return: 0 – ok, - 1 – error, -2 – ACD point not exists

SetViewMode (byte) – metoda, która odpowiada za tryb filtrowania pobranych dokumentów

- Parametry wejściowe:
 - 0 – generowanie wszystkich dokumentów;
 - 1 – generowanie jedynie wypełnionych dokumentów;
 - 2 – generowanie wybranych dokumentów;
- Zwraca wyniki:
 - 0 – operacja została wykonana poprawnie
 - -1 – wystąpił błąd

int SetViewMode(byte)

byte: 0 – all documents, 1 – correct documents, 2 – selected documents

return: 0 – ok, - 1 - error

Show – metoda, która otwiera obszar ACD

- brak parametrów wejściowych
- zwraca wyniki:
 - 0 – operacja została wykonana poprawnie
 - 1 – operator nie jest uprawniony do okna ACD – jeśli ta wartość została zwrócona, w aplikacji zostaje wyświetlony komunikat „Operator nie posiada uprawnień do automatycznego generowania dokumentów”

int Show()

return: 0 – ok, -1 - error

Uwaga

W wersji 2024.3.0 udostępniono metodę `Globals.ACD:int GetErrorList(ref ACDError[] errors)` w ramach skryptów C# na

zakładce  **[Konfiguracja automatycznego trybu pracy]**. Metoda zwraca listę ostatnich błędów (do 50 błędów) na bieżącym punkcie ACD, można ją zastosować dla punktów współpracujących z Comarch OCR oraz dla punktów współpracujących ze skrzynką pocztową.

Poniżej przedstawiono przykład z zastosowaniem nowej metody (została oznaczona zieloną czcionką) dla przykładowych punktów ACD typu Współpraca z Comarch OCR ("Faktury od Firmy A", "Faktury od Firmy B", "Faktury od Firmy C"):

```
Globals.ACD.Show();
```

```
while (1==1)
```

```
{
```

```
Globals.ACD.SetPoint("Faktury od Firmy A",1);
```

```
Globals.ACD.RefreshList();
```

```
Globals.ACD.GenerateDMSDocuments();
```

```
Globals.ACD.SetPoint("Faktury od Firmy B",1);
```

```
Globals.ACD.RefreshList();
```

```
ACDError[] errors;
```

```
Globals.ACD.GetErrorList(ref errors);
```

```
Globals.ACD.GenerateDMSDocuments();
```

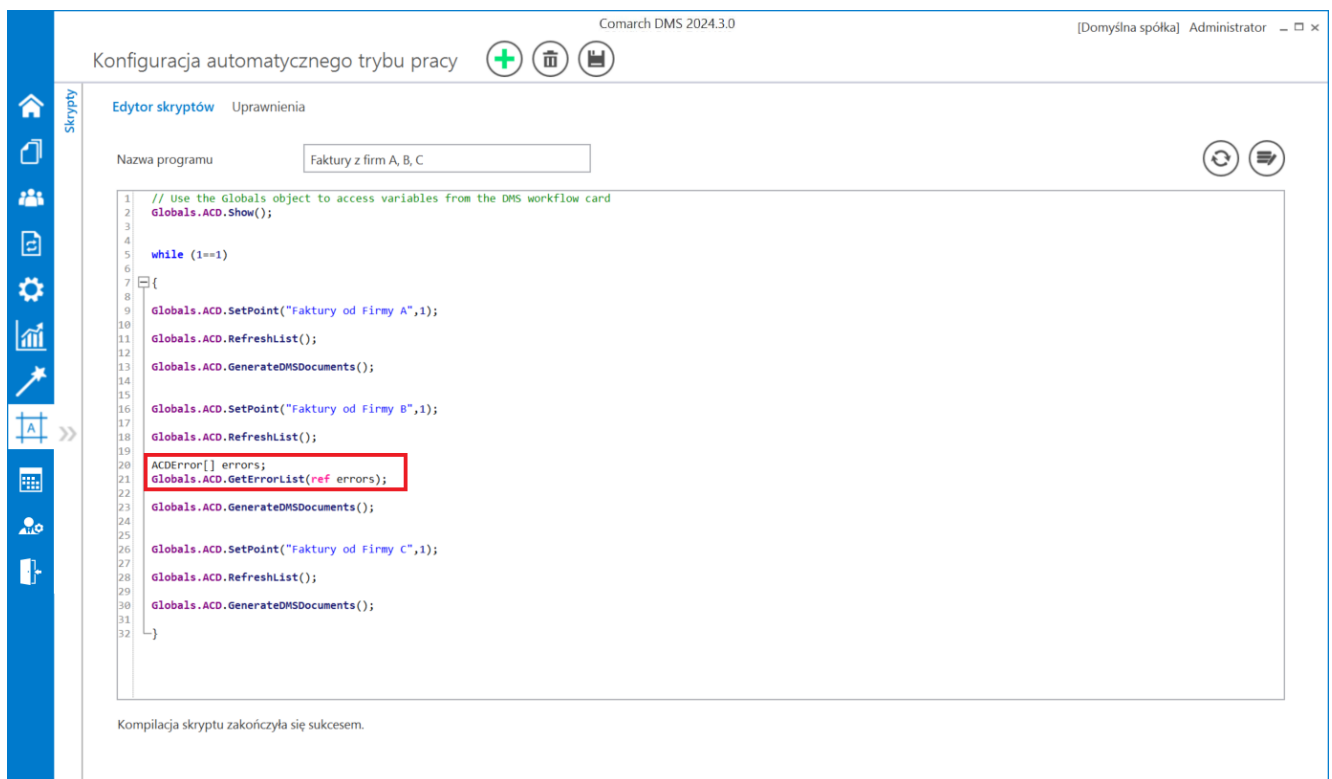


```
Globals.ACD.SetPoint("Faktury od Firmy C",1);
```

```
Globals.ACD.RefreshList();
```

```
Globals.ACD.GenerateDMSDocuments();
```

```
}
```



Zakładka „Konfiguracja automatycznego trybu pracy” z wprowadzonym kodem C# z nową metodą

Klasa Common

W ramach klasy Comon udostępniono następującą metodę:

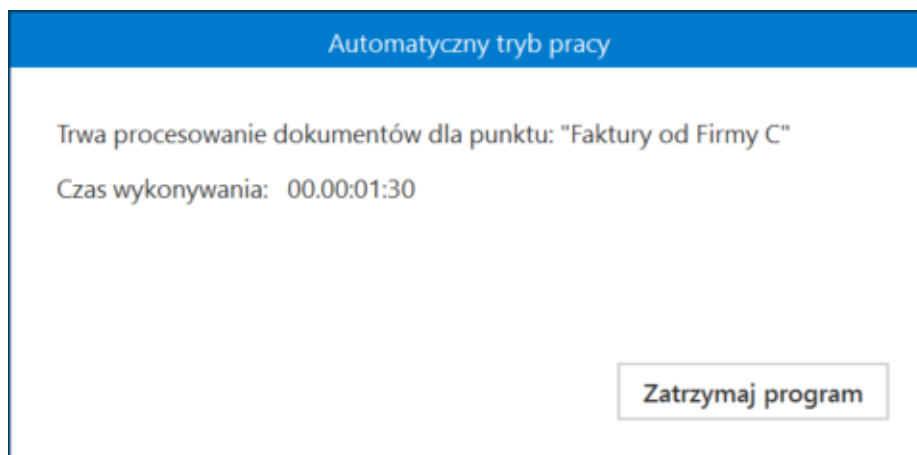
- `Trace(string)` – metoda umożliwia wyświetlanie dodatkowych informacji na oknie „Automatyczny tryb pracy”,`[/su_list]`

```
void Trace(string)
```

```
string:message
```

Przykład:

```
Globals.Common.Trace("Trwa procesowanie dokumentów dla punktu:  
\"Faktury od Firmy C\"");
```



Okno „Automatyczny tryb pracy” z przykładowym zastosowaniem metody Trace

Konfiguracja i uruchamianie automatycznego trybu pracy – przykład

- Administrator otworzył do edycji plik DMS.exe.config, znajdujący się w folderze z aplikacją desktop Comarch DMS i w ramach klucza „RPAFolderPath” wprowadził ścieżkę do aplikacji desktop Comarch DMS – w tym przypadku `<add key=“RPAFolderPath” value=“C:\dms202310”/>` – a później zapisał zmianę.

- Administrator dodał na zakładce  [Automatyczne generowanie dokumentów] trzy punkty ACD należące do typu „Współpraca z Comarch OCR” o następujących nazwach:

- Faktury od Firmy A (zgodnie z ustawieniami punktu dokumenty wygenerowane w tym punkcie mają trafiać do obiegu „Faktury – firma A”)
- Faktury od Firmy B (zgodnie z ustawieniami punktu dokumenty wygenerowane w tym punkcie mają trafiać do obiegu „Faktury – firma B”)
- Faktury od Firmy C (zgodnie z ustawieniami punktu dokumenty wygenerowane w tym punkcie mają trafiać do obiegu „Faktury – firma C”)

Punkt konfiguracyjny

Tryb współpracy

Ogólne

Lista

Kontrolki

Uprawnienia

Nazwa punktu

Faktury od Firmy A

Typ obiegu

Faktury - firma A

Typ rejestru VAT

Zakup

Domyślna forma płatności

Język dokumentu

Polski

Rozpoznawanie kodów kreskowych

Wyłączone

Zapisz

Zamknij

Ustawienia punktu ACD „Faktury od Firmy A” – zakładka „Ogólne”

Punkt konfiguracyjny

Tryb współpracy Ogólne Lista **Kontrolki** Uprawnienia

Skan faktury (Skan) 

Numer faktury (Numer) 

Wartość netto (netto) 

Wartość brutto (brutto) 

[Dodaj](#)

Pobierz plik

```

1 select OCD_SourceContent, OCD_FileName
2 from do.OC_Documents
3 where OCD_ID=@Id1@

```

[Test zapytania SQL](#)

Składnia:
 SELECT Varbinary, varchar(100)
 Varbinary – obraz skanu zapisany w bazie
 varchar(100) - typ pliku
 Przykład:
 BEGIN TRY
 select scan, 'pdf' from tabela
 where Id = @Id1@
 END TRY
 BEGIN CATCH

[Zapisz](#) [Zamknij](#)

Ustawienia punktu ACD „Faktury od Firmy A” – zakładka „Kontrolki”

- następnie administrator przeszedł na zakładkę  [Konfiguracja automatycznego trybu pracy] i kliknął przycisk  [Dodaj].

Wskazówka

Aby zobaczyć przykład, kliknij na film poniżej.

<https://pomoc.comarch.pl/dms/wp-content/uploads/2023/07/robo1.mp4>

- Na zakładce „Edytor skryptów” w ramach pola „Nazwa programu” wpisał „Faktury z firm A,B,C”, później na zakładce „Uprawnienia” nadał uprawnienia do programu dla siebie i kilku innych operatorów i kliknął  [Zapisz].

Wskazówka

Aby zobaczyć przykład, kliknij na filmy poniżej.

