



**Przykład zastosowania
kodu C# ze wskazaniem
identyfikatora connection string**

Spis treści

1	Opis przykładu	3
2	Dodawanie tabeli dbo.Powiązania w bazie Comarch BPM	3
3	Konfiguracja typu obiegu.....	4
4	Wyszukiwanie wartości w ramach kontrolki typu Lista zgodnie z wprowadzonym kodem C#	7

1 Opis przykładu

Poniższy przykład prezentuje zaawansowane użycie kodu C# w kontrolce Typu Lista, w celu dynamicznego wyszukiwania danych z bazy SQL i prezentowania ich bezpośrednio na karcie obiegu.

Główną zaletą tego rozwiązania jest:

- elastyczność modelowania kontrolek za pomocą kodu C#
- bezpieczne połączenie z bazą SQL, eliminując konieczność jawnego umieszczania loginu i hasła w connection stringu. Kod odwołuje się do bezpiecznie zarządzanego Identyfikatora Connection String, gwarantując ochronę poufnych poświadczeń.

2 Dodawanie tabeli dbo.Powiazania w bazie Comarch BPM

Dla zastosowania poniższego przykładu konieczne jest dodanie w bazie nowej tabeli dbo.Powiazania, w której znajdują się dwie kolumny:

- NIP
- Punkt

W tym celu w MS SQL Server Management Studio należy:

- na bazie (Comarch BPM, Comarch ERP Optima lub Comarch ERP XL, w zależności od tego, z którą bazą będzie następowało połączenie) następujące zapytanie:

```
CREATE TABLE dbo.Powiazania (  
    NIP NVARCHAR(20) NOT NULL,  
    Punkt NVARCHAR(100) NOT NULL  
);
```

- wprowadzić do tabeli wartości – przykładowo wykonując na bazie następujące zapytanie:

```
INSERT INTO dbo.Powiazania (NIP, Punkt)  
VALUES  
( '6770065406', 'Comarch S.A.' ),  
( '6751387586', 'Comarch Polska' ),  
( '2112223434', 'Firma ABC' );
```

- nadać operatorom Comarch BPM uprawnienie do wykonywania polecenia **SELECT** na utworzonej tabeli – wykonując na bazie następujące zapytanie (jeśli połączenie z bazą Comarch BPM następuje za pośrednictwem operatora ODUser):

```
GRANT SELECT ON dbo.Powiazania TO ODUser;
```

Results		Messages
	NIP	Punkt
1	6770065406	Comarch S.A.
2	6751387586	Comarch Polska
3	2112223434	Firma ABC

Rys 1. Przykładowa tabela dbo.Powiazania

3 Konfiguracja typu obiegu

W ramach typu obiegu należy dodać kontrolkę typu Lista o nazwie „Lista” (identyfikator: List1). W ramach kontrolki operator powinien wybrać następujące wartości:

- „C# Script” w polu „Inicjowanie”
- „Text” w polu „Rodzaj identyfikatora”
- „Wyszukiwanie” w polu „Tryb pracy”

Lista

Nazwa wyświetlana

Lista

Nazwa (identyfikator)

List1

Prezentuj na całej szerokości

☐

Ustaw kontrolkę na początku wiersza

☐

Pokaż na liście

☐

Inicjowanie wartości i zmiana wartości (obserwator)

Inicjowanie

C# Script

Rodzaj identyfikatora

Text

Ustaw

Inicjuj przy każdym otwarciu dokumentu

☐

Kontrolki powiązane

Dodaj

Tryb pracy

Wyszukiwanie

☐ Szukaj po dowolnym ciągu znaków

Rys 2. Przykładowa kontrolka „Lista”

Następnie należy kliknąć w przycisk **Ustaw** [Ustaw] znajdujący się pod polem „Rodzaj identyfikatora” i wprowadzić następujący kod:

```
using System.Data.SqlClient;
```

```
DmsSqlConnection conn = null;
String Point;
String cs;

try
{
    // Ponizej należy wstawić odpowiedni identyfikator connection stringa

    conn = new DmsSqlConnection(Globals.Common.ConnectionStrings.strConnectDMS);
    conn.Open();

    // Zapytanie wyszukuje wpisane przez operatora dane w kolumnie "Punkt"

    cs = "SELECT [NIP] ,[Punkt] FROM [dbo].[Powiazania] where Punkt like '" +
        Globals.MainFrame.List1.EnteredText + "%'";

    //throw new Exception (cs);

    DmsSqlCommand command = new DmsSqlCommand(conn, cs);

    Globals.MainFrame.List1.Clear();

    using (var reader = command.ExecuteReader())
    {
        while (reader.Read())

            // Dodaje pozycję: NIP jako wartość, Punkt jako tekst wyświetlany

            Globals.MainFrame.List1.AddItem(reader["NIP"].ToString(),
            reader["Punkt"].ToString());
    }
}

finally
{
    conn.Close(); // Zamyka połączenie
}
```

DmsSqlConnection to obiekt biblioteki będący rozszerzeniem standardowej klasy Microsoftu z ADO.NET o nazwie SqlConnection – biblioteka jest używana do komunikacji z Microsoft SQL Server, PostgreSQL i ODBC – powyższy przykład dotyczy komunikacji z Microsoft SQL Server.

