

Czy wiesz, że kontrolkę Dane tabelaryczne można zainicjować wartością domyślną?

Zastosowanie

Dzięki tej konfiguracji, proces obsługi dokumentów w Comarch BPM zyskuje bezpośrednią oszczędność czasu i gwarancję spójności danych. Umożliwia automatyczne wstępne wypełnienie kontrolki tabelarycznej na wskazanym etapie obiegu, eliminując potrzebę ręcznego dodawania wierszy. W efekcie, użytkownik zyskuje natychmiastową gotowość do pracy w momencie rozpoczęcia obiegu. Minimalizuje to błędy poprawności danych, gwarantując, że dane są dokładne, spójne i aktualne oraz znacząco przyspiesza obsługę standardowych, powtarzalnych procesów.

Zawartość przykładu

- **Definicja Typu Obiegu (Workflow):** Wzorcowa ścieżka obiegu, która zawiera konfigurację kontrolki Dane tabelaryczne

[Kliknij tutaj, aby pobrać przykład](#)

Zasada działania

Mechanizm wykorzystuje zasadę, że **kontrolka tabelaryczna to tabela tymczasowa SQL**, którą można wypełnić dowolnymi danymi za pomocą zapytania.

- **Zapytanie SQL** jest umieszczone na zakładce **Inicjowanie wartości** i zawiera dane, które mają pojawić się w tabeli.
- Na etapie, na którym kontrolka ma zostać zainicjowana, zaznacza się opcję „**Inicjalizacja SQL**”. Zapytanie jest uruchamiane raz, wypełniając wiersze wartością domyślną.
- Aby zainicjować kontrolkę więcej niż jednym wierszem, wykorzystuje się instrukcję operacji na zbiorach **UNION**, która łączy wyniki poszczególnych zapytań **SELECT**.
- Ostatnią kolumną w zapytaniu jest kolumna **LP (numer wiersza)**, którą w zapytaniu inicjalizujemy zerami (0 as LP). System automatycznie zastępuje te zera kolejnymi wartościami porządkowymi.

Konfiguracja przykładu

1. **Utworzono kontrolkę Tabelaryczną:** Dodano kontrolkę typu **Dane tabelaryczne** o nazwie (identyfikatorze) **Dokumenty** i skonfigurowano w niej następujące kolumny:

- **"Nazwa"** (identyfikator kolumny: **Nazwa**) – typ **Tekst**
- **"Kwota"** (identyfikator kolumny: **Kwota**) – typ **Liczba stałoprzecinkowa**
- **"Termin"** (identyfikator kolumny: **Termin**) – typ **Data i godzina**

Karta obiegu

Dokumenty

LP

Nazwa

Kwota

Termin



Kolumny w kontrolce Dane tabelaryczne

2. **Dodano zapytanie inicjujące SQL:** W kontrolce **Dane tabelaryczne**, na zakładce **Inicjowanie wartości** w sekcji **Inicjowanie SQL** dodano zapytanie, które uzupełni kontrolkę wartościami domyślnymi: **numerem faktury, kwotą oraz terminem**.

SELECT

'Faktura 001' AS nazwa, 1500.00 AS kwota, '2025-04-15' AS termin, 0 as LP

UNION

SELECT

'Faktura 002' AS nazwa, 3200.50 AS kwota, '2025-04-20' AS termin, 0 as LP

Zapytanie SQL

Uwaga! Wprowadzenie niepoprawnych danych może powodować błędy aplikacji.

Treść zapytania:

```

1 SELECT
2 'Faktura 001' AS nazwa, 1500.00 AS kwota, '2025-04-15' AS termin, 0 as LP
3 UNION
4 SELECT
5 'Faktura 002' AS nazwa, 3200.50 AS kwota, '2025-04-20' AS termin, 0 as LP

```

Kontrolki

^UserId
^DocumentId
^ParentDocumentId
^WorkflowId
^StageId

Składnia:

SELECT
varchar(900) - Nazwa
decimal(22, 4) - Kwota
'YYYYMMDD' - Termin
int - tryb: 1 - nadpisanie
wierszy, 2 - dodanie wierszyPOS -
pozycja (opcjonalny numer wiersza)

Zapisz

Zamknij

[Dane tabelaryczne] Dokumenty

Dane tabelaryczne

OGÓLNE
LISTY
INICJOWANIE WARTOŚCI
OGRANICZENIE EDYCJI

Inicjowanie wartości

Inicjowanie

SQL OD

Pokaż Usuń

Kontrolki powiązane

Dodaj

Wzór na wartość

SQL OD

Ustaw

Zapytanie SQL w kontrolce Dane tabelaryczne na zakładce Inicjowanie wartości

3. Włączenie Inicjalizacji: Na etapie, na którym kontrolka ma się zainicjować wartościami domyślnymi (np. na pierwszym etapie obiegu), zaznaczono opcję „Inicjalizacja SQL”.