<https://pomoc.comarch.pl/dms/wp-content/uploads/2023/07/robo2.mp4>

<https://pomoc.comarch.pl/dms/wp-content/uploads/2023/07/robo3.mp4>

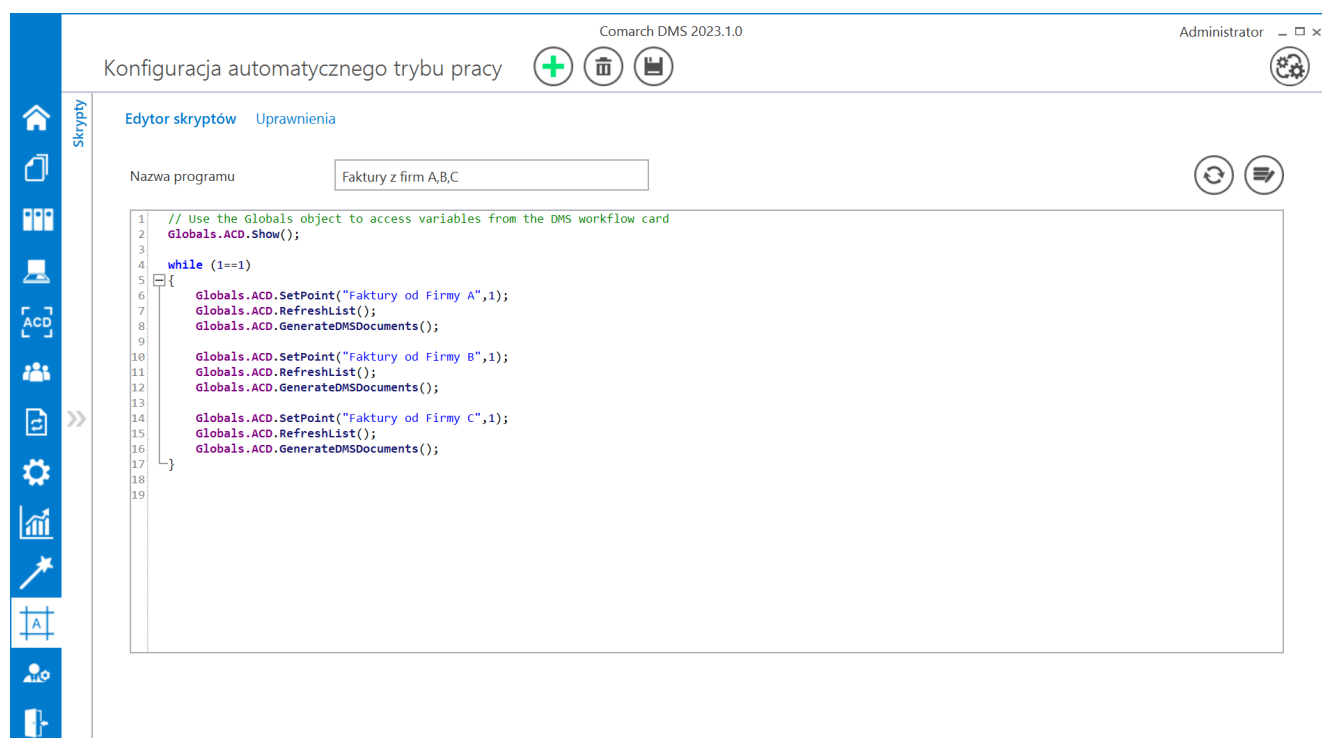
- Na zakładce „Edytor skryptów” w polu edytora administrator wprowadził następujący kod:

```
Globals.ACD.Show();
```

```
while (1==1)
{
Globals.ACD.SetPoint("Faktury od Firmy A",1);
Globals.ACD.RefreshList();
Globals.ACD.GenerateDMSDocuments();
```

```
Globals.ACD.SetPoint("Faktury od Firmy B",1);
Globals.ACD.RefreshList();
Globals.ACD.GenerateDMSDocuments();
```

```
Globals.ACD.SetPoint("Faktury od Firmy C",1);
Globals.ACD.RefreshList();
Globals.ACD.GenerateDMSDocuments();
}
```



Wprowadzanie kodu do edytora skryptów



- Następnie administrator nacisnął przycisk [Kompiluj i zapisz]. Kompilacja przebiegła pomyślnie, więc w dolnej części okna wyświetlona została informacja „Kompilacja skryptu zakończyła się sukcesem”.

Wskazówka

Aby zobaczyć przykład, kliknij na film poniżej.

<https://pomoc.comarch.pl/dms/wp-content/uploads/2023/07/robo4.mp4>

- Następnie administrator zapisał zmiany, klikając

przycisk  [Zapisz]. Później nacisnął przycisk  [Uruchom] i w ramach otwartego okna „Uruchom program” wybrał w polu „Uruchom program” program „Faktury z firm A,B,C” i (nie zaznaczając checkboxa „Zabezpiecz

hasłem”), a później kliknął przycisk [Uruchom].

Uruchom

Wskazówka

Aby zobaczyć przykład, kliknij na film poniżej.

<https://pomoc.comarch.pl/dms/wp-content/uploads/2023/07/robo5.mp4>

- wówczas administrator został przeniesiony na zakładkę „Automatyczne generowanie dokumentów”, gdzie otworzyło się okno „Automatyczny tryb pracy”, a w tle odbywało się inicjowanie dokumentów.

Wskazówka

Aby zobaczyć przykład, kliknij na film poniżej.

<https://pomoc.comarch.pl/dms/wp-content/uploads/2023/07/robo6.mp4>

- Gdy wszystkie dokumenty zostały wygenerowane, administrator kliknął przycisk  [Zatrzymaj program]. Okno „Automatyczny tryb pracy” zostało zamknięte.

<https://pomoc.comarch.pl/dms/wp-content/uploads/2023/07/robo7.mp4>

- Następnie administrator przeszedł na zakładkę  [Dokumenty w obiegu] – na liście dokumentów widoczne były wszystkie dokumenty wygenerowane w wyniku działania

programu „Faktury z firm A,B,C”. Po wejściu do przykładowego dokumentu można zauważyć, że na dokument zmapowano również wartości odczytane przez OCR, zgodnie z ustawieniami punktów ACD.

<https://pomoc.comarch.pl/dms/wp-content/uploads/2023/07/robo8.mp4>

Comarch DMS 2023.1.0 Administrator

FA/43/6/2023

Data utworzenia : 23-06-2023

Karta obiegu

Skan faktury

Numer faktury

FS-3_20_knt_2022-03

FS-1/20/KNT/02

Wartość netto

50,000.00

Wartość brutto

61,500.00

Podgląd

Faktura VAT

nr FS-1/20/KNT/02

Data wystawienia: 2020-02-03

Sprzedawca: COMARCH S.A.

Nabywca: ALOZA

Lipowa 11/15

99-400 Warszawa

NIP: 7752141104

Zamówienie:

Lp.	Kod	Nazwa towaru	PKT/VAT	PCN	Ilość / m.	Cena netto	Wartość netto	VAT
1	T1				100 szt.	500.00	50,000.00	23%

Forma płatności	Termin	Kwota Właściwa	Stawka	Netto	VAT	Brutto
Gotówka	2020-07-10	61,500.00 PLN		50,000.00	11,500.00	61,500.00
Razem:			23%	50,000.00	11,500.00	61,500.00
W tym:						

Przykładowy dokument wygenerowany w automatycznym trybie pracy

Film

W artykule [Automatyczne generowanie dokumentów dzięki mechanizmom RPA. Prezentacja funkcjonalności](#) dostępny jest również film z lektorem przedstawiający funkcjonalność.

Generowanie faktury zakupu w Comarch ERP XL z wykorzystaniem skryptów C#

Spis treści artykułu

- [Informacje podstawowe](#)
- [Konfiguracja typu obiegu](#)
- [Generowanie dokumentu FZ do Comarch ERP XL – jak działa](#)

Informacje podstawowe

Uwaga

Aby przykład działał poprawnie, konieczne jest spełnienie następujących warunków:

1. [Instalacja systemu Comarch ERP XL na serwerze IIS](#)
2. [Poprawna konfiguracja współpracy pomiędzy systemami Comarch ERP XL oraz Comarch DMS](#)
3. [Nadanie użytkownikowi IISUSER uprawnień do zapisu](#) w katalogu `bin\Scriptcs_bin` oraz katalogu `bin\.scriptcs_cache`, które znajdują się w [plikach aplikacji serwerowej Comarch DMS](#)
4. [Skopiowanie biblioteki `cdn\_api\[wersja\].net.dll` do katalogu `bin\Scriptcs\_bin`](#), biblioteka znajduje się w katalogu, w którym zainstalowano Comarch ERP XL, w przykładach wykorzystano `cdn_api20232.net.dll`

5. Należy sprawdzić, czy w katalogu bin aplikacji web Comarch DMS znajduje się plik `cdn_sys.dll`, jeżeli jest należy usunąć go z tej lokalizacji.

Skrypty C# zostały wykorzystane w kontrolkach typu Własna akcja i Komunikat. Skrypt dla kontrolki typu własna akcja utworzono na podstawie wzorca API XL

Wskazówka

Rekomendujemy wykorzystanie wzorca API XL jako elementu bazowego na potrzeby własnych rozwiązań.

Wzorzec API XL dostępny jest na indywidualnych stronach dla Partnerów w obszarze Comarch DMS\Przykłady\ERP XL: [Wzorzec API XL](#)

Konfiguracja typu obiegu

Przykład pokazuje, w jaki sposób administrator Comarch DMS może skonfigurować definicję obiegu, aby generować dokument „Faktura zakupu” do systemu Comarch ERP XL, bez pomocy kontrolki typu Dokument ERP XL. W tym celu należy:

1. **dodać nowy typ obiegu** w ramach zakładki  [Definicje obiegu dokumentów]
2. **nadać nazwę typowi obiegu** – np. *Generowanie faktury zakupu* i prefix – np. GEN
3. **nadać danemu administratorowi uprawnienie do inicjowania dokumentu** na zakładce „Ustawienia obiegu” w sekcji „Uprawnienia do inicjowania dokumentu”

Generowanie faktury zakupu

Prefix: GEN

KARTA OBIEGU

SCHEMAT OBIEGU

USTAWIENIA OBIEGU

Ogólne

Lista dokumentów jako książka nadawcza

Ustawienia dla aplikacji mobilnej

Filtry

Uprawnienia do inicjowania dokumentu

Nazwa spółki	Typ	Wartość	
Firma X	Pracownik	Administrator systemu	
Dodaj			
Obserwator dokumentu			

Nadawanie uprawnień do inicjowania dokumentu dla administratora – przykład

4. W ramach zakładki „Schemat obiegu” utworzyć etap początkowy „Generuj dokument” i etap końcowy „Koniec” oraz dodać uprawnienie dla danego administratora do etapów.

Generowanie faktury zakupu

Prefix: GEN

←

🏠

⚙️

🗑️

KARTA OBIEGU

SCHEMAT OBIEGU

USTAWIENIA OBIEGU

🏠

📄

👤👤

ACD

👥

📄

⚙️

👤⚙️

📱

Etap

Szukaj

🗑️ ⬆️ ⬇️ +

Generuj dokument

Koniec

Etap

OGÓLNE

UPRAWNIENIA

PRZEKAZANIE

MAIL

KONTROLKI

Cofnij dokument

Właściciel dokumentu

Uprawnienia do etapu

Uprawnienia dodatkowe

Zawężanie uprawnień na wcześniejszym etapie

Dostęp do dok. dla operatora niepracującego z dok.