Do klasy DmsSqlConnection zostaje przekazana nazwa połączenia bazodanowego (nazwa bazy danych z pliku Web.config – dla trybu jednospółkowego lub z zakładki „Ustawienia” – dla trybu wielospółkowego – por. poniżej.

Dodatkowo mogą zostać doczytane inne połączenia zdefiniowane w Web.config). Po wprowadzeniu parametru Globals.Common.ConnectionStrings, po kropce jest wyświetlana lista wszystkich połączeń dostępnych w danej instancji Comarch BPM.

Model programowania w klasie DmsSqlConnection jest wzorowany na modelu programowania ADO.NET.

W powyższym przykładzie operator tworzy obiekt połączenia bazodanowego za pomocą klasy DmsSqlConnection (conn = new DmsSqlConnection(Globals.Common.ConnectionStrings.strConnectDMS)).

Operator wskazuje identyfikator connection stringa za pomocą kodu C#, określając w ten sposób, na podstawie którego connection stringa ma nastąpić połączenie do bazy. Jeżeli w ramach pliku Web.config w folderze z plikami aplikacji serwerowej:

- `<add key="MultiCompany" value="false" />` – wtedy connection stringi strConnectERPComp i strConnectERPConf1 są przechowywane w pliku Web.config – w takim przypadku, podobnie jak w powyższym przykładzie, operator wprowadza parametr Globals.Common.ConnectionStrings.strConnectDMS, aby połączyć się z bazą
- `<add key="MultiCompany" value="true" />` – w takim przypadku connection stringi są przechowywane w pliku Web.config i w tabeli do.DF_ConfCMDictionary – wówczas 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 to wartość wprowadzona w polu „Nazwa spółki” / „Nazwa spółki (baza)” na zakładce „Połączenia

z ERP” na zakładce  **[Ustawienia]**.

Przykładowo: jeśli w trybie wielofirmowym na zakładce „Ustawienia”:

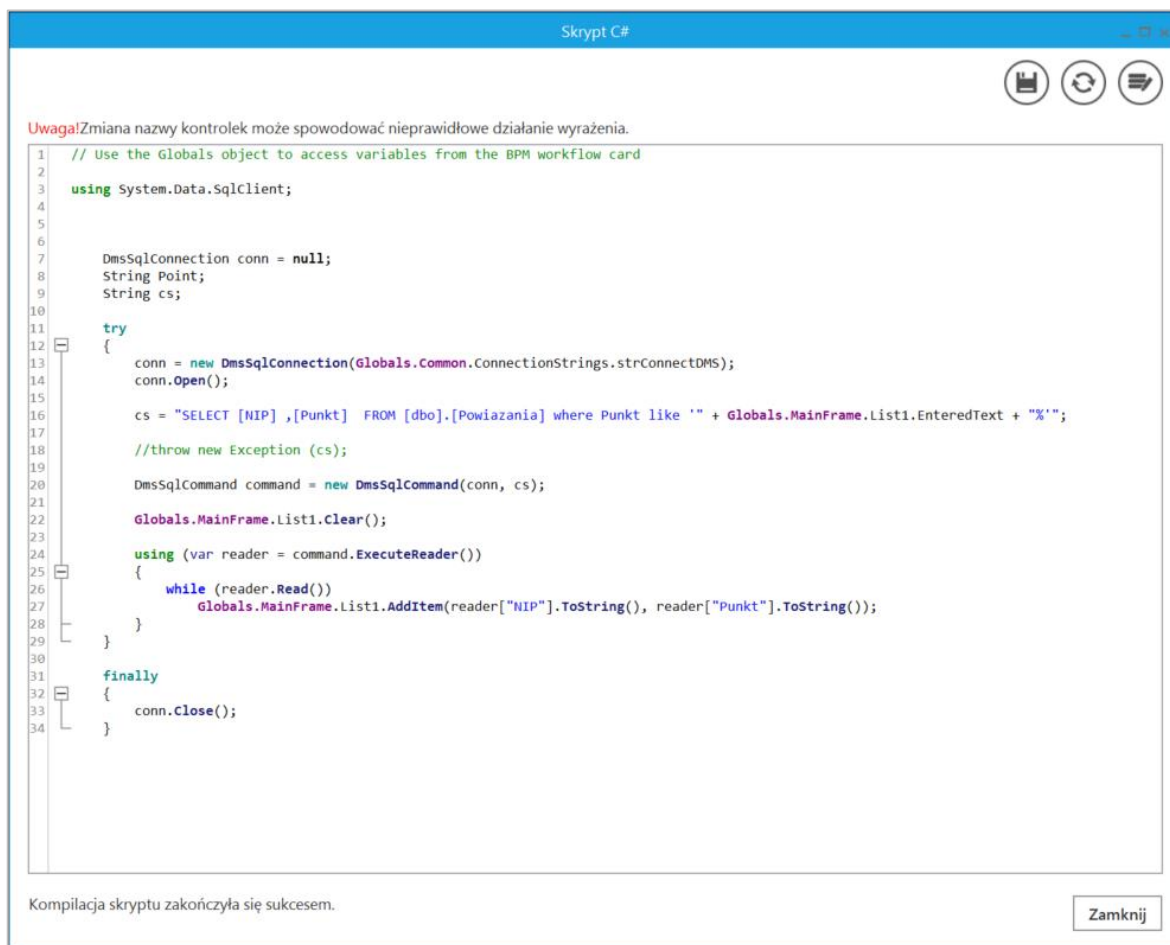
- dla danej spółki XL w ramach pola „Nazwa spółki” wprowadzono nazwę „FirmaX”, wówczas w powyższym przykładzie operator wprowadza Globals.Common.ConnectionStrings.FirmaX_Comp
- dla danej spółki Optima w ramach pola „Nazwa spółki (baza)” wprowadzono nazwę „FirmaB”, wówczas w powyższym przykładzie operator wprowadza Globals.Common.ConnectionStrings.FirmaB_Comp