Wprowadzenie

Weryfikacja
Archiwizacja

Etap

OGÓLNE
UPRAWNIENIA
PRZEKAZANIE
MAIL
KONTROLKI
PARAMETRY

☒ Włącz funkcję obserwatora podczas otwierania dokumentu

Kontrolka	☒ Zaznacz wszystko Widoczna	☐ Zaznacz wszystko Tylko do odczytu	☐ Zaznacz wszystko Wymagana	☒ Zaznacz wszystko Inicjalizacja Sql/C#	☐ Zaznacz wszystko Odświeżanie tabeli na żądanie	Kolejność inicjowania
Zakładka 1	☒	☐	☐	☒	☐	
Karta obiegu	☒	☐	☐	☒	☐	
Dokumenty	☒	☐	☐	☒	☐	

Włączenie Inicjalizacji SQL

Uzupełniona Kontrolka typu Dane Tabelaryczne

Po utworzeniu dokumentu, kontrolka typu **Dane Tabelaryczne** została **automatycznie zainicjowana** domyślnymi wartościami pobranymi z zapytania SQL. **Tabela zawiera wiersze z kompletnymi danymi: numer faktury, kwotą, terminem**, eliminując tym samym konieczność **ręcznego** dodawania pozycji i **skracając czas** obsługi dokumentu.

Karta obiegu

Dokumenty

LP	Nazwa	Kwota	Termin
1	Faktura 001	1 500,00	15.04.2025
2	Faktura 002	3 200,50	20.04.2025

+

✖

Kontrolka typu Dane tabelaryczne z uzupełnionymi kolumnami

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

[Sprawdź Szkolenia Comarch BPM!](#)

Czy wiesz, że po dodaniu wiersza w kontrolce typu Dane tabelaryczne poszczególne kolumny mogą zainicjować się wartością domyślną?

Zastosowanie

Dzięki tej konfiguracji, proces **wprowadzania danych** jest znacząco **przyspieszony** w przypadku dokumentów wymagających wielokrotnego dodawania tych samych lub podobnych pozycji. Głównym celem jest **automatyczne** wypełnienie kolumn **nowo dodanego wiersza** domyślnymi wartościami (np. nazwa, ilość, stawka VAT). **Eliminuje** to błędy i powtarzalne czynności użytkownika, co pozwala pracownikom skupić się na **weryfikacji danych**, a nie na ich rutynowym wprowadzaniu. Dodatkową, unikalną korzyścią tego przykładu jest to, że zawiera on w sobie pełną logikę **dynamicznego i dwukierunkowego przeliczania kwot (Netto/Brutto)** w wierszu, dzięki dodatkowemu zapytaniu SQL.

Zawartość przykładu

- **Definicja Typu Obiegu (Workflow):** Wzorcową ścieżkę obiegu dokumentu kosztowego z krokami akceptacji i konfiguracją kontrolki Dane tabelaryczne i

[Kliknij tutaj, aby pobrać przykład](#)

Zasada działania

Mechanizm opiera się na monitorowaniu **zdarzenia dodania nowego wiersza** i uruchamianiu zapytania SQL, które w odpowiedzi wypełnia pola domyślnymi wartościami.

- Kontrolka **Dane Tabelaryczne** jest skonfigurowana tak, aby **obserwować samą siebie** i reagować na zdarzenie wywołane przez dodanie **nowego wiersza** za pomocą przycisku „+” (**Dodaj**).
- Na zakładce **Inicjowanie wartości** umieszczone jest zapytanie **SQL**, które sprawdza, czy kolumna ma **wartość -1** (@Elementy_Column@ = -1). **Wartość -1** oznacza, że operacja dotyczy **dodania nowego wiersza**.
- Wewnątrz bloku BEGIN zapytanie SQL używa komendy **UPDATE** @Elementy@ do ustawienia domyślnych wartości (np. Nazwa = 'a-vista', Ilosc = 1, StawkaVAT=1) dla **nowo dodanej pozycji**.
- Kolumna wynikowa **Netto** została ustawiona w trybie „Do odczytu”, aby użytkownik nie mógł ręcznie jej edytować.