Uprawnienia

Nazwa spółki	Typ	Wartość
Firma X	Pracownik	Administrator systemu
Dodaj		

Dodawanie etapów i uprawnień

5. W ramach zakładki „Karta obiegu” **dodać następujące kontrolki:**

A Kontrolka typu Kontrahent – „**Kontrahent**” (identyfikator: Kontrahent)

Kontrahent	
Nazwa wyświetlana	Kontrahent
Nazwa (identyfikator)	Kontrahent
Prezentuj na całej szerokości	☐
Ustaw kontrolkę na początku wiersza	☐
Nazwa spółki	Firma X
Pokaż na liście	☐
Inicjowanie wartości	
Inicjowanie	Brak
Zmiana wartości (Obserwator)	
Kontrolki powiązane	Dodaj
Wzór na wartość	Wyrażenie
	Ustaw

Kontrolka „Kontrahent”

B Kontrolka typu Dane tabelaryczne – „Elementy”

(identyfikator: Elementy) – w ramach kontrolki należy:

- w zakładce „Listy” dodać następujące listy:
 - „Jm” z wartością „Jednostki miary” w polu „Inicjowanie”
 - „VAT” z wartością „Stawki VAT” w polu „Inicjowanie”

Dane tabelaryczne

OGÓLNE LISTY INICJOWANIE WARTOŚCI OGRANICZENIE EDYCJI

Lista 1

Nazwa

Jm

Inicjowanie

Jednostki miary

Rodzaj identyfikatora

Int

Nazwa spółki

Firma X

Pokaż

Usuń

Tryb pracy

Lista

Lista 2

Nazwa

VAT

Inicjowanie

Stawki VAT

Rodzaj identyfikatora

Int

Nazwa spółki

Firma X

Pokaż

Usuń

Tryb pracy

Lista

Dodaj

Kontrolka „Elementy”, zakładka „Listy”

- w zakładce „**Ogólne**” dodać następujące kolumny:
 - kolumna typu **Towar** – „**Towar**” (identyfikator: Towar)
 - kolumna typu **Liczba stałoprzecinkowa** – „**Ilość**” (identyfikator: Ilosc)
 - kolumna typu **Lista** – „**Jm.**” (identyfikator: Jm) z wartością „**Jm**” wybraną w polu „**Listy**”
 - kolumna typu **Lista** – „**VAT**” (identyfikator: VAT) z wartością „**VAT**” wybraną w polu „**Listy**”
 - kolumna typu **Liczba stałoprzecinkowa** –

- „Cena netto” (identyfikator: CenaNetto)
- kolumna typu Liczba stałoprzecinkowa – „Cena brutto” (identyfikator: CenaBrutto)
- kolumna typu Liczba stałoprzecinkowa – „Wartość Netto” (identyfikator: WartoscNetto)
- kolumna typu Liczba stałoprzecinkowa – „Wartość Brutto” (identyfikator: WartoscBrutto)

Dane tabelaryczne

OGÓLNE

LISTY

INICJOWANIE WARTOŚCI

OGRANICZENIE EDYCJI

Nazwa wyświetlana

Elementy

Nazwa (identyfikator)

Elementy

Prezentuj na całej szerokości

☒

Ustaw kontrolkę na początku wiersza

☐

Liczba prezentowanych wierszy

5

Kolumny

Kolumna 1

Towar

Nazwa kolumny

Towar

Identyfikator kolumny

Towar

Nazwa spółki

Firma X

Kolumna 2

Liczba stałoprzecinkowa

Nazwa kolumny

Ilość

Identyfikator kolumny

Ilość

Liczba miejsc po przecinku, do której będzie zaokrąglana wartość

2

Kolumna 3

Lista

Nazwa kolumny

Jm.

Identyfikator kolumny

Jm

Listy

Jm

Kontrolka „Elementy”, zakładka „Ogólne” – widok kolumn 1-3

Dane tabelaryczne

Kolumna 4

Lista



Nazwa kolumny

VAT

Identyfikator kolumny

VAT

Listy

VAT



Kolumna 5

Liczba stałoprzecinkowa



Nazwa kolumny

Cena netto

Identyfikator kolumny

CenaNetto

Liczba miejsc po przecinku, do której będzie zaokrąglana wartość

2



Kolumna 6

Liczba stałoprzecinkowa



Nazwa kolumny

Cena brutto

Identyfikator kolumny

CenaBrutto

Liczba miejsc po przecinku, do której będzie zaokrąglana wartość

2



Kolumna 7

Liczba stałoprzecinkowa



Nazwa kolumny

Wartość Netto

Identyfikator kolumny

WartoscNetto

Liczba miejsc po przecinku, do której będzie zaokrąglana wartość

2



Kolumna 8

Liczba stałoprzecinkowa



Nazwa kolumny

Wartość Brutto

Identyfikator kolumny

WartoscBrutto

Liczba miejsc po przecinku, do której będzie zaokrąglana wartość

2



Kontrolka „Elementy”, zakładka „Ogólne” – widok kolumn

- w zakładce „Inicjowanie wartości” :
 - w ramach pola „Kontrolki powiązane” kliknąć przycisk „Dodaj” i wybrać kontrolkę „Elementy”
 - w ramach pola „Wzór na wartość” wybrać opcję „SQL OD”, kliknąć „Ustaw”, a następnie **wprowadzić następujące zapytanie:**

```

if @^InitSenderControlName@ = 'Elementy'
Begin

    declare @vat decimal(22,4) = 0.0
    declare @netto decimal (22,4) = 0.0
    declare @brutto decimal (22,4) = 0.0
    declare @ilosc decimal (22,4) = 0.0

    if @Elementy_Column@ in (2,4,5)
    Begin
        select @netto = CenaNetto, @ilosc = Ilosc
    from   @Elementy@
    where  POS = @Elementy_Row@
        select @vat = case
            when VAT = 1 then 23
            when VAT = 2 then 8
            when VAT = 3 then 0
            when VAT = 4 then 0
            when VAT = 5 then 0
            when VAT = 6 then 7
            when VAT = 7 then 5
            else 0 end
        from   @Elementy@ where POS = @Elementy_Row@
        set @brutto = ((@netto*@vat)/100)+@netto

        update @Elementy@ set CenaBrutto = @brutto,
        WartoscBrutto = @ilosc * @brutto,
        WartoscNetto = @ilosc * @netto where POS =
        @Elementy_Row@
        select * from @Elementy@
    End

```

```

if @Elementy_Column@ = -1
Begin
    update @Elementy@ set Ilosc = 1, JM = 1, VAT = 1
    where POS = @Elementy_Row@
    select * from @Elementy@
End
End

```

Wprowadzone zapytanie należy **zapisać**, klikając w przycisk „Zapisz” w oknie „Wyrażenie kontrolki”

Dane tabelaryczne

OGÓLNE
LISTY
INICJOWANIE WARTOŚCI
OGRANICZENIE EDYCJI

Inicjowanie wartości

Inicjowanie

Brak

▼

Kontrolki powiązane

Elementy (Elementy)

Dodaj

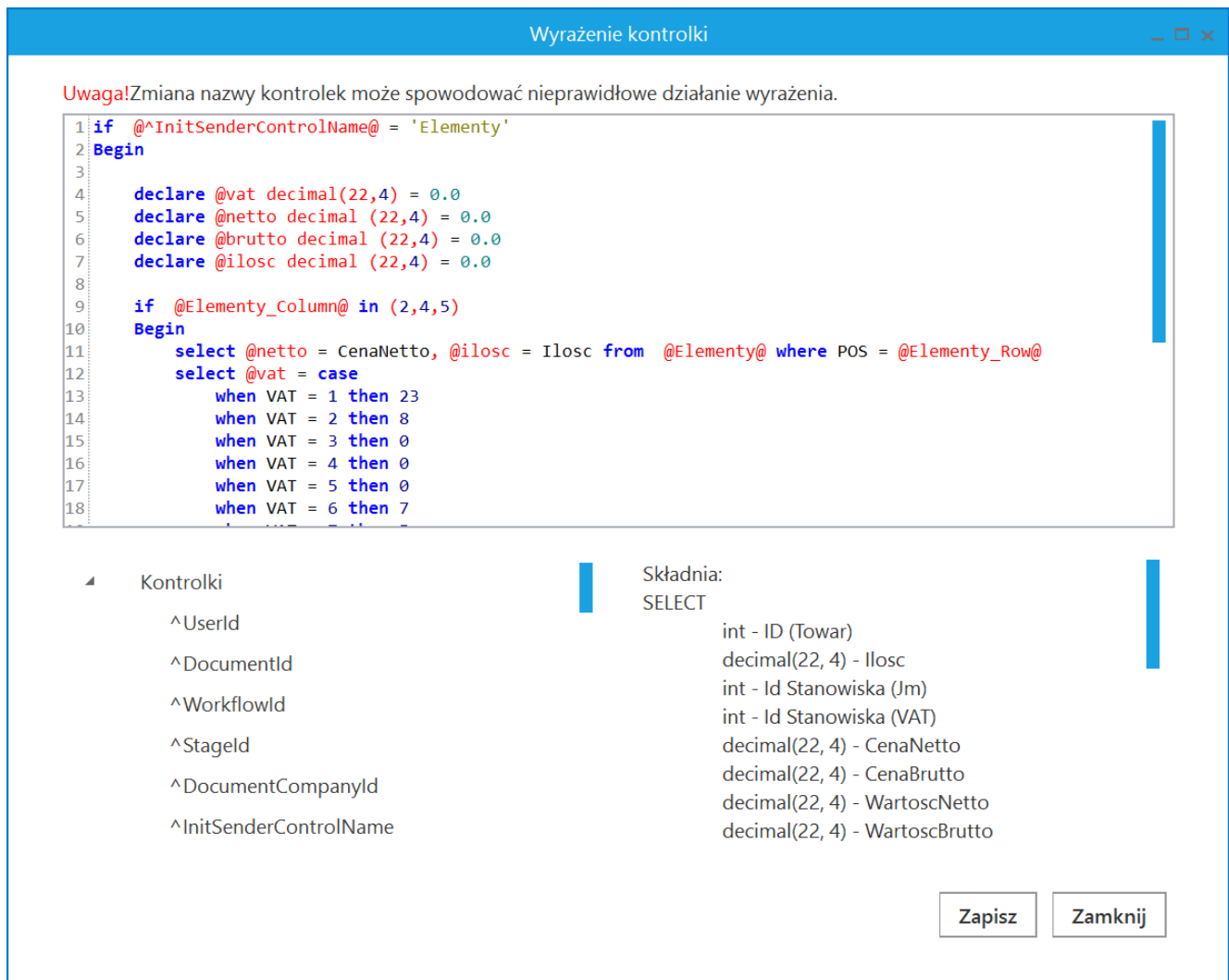
Wzór na wartość

SQL OD

▼

Ustaw

Kontrolka „Elementy”, zakładka „Inicjowanie wartości”



Okno „Wyrażenie kontrolki” z wprowadzonym zapytaniem

- w zakładce „Ograniczenie Edycji” zaznaczyć checkboxy na obydwóch etapach dla kolumn: „Cena netto”, „Cena brutto”, „Wartość Netto”, „Wartość Brutto”

Dane tabelaryczne

OGÓLNE

LISTY

INICJOWANIE WARTOŚCI

OGRANICZENIE EDYCJI

Zaznaczenie opcji spowoduje zablokowanie funkcjonalności lub edycji poniższych parametrów.