Następnie operator otwiera połączenie bazodanowe (conn.Open()) i wprowadza zapytanie (w tym przypadku cs = "SELECT [NIP] ,[Punkt] FROM [dbo].[Powiazania] where Punkt like '" + Globals.MainFrame.List1.EnteredText + "%'").

Następnie operator tworzy obiekt komendy, który wymaga obiektu połączenia bazodanowego oraz treści tekstowej komendy (DmsSqlCommand command = new DmsSqlCommand(conn, cs)). Później czyści kontrolkę typu Lista (Globals.MainFrame.List1.Clear()).

Następnie operator tworzy obiekt readera, który wykonuje odczyt na komendzie zgodnie z wprowadzonymi warunkami – na pierwszej (NIP) i drugiej (Punkt) kolumnie tabeli - każdy odczytany wiersz zostaje dodany do listy w kontrolce typu Lista. Zgodnie z zapytaniem w ramach opcji dostępnych na liście w ramach kontrolki typu Lista zostaną wyświetlone takie pozycje z tabeli dbo.Powiazania, które w kolumnie „Punkt” rozpoczynają się tak, jak brzmi fraza wprowadzona przez operatora.

Później operator zamyka obiekt połączenia bazodanowego (conn.Close()).



Rys 3. Wprowadzanie kodu C# w kontrolce typu Lista



Po wprowadzeniu kodu C# należy kliknąć w ikonę **[Kompiluj i zapisz]** – jeśli wprowadzone dane są poprawne, w dolnej części okna zostanie wyświetlony komunikat „Kompilacja skryptu zakończyła się sukcesem”. Następnie operator powinien zapisać zmiany w typie obiegu.

4 Wyszukiwanie wartości w ramach kontrolki typu Lista zgodnie z wprowadzonym kodem C#

Na nowododanym dokumencie w ramach typu obiegu, który skonfigurowano w poprzednim punkcie, operator może w ramach kontrolki typu Lista wprowadzić początek wyszukiwanej frazy – wówczas na liście wyników wyszukiwania zostaną wyświetlone wszystkie pozycje z tabeli dbo.Powiazania, dla których wartość w kolumnie „Punkt” zaczyna się od danej frazy.

Comarch BPM 2025.2.1 Administrator2

PROJ/10/9/2025

Data utworzenia : 25-09-2025

Karta obiegu

Lista:

Comarch S.A.
Comarch Polska

Numer dokumentu: Data zakupu:

Data otrzymania: Termin płatności: Tytuł maila:

Kontrahent: NIP: Wartość netto:

Stawka VAT (w %): Kwota VAT: Wartość brutto:

Do wiadomości: Treść maila: Poczta kontrahenta:

Państwo kontrahenta: Czy kontrahent jest aktywnym płatnikiem VAT?: za2:

Akcja:

Przebieg

Administrator2
25.09.2025 15:31

Etap 1

Uprawnienia do obecnego etapu

Następny etap

Etap 2

Uprawnieni do wybranego etapu

Osoby uprawnione do następnego etapu

- 80 % +

Rys 4. Wyszukiwanie wyników zgodnie z wprowadzoną frazą w kontrolce typu Lista

COMARCH ERP

Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiegokolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie na nośniku filmowym, magnetycznym lub innym, powoduje naruszenie praw autorskich niniejszej publikacji.

Copyright © 2025 COMARCH
Wszelkie prawa zastrzeżone.