Konfiguracja przykładu

1. **Utworzono kontrolkę Tabelaryczną:** Dodano kontrolkę typu **Dane tabelaryczne** o nazwie (identyfikatorze) **Elementy** i skonfigurowano w niej następujące kolumny:

- **"Nazwa"** (identyfikator kolumny: **Nazwa**) – typ **Tekst**
- **"Cena"** (identyfikator kolumny: **Cena**) – typ **Liczba stałoprzecinkowa**

- **"Ilość"** (identyfikator kolumny: **Ilosc**) – typ **Liczba stałoprzecinkowa**
- **"Netto"** (identyfikator kolumny: **Netto**) – typ **Liczba stałoprzecinkowa**
- **"Stawka VAT"** (identyfikator kolumny: **StawkaVAT**) – typ **Lista**
- **"Brutto"** (identyfikator kolumny: **Brutto**) – typ **Liczba stałoprzecinkowa**

Pozycje dokumentu

Elementy

LP	Nazwa	Cena	Ilość	Netto	Stawka VAT	Brutto
						

Kolumny w kontrolce Dane tabelaryczne

2. W kontrolce typu **Dane tabelaryczne** na zakładce **Listy** zdefiniowano zapytanie pobierające:

- **Stawki VAT:**

```
select Naz_gidLp as Id,
CASE
  WHEN Naz_Nazwa='A 23.00' THEN '23%'
  WHEN Naz_Nazwa='B 8.00' THEN '8%'
  WHEN Naz_Nazwa='C 0.00' THEN '0%'
  WHEN Naz_Nazwa='D 0.00' THEN 'ZW'
  WHEN Naz_Nazwa='E 0.00' THEN 'NP'
  WHEN Naz_Nazwa='F 7.00' THEN '7%'
  WHEN Naz_Nazwa='G 5.00' THEN '5%'
END
```

```
from CDN.Nazwy
where Naz_GIDTyp = 624 AND trim(substring(Naz_Nazwa,9,2))= ''
```

[Dane tabelaryczne] Elementy

Dane tabelaryczne

OGÓLNE

LISTY

INICJOWANIE WARTOŚCI

OGRANICZENIE EDYCJI

Lista 1

Nazwa

Stawka VAT

Inicjowanie

SQL ERP

Rodzaj identyfikatora

Int

Pokaż

Usuń

Tryb pracy

Lista

Lista Stawka VAT w kontrolce Dane tabelaryczne

3. **Dodano obserwację kontrolki:** W kontrolce **Dane tabelaryczne**, na zakładce **Inicjowanie wartości**, dodano powiązanie do samej siebie.

[Dane tabelaryczne] Elementy

Dane tabelaryczne

OGÓLNELISTY**INICJOWANIE WARTOŚCI**OGRANICZENIE EDYCJI

Inicjowanie wartości

InicjowanieBrak

Kontrolki powiązaneElementy (Elementy)

Dodaj

Wzór na wartośćSQL ERP

Ustaw

Powiązanie kontrolki do samej siebie

4. **Dodano zapytanie inicjujące SQL:** W kontrolce typu **Dane tabelaryczne**, na zakładce **Inicjowanie wartości** w sekcji **Inicjowanie SQL ERP** dodano zapytanie, które zawiera warunek sprawdzający dodanie nowego wiersza **IF @Elementy_Column@ = -1** oraz instrukcję **update**, która ustawia wartości domyślne dla kolumn (**Nazwa**, **Ilość**, **Stawka VAT**) w bieżącym wierszu.