Etap	☐ Zaznacz wszystko Zablokuj dodanie wierszy	☐ Zaznacz wszystko Zablokuj usuwanie wierszy	☐ Zaznacz wszystko Towar Towar	☐ Zaznacz wszystko Ilość Ilość	☐ Zaznacz wszystko Jm Jm.	☐ Zaznacz wszystko VAT VAT	☐ Zaznacz wszystko CenaNetto Cena netto	☒ Zaznacz wszystko CenaBrutto Cena brutto	☒ Zaznacz wszystko WartoscNetto Wartość Netto	☒ Zaznacz wszystko WartoscBrutto Wartość Brutto
Generuj dokument	☐	☐	☐	☐	☐	☐	☐	☒	☒	☒
Koniec	☐	☐	☐	☐	☐	☐	☐	☒	☒	☒

Kontrolka „Elementy”, zakładka „Ograniczenie Edycji”

C kontrolka typu Własna akcja – „Generuj dokument” (identyfikator: GenerujDokument) – w ramach definicji kontrolki należy kliknąć w link „Ustaw” w polu „C# Script”, następnie w otwartym oknie „Skrypt C#” w polu „Miejsce wykonania skryptu” wybrać „Proces (exe)” (jeśli istnieje opcja wyboru) i wprowadzić następujący kod:

```
using System.Threading;
using System.Runtime.InteropServices;
using cdn_api;

static ThreadLocal<bool> threadToClarionAttached = new
ThreadLocal<bool>(() => false);

[DllImport("ClaRUN.dll")]
private static extern void AttachThreadToClarion(int flag);
public void AttachThreadToClarion()
{
    try
    {
        if (!threadToClarionAttached.Value)
        {
            AttachThreadToClarion(1); // C8
            threadToClarionAttached.Value = true;
        }
    }
    catch (DllNotFoundException ex)
    {
        // -----
        // Tu przekaz informacje o statusie działania skryptu
        // -----
        Globals.MainFrame.GenerujDokument_CSSSCRIPT.Text =
ex.Message;
    }
}
```

```

AttachThreadToClarion();

int _lSesjaID = 0;
string blad = "";

try
{
    int documentId = 0;
    int wersjaApi = odpowiedniawersjaapi;

    var loginInfo = new XLLoginInfo_odpowiedniawersjaapi()
    {
        Wersja = odpowiedniawersjaapi,
        ProgramID = "Comarch DMS",
        Winieta = -1,
        TrybWsadowy = 1,
        Baza = "nazwafirmy",
        OpeIdent = "Login",
        OpeHaslo = "hasło",
        SerwerKlucza = @"serwerklucza"
    };

    var XLLoginResult = cdn_api.cdn_api.XLLLogin(loginInfo, ref
_lSesjaID);
    if (XLLoginResult != 0)
    {
        blad = "Funkcja XLLLogin zwróciła błąd nr: " +
XLLoginResult;
        throw new Exception();
    }

    // -----
-----
    // Tu dodaj kod obsługi API
    // -----
-----

        var dokumentNagInfo = new
XLDokumentNagInfo_odpowiedniawersjaapi
    {
        Wersja = odpowiedniawersjaapi,

```



```

        Typ = 1521, //FZ
        KntTyp = 32,
        KntNumer = Globals.MainFrame.Kontrahent.Id ?? 0, //
Pobierz GidNumer kontrahenta
        RodzajZakupu = 1,
        //Avista = 1
    };

    var XLNowyDokumentResult = cdn_api.cdn_api.XLNowyDokument
(_lSesjaID, ref documentId, dokumentNagInfo);
    if (XLNowyDokumentResult != 0)
    {
        blad = "Funkcja XLNowyDokument zwróciła błąd nr: " +
XLNowyDokumentResult;
        throw new Exception();
    }

    //var element;

    for (int i=0; i<Globals.MainFrame.Elementy.RowCount; i++)
    {

        var element = new
XLDokumentElemInfo_odpowiedniawersjaapi();

        element.Wersja = odpowiedniawersjaapi;
        element.TwrTyp = 16;

        element.Ilosc =
Globals.MainFrame.Elementy.Items[i].Ilosc.Value.ToString();
        element.TwrNumer =
Globals.MainFrame.Elementy.Items[i].Towar.Id;
        element.JmZ =
Globals.MainFrame.Elementy.Items[i].Jm.Text;
        element.Cena =
Globals.MainFrame.Elementy.Items[i].CenaNetto.Value.ToString()
;

        var XLDodajPozycjeResult =
cdn_api.cdn_api.XLDodajPozycje(documentId, element);
        if (XLDodajPozycjeResult != 0)

```

```

        {
            blad = "Funkcja XLDodajPozycje zwróciła błąd nr: "
+ XLDodajPozycjeResult;
            throw new Exception();
        }
    }

    var zamkniecieDokumentuInfo = new
XLZamkniecieDokumentuInfo_odpowiedniawersjaapi
    {
        Wersja = odpowiedniawersjaapi,
        Tryb = 1
    };

    var XLZamknijDokumentResult =
cdn_api.cdn_api.XLZamknijDokument(documentId,
zamkniecieDokumentuInfo);
    if (XLZamknijDokumentResult != 0)
    {
        blad = "Funkcja XLZamknijDokument zwróciła błąd nr: "
+ XLZamknijDokumentResult;
        throw new Exception();
    }
}
catch (Exception ex)
{
    if (blad == "")
        blad = "Nieokreślony błąd: " + ex.Message;
}
finally
{
    var XLLogoutResult = cdn_api.cdn_api.XLLogout(_lSesjaID);
    if (XLLogoutResult != 0)
    {
        blad = "Funkcja XLLogout zwróciła błąd nr: " +
XLLogoutResult;
    }
    // -----
    -----
    // Tu przekaż informacje o statusie działania skryptu

```

```
// -----
-----
if ( blad == "" )
    Globals.MainFrame.GenerujDokument_CSSSCRIPT.Text =
"Operacja zakończona sukcesem.";
else
    Globals.MainFrame.GenerujDokument_CSSSCRIPT.Text =
blad;
}
```

gdzie **należy zmienić następujące wartości:**

odpowiedniawersjaapi – **wersja API ERP XL** zgodna z wersją pliku
cdn_api[wersja].net.dll

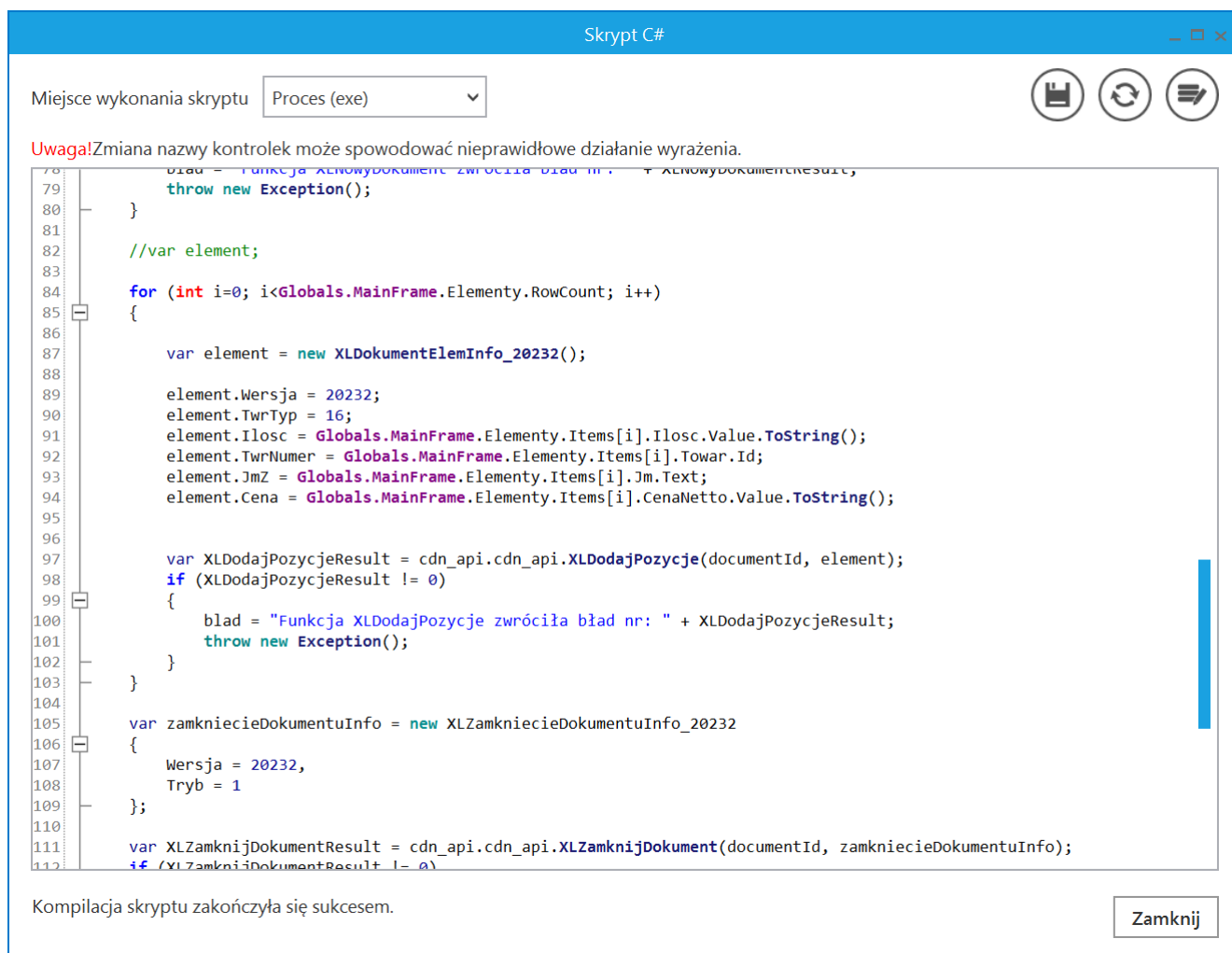
nazwafirmy – **nazwa Firmy Comarch ERP XL** (nazwę firmy można
sprawdzić w managerze baz Comarch ERP XL)

Login – **login operatora Comarch ERP XL**

hasło – **hasło operatora Comarch ERP XL**

serwerklucza – **serwer klucza licencji**

Po wprowadzeniu kodu należy **nacisnąć przycisk**  **[Kompiluj i zapisz]**, a następnie, gdy w dolnej części okna zostanie wyświetlony napis „Kompilacja skryptu zakończyła się sukcesem” nacisnąć przycisk **„Zamknij”**.



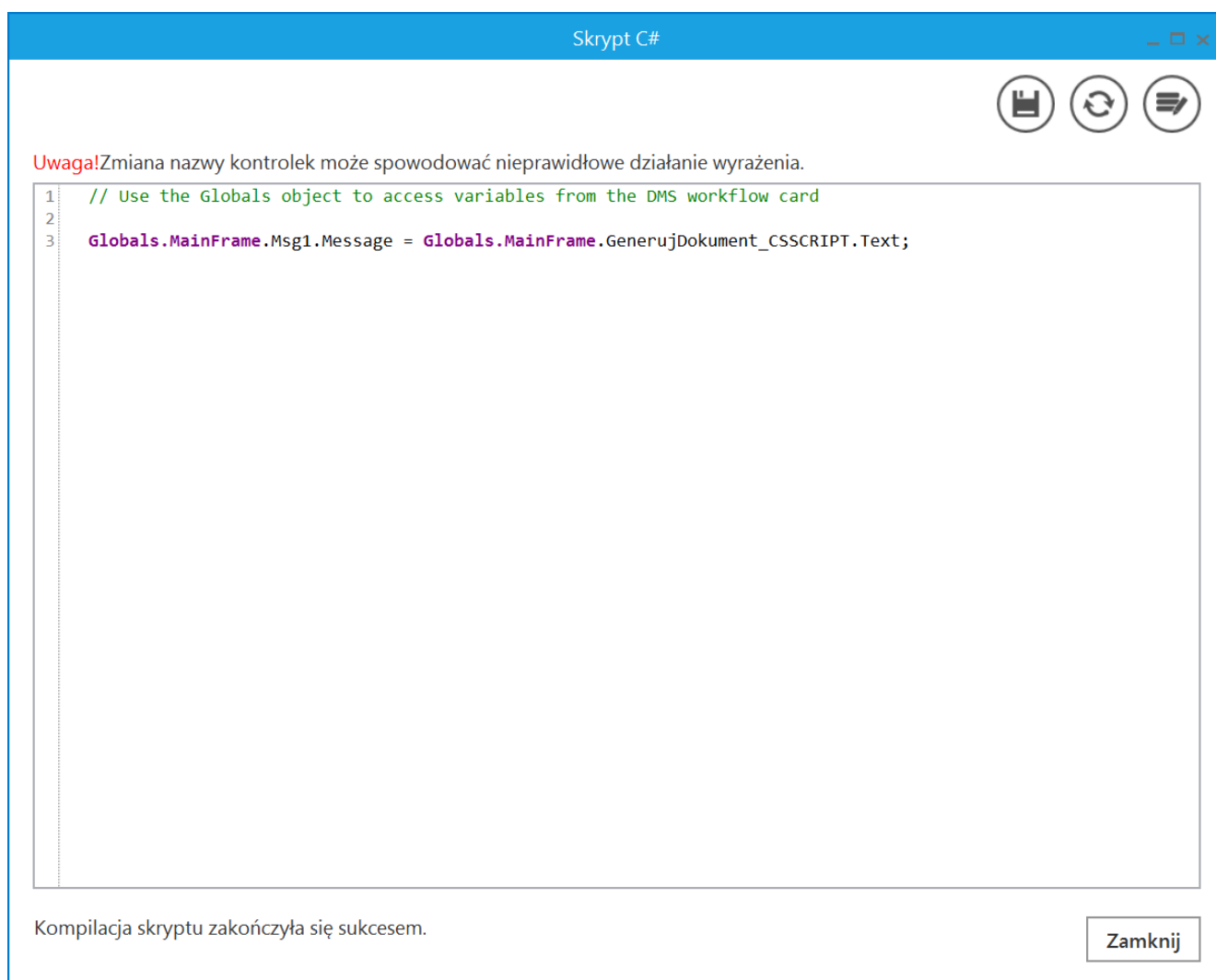
Widok okna „Skrypt C#” po skompilowaniu skryptu

D Kontrolka typu Komunikat – należy w ramach pola:

- „Nazwa (identyfikator)” wprowadzić nazwę „**Msg1**”
- „Tytuł” wprowadzić tekst „**Status wykonania API ERP XL**”
- „Kontrolki powiązane” kliknąć w link „**Dodaj**” i wybrać kontrolkę „**Generuj dokument**”
- „Wzór na wartość” wybrać wartość „**C# Script**”, kliknąć „**Ustaw**”, a następnie wprowadzić następujący kod:

```
Globals.MainFrame.Msg1.Message
= Globals.MainFrame.GenerujDokument_CSSSCRIPT.Text;
```

Po wprowadzeniu kodu należy **nacisnąć przycisk**  **[Kompiluj i zapisz]**, a następnie, gdy w dolnej części okna zostanie wyświetlony napis „Kompilacja skryptu zakończyła się sukcesem” nacisnąć przycisk **„Zamknij”**.



Widok okna „Skrypt C#” po skompilowaniu skryptu

6. **zapisać typ obiegu** za pomocą przycisku  **[Zapisz]**

Zapisywanie typu obiegu

Generowanie dokumentu FZ do Comarch ERP XL – jak działa

Po zdefiniowaniu typu obiegu operator przechodzi na zakładkę



[Dokumenty], klika na typ obiegu „**Generowanie faktury**

zakupu” i naciska przycisk  **[Dodaj]**. Następnie kolejno:

1. **zapisuje** nowy dokument, naciskając  **[Zapisz]**

2. **wybiera kontrahenta** w ramach kontrolki „**Kontrahent**”

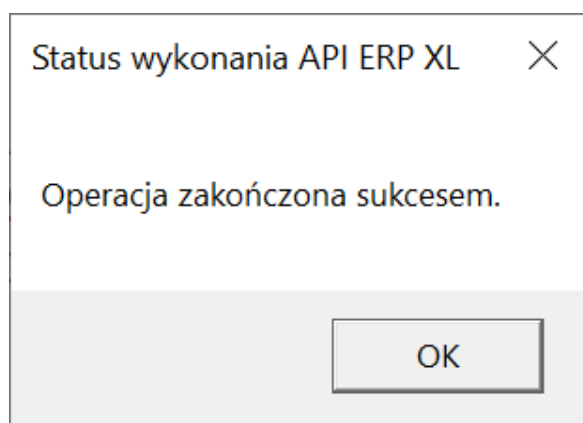
3. w ramach kontrolki „**Elementy**” dodaje towar w kolumnie „**Towar**”, wybiera opcję w kolumnach „**Jm.**” i „**VAT**” oraz wpisuje wartość w kolumnie „**Cena netto**” (jeśli poprawnie skonfigurowano obieg, wówczas kolumny „**Cena brutto**”, „**Wartość Netto**” i „**Wartość Brutto**” powinny zostać uzupełnione automatycznie)

Uwaga

Należy zwrócić uwagę, aby wartość wybrana w kolumnie „VAT” dla danego towaru była zgodna ze stawką przypisaną do tego towaru w Comarch ERP XL.

4. naciska przycisk **„Generuj dokument”** – wówczas następuje generowanie dokumentu – poniżej przycisku widoczne są poruszające się punkty

5. **Jeśli generowanie dokumentu zakończyło się powodzeniem**, wówczas na środku ekranu wyświetlany jest następujący komunikat:



Komunikat wyświetlany, jeśli generowanie dokumentu zakończyło się powodzeniem

Film

Aby zobaczyć przebieg generowania dokumentu do Comarch ERP XL, kliknij w poniższy film:

<https://pomoc.comarch.pl/dms/wp-content/uploads/2024/06/genxl.mp4>

Wygenerowany dokument jest już **widoczny w Comarch ERP XL**:

Zamówienia [FirmaKSI] - Comarch ERP XL 2024.0.3

System Ogólne Listy Dokumenty Zadania Sztuczna inteligencja Ołno Pomoc Lista dokumentów handlowych, magazynowych i tra Dokument - Faktura zakupu [FZ-11/24] - zostanie z

Drukuj Wyślij Rysuj wykres Wykonanie raportu Eksport do arkusza Poprzednia pozycja z listy Następną pozycją z listy Statystyka dokumentu Dekret wyników Stany towaru na magazynach Historia towaru Historia kontrahenta Struktura rabatów Historia zapisów Pomoc Wytnij Kopiuć Wklej Edycja

Lista dokumentów handlowych, magazynowych

Dokument - Faktura zakupu [FZ-11/24] - zostanie zmieniony

Ogólne Kontrahent Kaucje Płatności Nagłówki VAT Intrastat KSeF Atrybuty Załączniki

Kontrahent: FIR1 Kraków

Kraj: PL Polska

Transakcja: Krajowa

Dostawa:

Rodzaj transportu: 3 Transport drogowy

Płatność: Gotówka 0 dni Do 05.06.2024

Netto: 400,00
Brutto: 492,00
Kaucje: 0,00
Zapłacono: 0,00
Do zapłaty: 492,00

Lp.	Towar	Ilość	Jm.	Cena netto	Cena brutto	Nazwa
00001	T1	2,0000	szt.	200,0000 PLN	246,0000 PLN	Towar 1

Filtr: Docelowy MAG Aktualizuj kaucje

Firma1 - Administrator systemu Środa, 05.06.2024 16:18

Dokument wygenerowany z Comarch DMS do Comarch ERP XL

Rozpoczynasz pracę z Comarch DMS i chcesz dowiedzieć się, jak korzystać z programu? A może masz już podstawową wiedzę o Comarch DMS i chcesz dowiedzieć się więcej?

[Sprawdź Szkolenia Comarch DMS!](#)

Zatwierdzanie faktury zakupu w Comarch ERP XL z wykorzystaniem skryptów C#

Spis treści artykułu

- [Informacje podstawowe](#)
- [Konfiguracja typu obiegu](#)
- [Zatwierdzanie dokumentu FZ w Comarch ERP XL – jak działa](#)

Informacje podstawowe

Uwaga

Aby przykład działał poprawnie, konieczne jest spełnienie następujących warunków:

1. [Instalacja systemu Comarch ERP XL na serwerze IIS](#)
2. [Poprawna konfiguracja współpracy pomiędzy systemami Comarch ERP XL oraz Comarch DMS](#)
3. [Nadanie użytkownikowi IISUSER uprawnień do zapisu w katalogu bin\Scriptcs_bin oraz katalogu bin\.scriptcs_cache, które znajdują się w plikach aplikacji serwerowej Comarch DMS](#)
4. [Skopiowanie biblioteki cdn_api\[wersja\].net.dll do katalogu bin\Scriptcs_bin, biblioteka znajduje się w katalogu, w którym](#)

zainstalowano Comarch ERP XL, w przykładach wykorzystano `cdn_api20232.net.dll`

5. Należy sprawdzić, **czy w katalogu bin aplikacji web Comarch DMS znajduje się plik `cdn_sys.dll`**, jeżeli jest należy **usunąć go** z tej lokalizacji.

Skrypty C# zostały wykorzystane w kontrolkach typu Własna akcja i Komunikat. Skrypt dla kontrolki typu własna akcja utworzono na podstawie wzorca API XL

Wskazówka

Rekomendujemy wykorzystanie wzorca API XL jako elementu bazowego na potrzeby własnych rozwiązań.

Wzorzec API XL dostępny jest na indywidualnych stronach dla Partnerów w obszarze Comarch DMS\Przykłady\ERP XL: [Wzorzec API XL](#)

Konfiguracja typu obiegu

Przykład pokazuje, **w jaki sposób administrator Comarch DMS może skonfigurować definicję obiegu, aby zatwierdzić dokument „Faktura zakupu” w systemie Comarch RP XL z poziomu Comarch DMS**. W tym celu należy:

1. **dodać nowy typ obiegu** w ramach zakładki  **[Definicje obiegu dokumentów]**
2. **nadać nazwę typowi obiegu** – np. *Zatwierdzanie faktury zakupu* i prefix – np. ZAT
3. **nadać danemu administratorowi uprawnienie do inicjowania**

dokumentu na zakładce „Ustawienia obiegu” w sekcji „Uprawnienia do inicjowania dokumentu”.

Zatwierdzanie faktury zakupu

Prefix: ZAT

←

KARTA OBIEGU

SCHEMAT OBIEGU

USTAWIENIA OBIEGU

Ogólne

▼

Lista dokumentów jako książka nadawcza

▼

Ustawienia dla aplikacji mobilnej

▼

Filtry

▼

Uprawnienia do inicjowania dokumentu

▲

Nazwa spółki	Typ	Wartość	
Firma X	Pracownik	Administrator systemu	
Dodaj			
Obserwator dokumentu ▼			

Nadawanie uprawnień do inicjowania dokumentu dla administratora – przykład

4. W ramach zakładki „Schemat obiegu” utworzyć etap początkowy „Zatwierdzenie dokumentu” i etap końcowy „Koniec” oraz dodać uprawnienie dla danego administratora do etapów.

Zatwierdzanie faktury zakupu

Prefix: ZAT

KARTA OBIEGU SCHEMAT OBIEGU USTAWIENIA OBIEGU

Etap

Szukaj

Zatwierdzanie dokumentu

Koniec

Etap

OGÓLNE

UPRAWNIENIA

PRZEKAZANIE

MAIL

KONTROLKI

Cofnij dokument

Właściciel dokumentu

Uprawnienia do etapu

Uprawnienia dodatkowe

Zawężanie uprawnień na wcześniejszym etapie

Dostęp do dok. dla operatora niepracującego z dok.

Uprawnienia

Nazwa spółki	Typ	Wartość
Firma X	Pracownik	Administrator systemu

Dodaj

Dodawanie etapów i uprawnień

5. W ramach zakładki „Karta obiegu” **dodać następujące kontrolki:**

A Kontrolka typu Dokument elektroniczny – „Faktura zakupu” (identyfikator: FakturaZakupu) – w polu „Typ” należy wybrać „Faktura zakupu”

Dokument elektroniczny

Nazwa wyświetlana	Faktura zakupu
Nazwa (identyfikator)	FakturaZakupu
Nazwa spółki	Firma X
Typ	Faktura zakupu
Prezentuj na całej szerokości	☐
Ustaw kontrolkę na początku wiersza	☐
Pokaż na liście	☐
Inicjowanie wartości	
Inicjowanie	Brak
Zmiana wartości (Obserwator)	
Kontrolki powiązane	Dodaj
Wzór na wartość	Wyrażenie
	Ustaw

Kontrolka „Faktura zakupu”

B kontrolka typu Własna akcja – „Zatwierdź dokument” (identyfikator: ZatwierdźDokument) – w ramach definicji kontrolki należy kliknąć w link „Ustaw” w polu „C# Script”, następnie w otwartym **oknie „Skrypt C#”** w polu „Miejsce wykonania skryptu” wybrać „Proces (exe)” (jeśli istnieje opcja wyboru) i **wprowadzić następujący kod**:

```
using System.Threading;
using System.Runtime.InteropServices;
using cdn_api;
```

```
static ThreadLocal<bool> threadToClarionAttached
= new ThreadLocal<bool>(() => false);
```

```

[DllImport("ClaRUN.dll")]
private static extern void AttachThreadToClarion(int flag);
public void AttachThreadToClarion()
{
    try
    {
        if (!threadToClarionAttached.Value)
        {
            AttachThreadToClarion(1); // C8
            threadToClarionAttached.Value = true;
        }
    }
    catch (DllNotFoundException ex)
    {
        // -----
        // Tu przekaz informacje o statusie działania skryptu
        // -----
        //Globals.MainFrame.ZatwierdzDokument_CSSSCRIPT.Text =
ex.Message;
        Globals.MainFrame.ZatwierdzDokument_CSSSCRIPT.Text=ex.Messa
ge;
    }
}

```

```

AttachThreadToClarion();

```

```

int _lSesjaID = 0;
string blad = "";

```

```

try
{
    int documentId = 0;
    int wersjaApi = odpowiedniawersjaapi;

    var loginInfo = new XLLoginInfo_odpowiedniawersjaapi()
    {
        Wersja = odpowiedniawersjaapi,
        ProgramID = "Comarch DMS",
    }
}

```

```

        Winieta = -1,
        TrybWsadowy = 1,
        Baza = "nazwafirmy",
        OpeIdent = "Login",
        OpeHaslo = "hasło",
        SerwerKlucza = @"serwerklucza"
    };

        var XLLoginResult =
cdn_api.cdn_api.XLLogin(loginInfo, ref _lSesjaID);
        if (XLLoginResult != 0)
        {
            blad = "Funkcja XLLogin zwróciła błąd nr:
" + XLLoginResult;
            throw new Exception();
        }

        // -----
-----
        // Tu dodaj kod obsługi API

        var otworzDocumentInfo
= new XL0twarcieNagInfo_odpowiedniawersjaapi
        {
            Wersja = odpowiedniawersjaapi,
            GIDTyp = 1521,
            GIDNumer
= Globals.MainFrame.FakturaZakupu.Id ?? 0,
            Tryb = 2
        };

        var XL0twarcieInfoResult =
cdn_api.cdn_api.XL0tworzDokument(_lSesjaID, ref documentId, ot
worzDocumentInfo);
        if (XL0twarcieInfoResult != 0)
        {
            blad = "Funkcja XL0tworzDokument zwróciła
błąd nr: " + XL0twarcieInfoResult;
            throw new Exception();
        }

```

```

                                var zamkniecieDokumentuInfo
= new XLZamkniecieDokumentuInfo_odpowiedniawersjaapi
                                {
                                    Wersja = odpowiedniawersjaapi,
                                    Tryb = 0
                                };

                                var XLZamknijDokumentResult =
cdn_api.cdn_api.XLZamknijDokument(documentId, zamkniecieDokume
ntuInfo);
                                if (XLZamknijDokumentResult != 0)
                                {
                                    blad = "Funkcja XLZamknijDokument zwróciła
bład nr: " + XLZamknijDokumentResult;
                                    throw new Exception();
                                }
                                // -----
                                -----
                                }
                                catch (Exception ex)
                                {
                                    if (blad == "")
                                        blad = "Nieokreślony bład: " + ex.Message;
                                }
                                finally
                                {
                                    var XLLogoutResult = cdn_api.cdn_api.XLLogout(_lSesjaID);
                                    if (XLLogoutResult != 0)
                                    {
                                        blad = "Funkcja XLLogout zwróciła bład nr:
" + XLLogoutResult;
                                    }
                                    // -----
                                }
                                -----
                                // Tu przekaz informacje o statusie działania skryptu
                                // -----
                                -----

                                //Globals.MainFrame.ZatwierdzDokument_CSSSCRIPT.Text =
blad;

```



```

        if (blad == "")
            Globals.MainFrame.ZatwierdźDokument_CSSSCRIPT.Text
= "Operacja zakończona sukcesem";
        else
            Globals.MainFrame.ZatwierdźDokument_CSSSCRIPT.Text =
blad;

}

```

gdzie **należy zmienić następujące wartości:**

odpowiedniawersjaapi – **wersja API ERP XL** zgodna z wersją pliku **cdn_api[wersja].net.dll**

nazwafirmy – **nazwa Firmy Comarch ERP XL** (nazwę firmy można sprawdzić w managerze baz Comarch ERP XL)

Login – **login operatora Comarch ERP XL**

hasło – **hasło operatora Comarch ERP XL**

serwerklucza – **serwer klucza licencji**



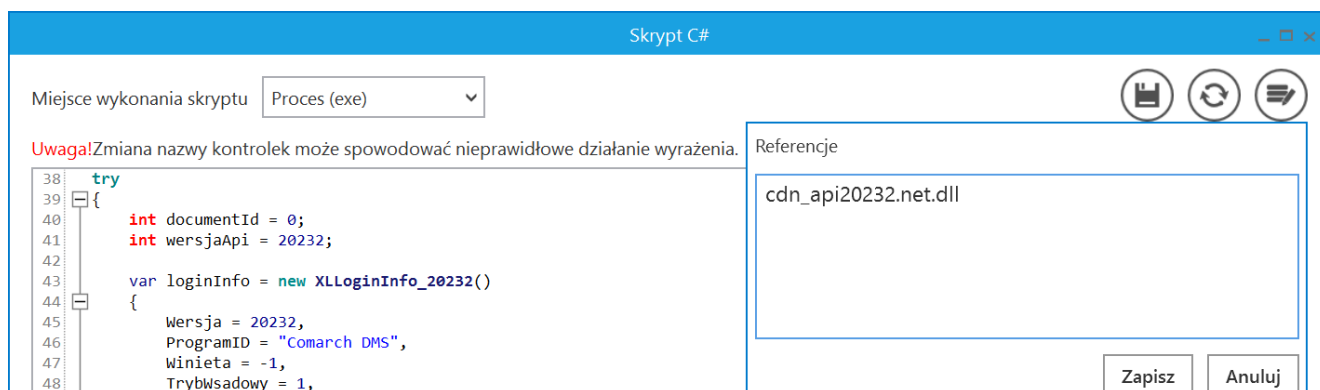
Następnie należy kliknąć w przycisk **[Referencje]**, znajdujący się w prawym górnym rogu okna „Skrypt C#”, w wpisać tam nazwę odpowiedniego pliku dll:

cdn_api[odpowiedni numer wersji].net.dll

Zapisz

Następnie należy nacisnąć przycisk

[Zapisz].



Fragment okna „Skrypt C#” z otwartym oknem „Referencje”

Po wprowadzeniu kodu do edytora i nazwy dll w oknie

„Referencje” należy **nacisnąć przycisk**  **[Kompiluj i zapisz]**, a następnie, gdy w dolnej części okna zostanie wyświetlony napis „Kompilacja skryptu zakończyła się sukcesem” nacisnąć przycisk **„Zamknij”**.

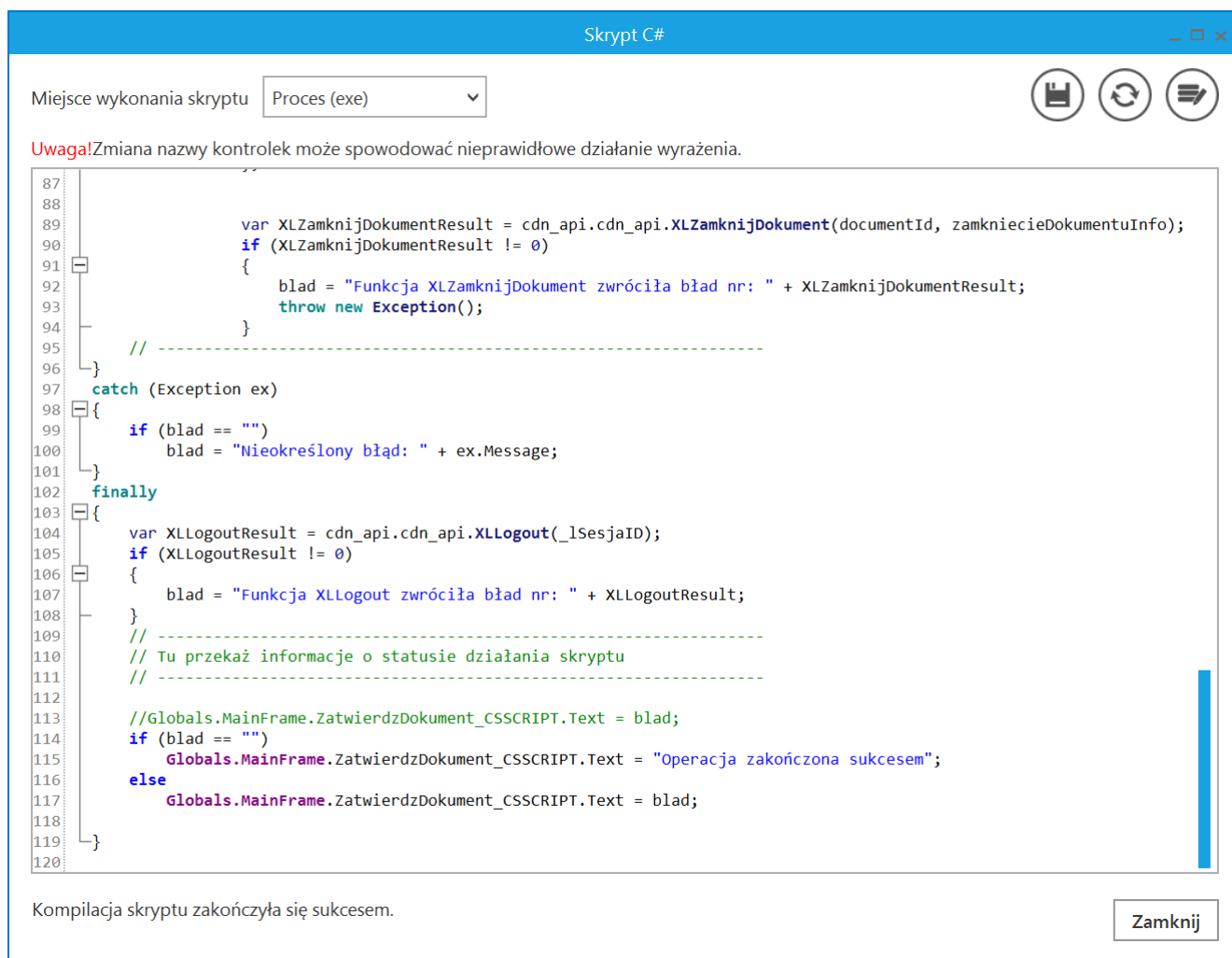
Uwaga

Jeśli w ramach okna „Referencje” po kliknięciu przycisku



[Referencje] wprowadzono niewłaściwą nazwę pliku dll,

wówczas po naciśnięciu przycisku  **[Kompiluj i zapisz]** w dolnej części okna wyświetlana jest informacja „Assembly not found”.



Widok okna „Skrypt C#” po skompilowaniu skryptu

C Kontrolka typu Komunikat – należy w ramach pola:

- „Nazwa (identyfikator)” wprowadzić nazwę „**Message1**”
- „Tytuł” wprowadzić tekst „**Status działania API ERP XL**”
- „Kontrolki powiązane” kliknąć w link „Dodaj” i wybrać kontrolkę „**Zatwierdź dokument**”
- „Wzór na wartość” wybrać wartość „**C# Script**”, kliknąć „Ustaw”, a następnie wprowadzić następujący kod:

```
Globals.MainFrame.Message1.Message  
= Globals.MainFrame.ZatwierdzDokument_CSSSCRIPT.Text
```

Po wprowadzeniu kodu należy **nacisnąć przycisk**  **[Kompiluj i zapisz]**, a następnie, gdy w dolnej części okna zostanie wyświetlony napis „Kompilacja skryptu zakończyła się sukcesem” nacisnąć przycisk **„Zamknij”**.

Komunikat

Nazwa (identyfikator)

Message1

Zmiana wartości (Obserwator)

Tytuł

Status działania API ERP XL

Kontrolki powiązane

Zatwierdź dokument (ZatwierdzDokument) 

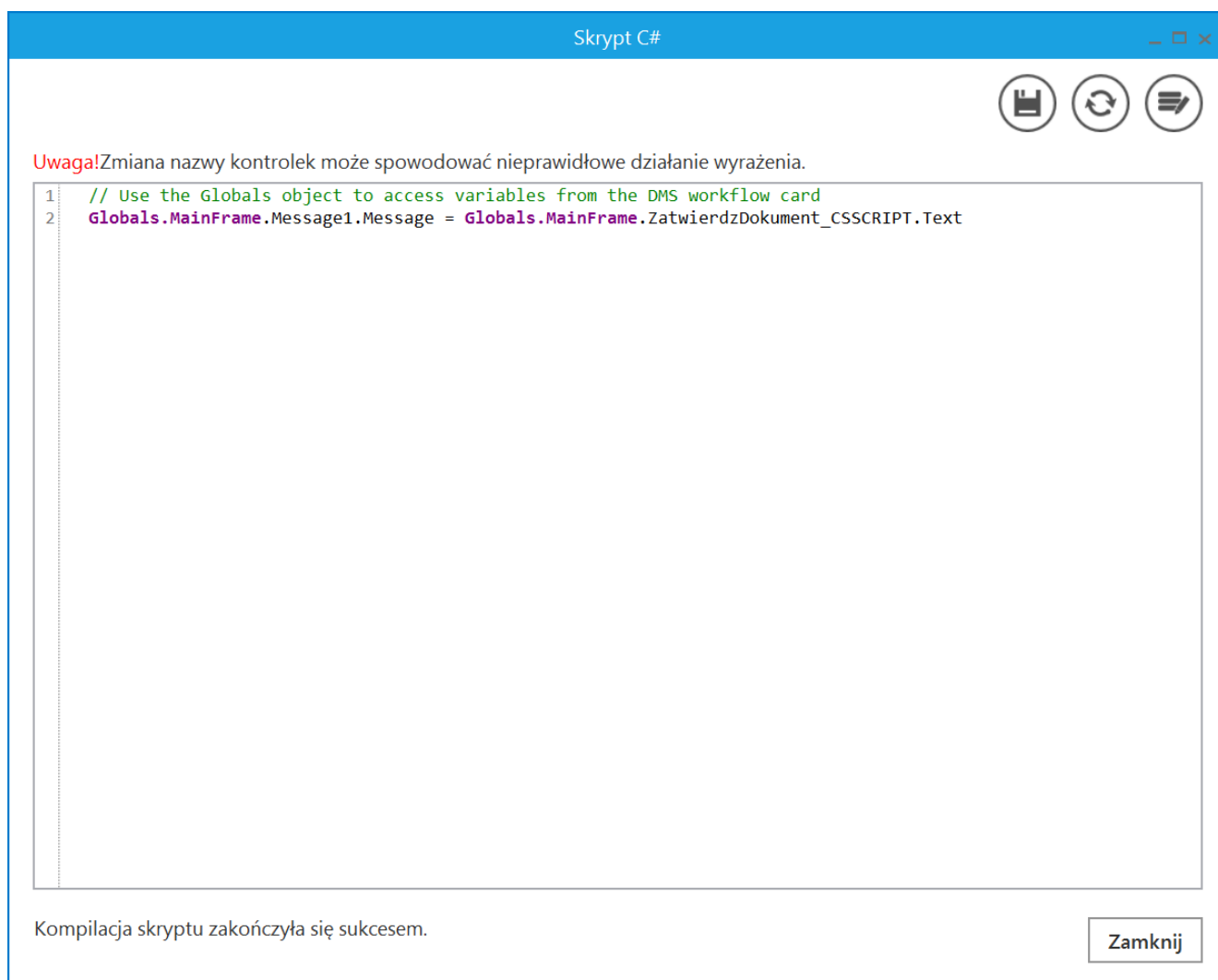
[Dodaj](#)

Wzór na wartość

C# Script 

[Ustaw](#)

Kontrolka „Message1”



Widok okna „Skrypt C#” po skompilowaniu skryptu

6. **zapisać typ obiegu** za pomocą przycisku



[Zapisz].

Zapisywanie typu obiegu

Zatwierdzanie dokumentu FZ w Comarch ERP XL – jak działa

Po zdefiniowaniu typu obiegu operator przechodzi na zakładkę



[Dokumenty], klika na typ obiegu „Zatwierdzanie faktury

zakupu” i naciska przycisk  [Dodaj]. Następnie kolejno:

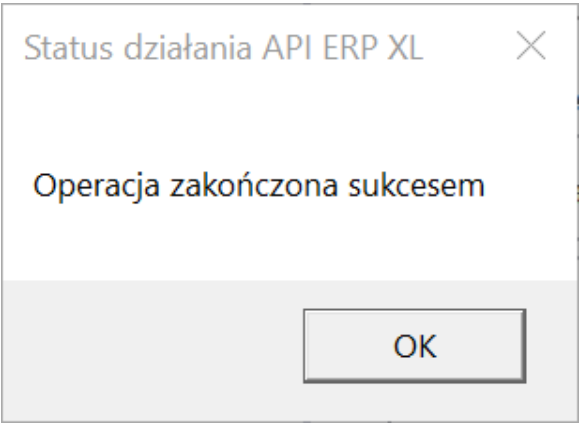
1. zapisuje nowy dokument, naciskając  [Zapisz]

2. dodaje dokument z Comarch ERP XL w ramach kontrolki „Faktura zakupu”

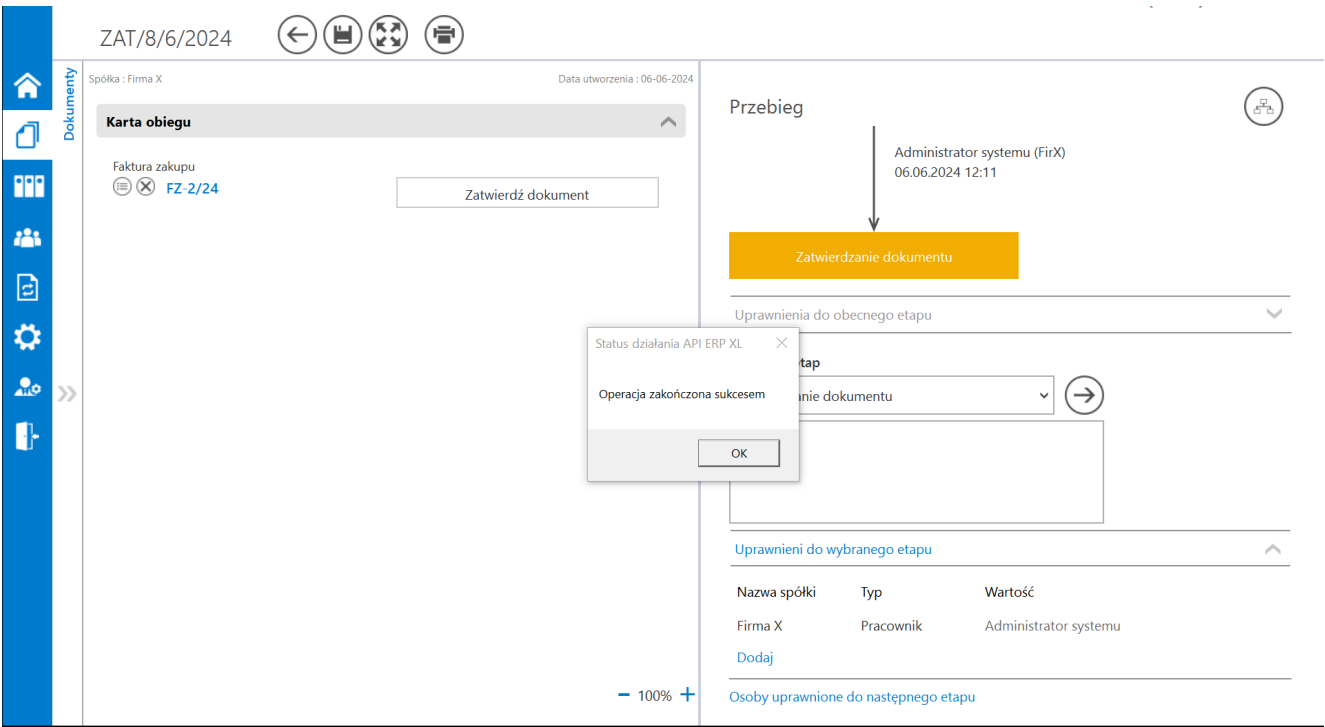
3. naciska przycisk „Zatwierdź dokument” – wówczas następuje zatwierdzanie dokumentu – poniżej przycisku widoczne są poruszające się punkty

4. Jeśli zatwierdzanie dokumentu zakończyło się powodzeniem,

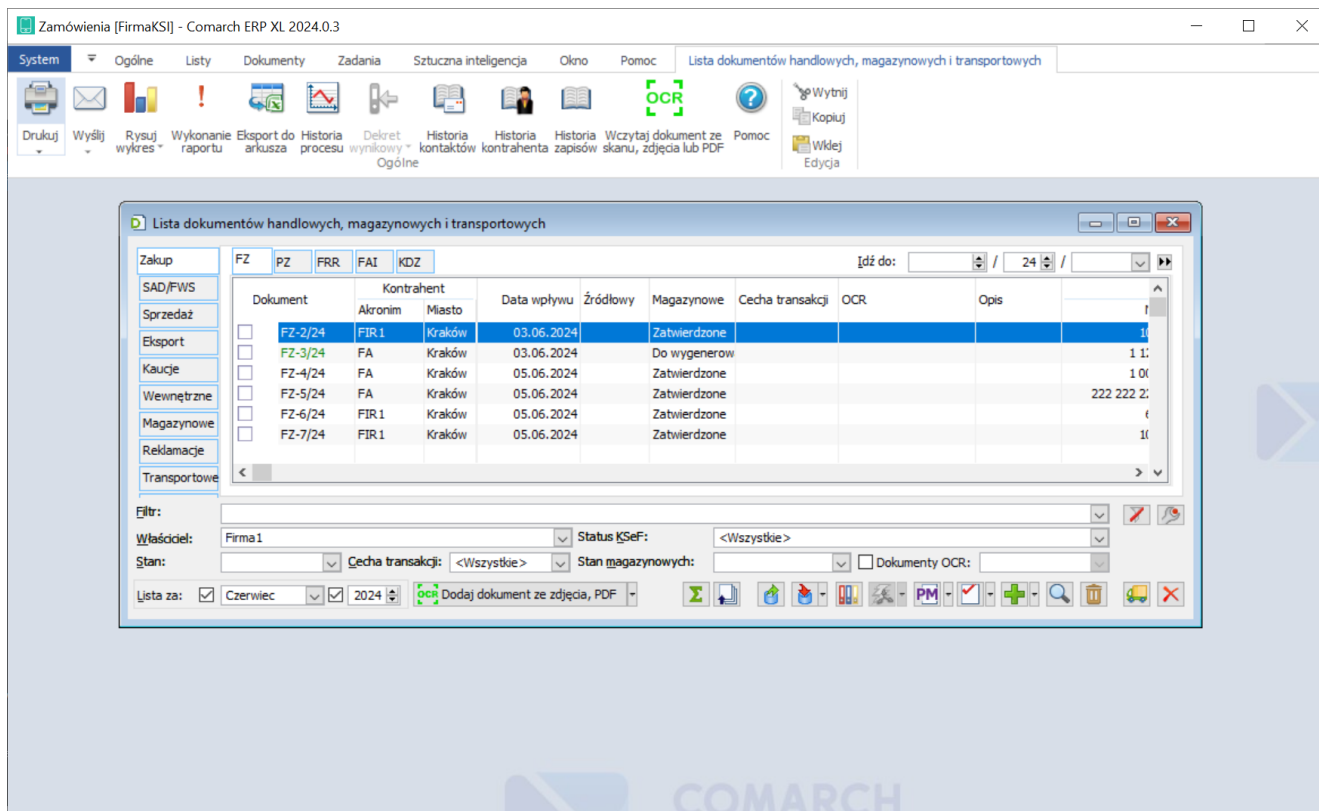
wówczas na środku ekranu wyświetlany jest następujący komunikat:



Komunikat wyświetlany, jeśli
zatwierdzanie dokumentu
zakończyło się powodzeniem



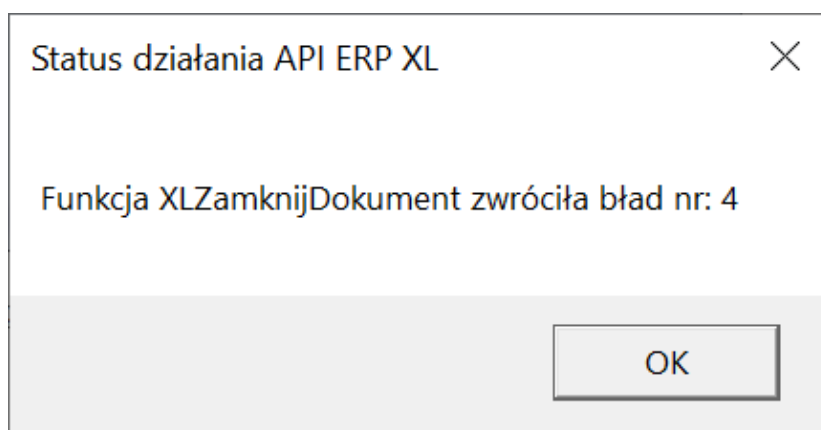
Widok szczegółów dokumentu z komunikatem o udanym
zatwierdzeniu dokumentu



Widok listy dokumentów w Comarch ERP XL z dokumentem FZ-2/24 zatwierdzonym z poziomu Comarch DMS

Uwaga

Jeśli operator w kontrolce „Faktura zakupu” wybierze dokument, który już został zatwierdzony w Comarch ERP XL, i kliknie przycisk „**Zatwierdź dokument**”, wówczas zostaje wyświetlony następujący komunikat:



Komunikat wyświetlany przy próbie
zatwierdzenia zatwierdzonego dokumentu

Rozpoczynasz pracę z Comarch DMS i chcesz dowiedzieć się, jak korzystać z programu? A może masz już podstawową wiedzę o Comarch DMS i chcesz dowiedzieć się więcej?

[Sprawdź Szkolenia Comarch DMS!](#)

[Powrót do początku artykułu](#)

Wskazanie identyfikatora connection stringa za pomocą kodu C#

Od wersji 2024.0.2 możliwe jest wskazanie identyfikatora connection stringa za pomocą kodu C#, aby określić connection string, na podstawie którego będzie następowało połączenie do bazy danych. Jeżeli:

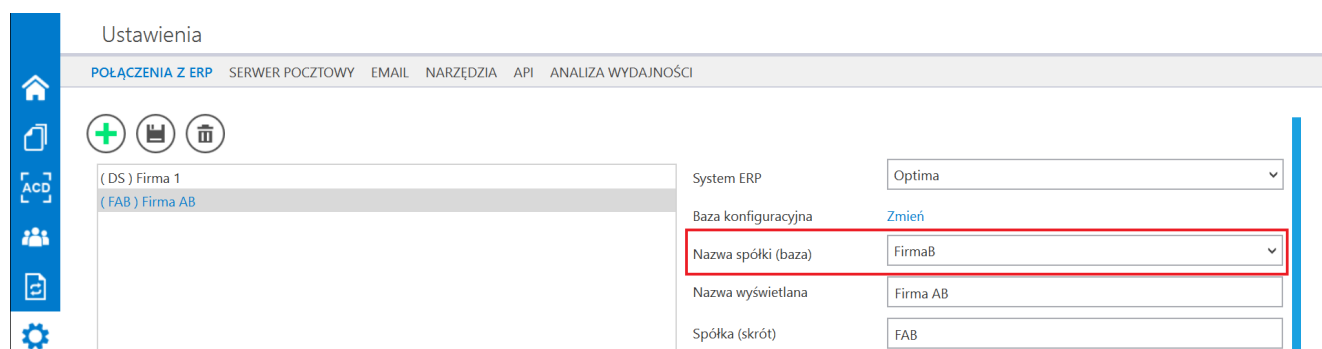
- `<add key="MultiCompany" value="false" />` – wówczas connection stringi strConnectERPComp i strConnectERPConf1 są przechowywane w pliku Web.config
- `<add key="MultiCompany" value="true" />` – wówczas connection stringi są przechowywane w pliku Web.config i w tabeli do.DF_ConfCMDictionary – w tym przypadku connection stringi do baz: konfiguracyjnej i firmowej są

zapisywane pod jednym ID połączenia, zastosowano zatem następujące identyfikatory:

- **nazwa_spółki_Comp** – pobiera connection string zapisany w kolumnie CMD_ConnectionStringERP / CMD_ConnectionStringERPEncrypted
- **nazwa_spółki_Conf** – pobiera connection string zapisany w kolumnie CMD_ConnectionStringERPConf / CMD_ConnectionStringERPConfEncrypted

gdzie nazwa_spółki – wartość wprowadzona w polu „Nazwa spółki” / „Nazwa spółki (baza)” na zakładce „Połączenia z ERP” na

zakładce  [Ustawienia]



Ustawienia

POŁĄCZENIA Z ERP SERWER POCZTOWY EMAIL NARZĘDZIA API ANALIZA WYDAJNOŚCI

(DS) Firma 1
(FAB) Firma AB

System ERP: Optima

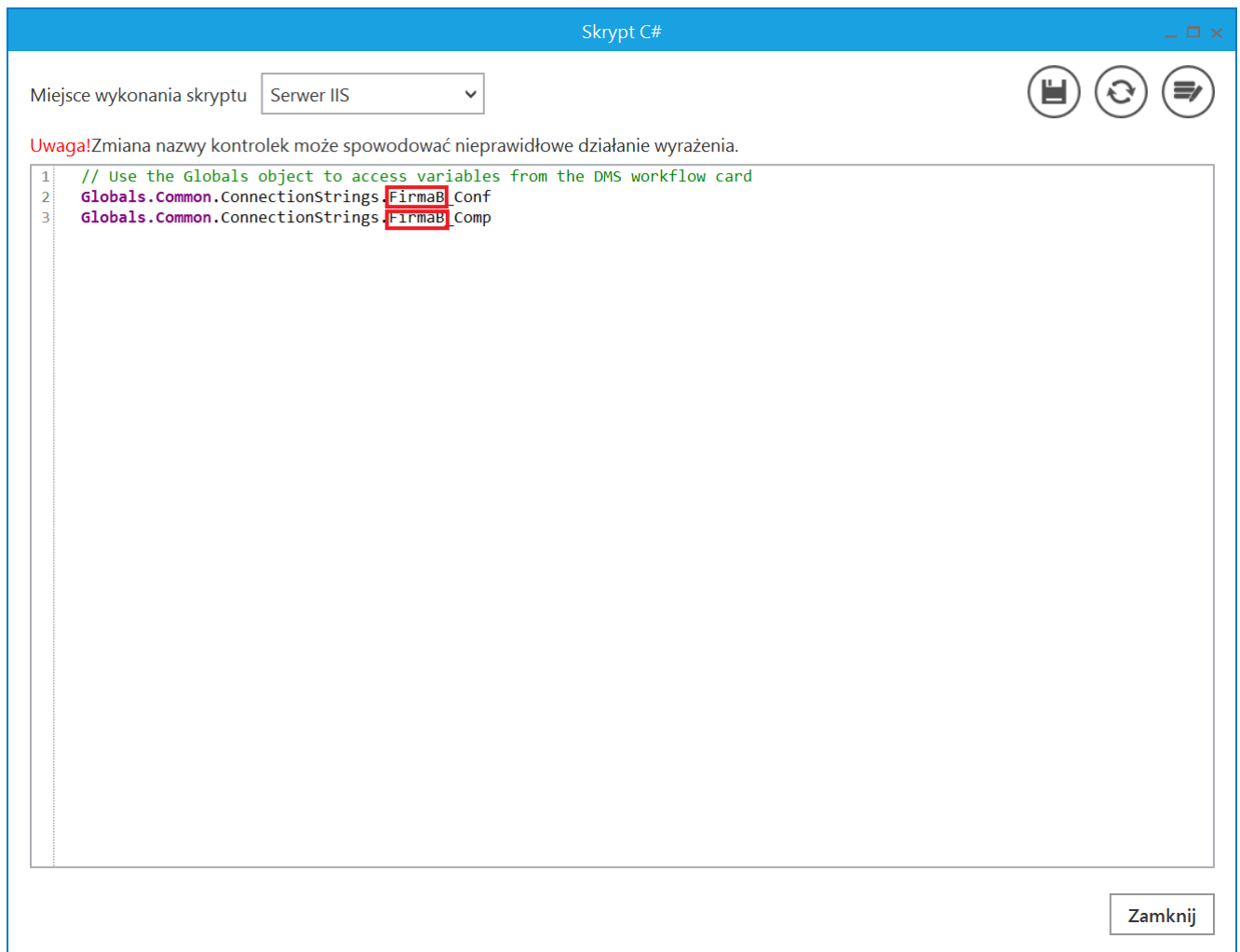
Baza konfiguracyjna: [Zmień](#)

Nazwa spółki (baza): FirmaB

Nazwa wyświetlana: Firma AB

Spółka (skrót): FAB

Wartość wprowadzona w polu „Nazwa spółki (baza)”, która wchodzi w skład identyfikatorów connection stringów



Przykład wykorzystania identyfikatorów connection stringów dla bazy Comarch ERP Optima w ramach kontrolki typu Własna akcja

Identyfikatory connection stringów są wykorzystywane w kodzie C#:

- dla MS SQL:
 - connection = new DmsSqlConnection(Globals.Common.ConnectionStrings.strConnectDMS)
 - DmsSqlCommand command = new DmsSqlCommand(connection, sqlQuery)
- dla PostgreSQL:
 - connection = new DmsNpgsqlConnection(„pgStrConnectDMS”)
 - DmsNpgsqlCommand command = new

```
DmsNpgsqlCommand(connection, sqlQuery)
```