Oprócz inicjacji nowego wiersza, zapytanie zawiera również logikę **dwukierunkowego przeliczania wartości** (Netto, Brutto, Cena) dla już istniejących wierszy, co czyni ten wzorzec gotowym rozwiązaniem dla każdej kontrolki z kalkulacjami finansowymi.

```
IF @^SenderControlName@ = 'Elementy'
```

```
IF @Elementy_Column@ = -1 /*[1] LOGIKA: INICJACJA NOWEGO  
WIERSCZA (WYZWALANE KLIKNIĘCIEM '+') */
```

```
BEGIN
```

```
/*Ustawienie domyślnych wartości dla nowego wiersza */
```

```
UPDATE @Elementy@
```

```
SET [Nazwa] = 'a-vista',
```

```
[Ilosc] = 1,
```

```
[StawkaVAT] = 1 /*Identyfikator stawki domyślnej (ID=1 -  
stawka 23%)*/*
```

```
WHERE POS = @Elementy_Row@
```

```
/*!!! OBOWIĄZKOWE: Odświeżenie wiersza w kontrolce Dane  
Tabelaryczne !!! */
```

```
SELECT * FROM @Elementy@
```

```
END
```

```
ELSE IF @Elementy_Column@=2 OR @Elementy_Column@=3 OR  
@Elementy_Column@=5
```

```
/*[2] LOGIKA: PRZELICZANIE W PRZÓD (Cena/Ilość/Stawka VAT ->  
Netto/Brutto) */
```

```
BEGIN
```

```
DECLARE @VAT AS DECIMAL(4, 2)
```

```
/*Pobranie stawki VAT z tabel CDN.Nazwy */
```

```
SELECT @VAT = CAST(SUBSTRING(Naz_Nazwa1, 0, CHARINDEX(' ',  
Naz_Nazwa1))
```

```
AS DECIMAL(4, 2)) / 100
```

```
FROM CDN.Nazwy
```

```
WHERE Naz_GIDTyp = 624 AND Naz_GIDLp = (SELECT [StawkaVAT]  
FROM @Elementy@
```

```
WHERE POS = @Elementy_Row@)
```

```
/*Wyliczenie i aktualizacja netto i brutto*/
```

```
UPDATE @Elementy@
```

```
SET [Netto] = (Ilosc * Cena),
```

```
[Brutto] = (Ilosc * Cena) * (@VAT + 1)
```

```
WHERE POS = @Elementy_Row@
```

```
/*!!! OBOWIĄZKOWE: Odświeżenie wiersza w kontrolce Dane  
Tabelaryczne !!! */
```

```
SELECT * FROM @Elementy@
```

```
END
```

```
ELSE BEGIN
```

```
IF @Elementy_Column@=3 OR @Elementy_Column@=5 OR  
@Elementy_Column@=6 -- [3] LOGIKA: PRZELICZANIE W TYŁ (Brutto
```

-> Cena/Netto)

/*Pobranie stawki VAT*/

SELECT @VAT = CAST(SUBSTRING(Naz_Nazwa1, 0, CHARINDEX(' ',
Naz_Nazwa1))

AS DECIMAL(4, 2)) / 100

FROM CDN.Nazwy

WHERE Naz_GIDTyp = 624 AND Naz_GIDLp = (SELECT [StawkaVAT]
FROM @Elementy@

WHERE POS = @Elementy_Row@)

/*Aktualizacja ceny i netto*/

UPDATE @Elementy@

SET [Cena] = [Brutto] / (@VAT + 1),

[Netto] = [Ilosc] * ([Brutto] / (@VAT + 1))

WHERE POS = @Elementy_Row@

/*!!! OBOWIĄZKOWE: Odświeżenie wiersza w kontrolce Dane
Tabelaryczne !!!*/

SELECT * FROM @Elementy@

END

Wyrażenie kontrolki

Uwaga! Zmiana nazwy kontrolki może spowodować nieprawidłowe działanie wyrażenia.

```
1 IF @SenderControlName = 'Elementy'
2 IF @Element_Column = -1 /*[1] LOGIKA: INICJACJA NOWEGO Wiersza (WYZNALANE KLIKNIĘCIEM '+') */
3 BEGIN
4 /*Ustawienie domyślnych wartości dla nowego wiersza */
5 UPDATE @Elementy@
6 SET [Nazwa] = 'a-vista',
7 [Ilość] = 1,
8 [StawkaVAT] = 1 /*Identyfikator stawki domyślnej (ID=1 - stawka 23%)/
9 WHERE POS = @Element_Row@
10
11 /*!!! OBOWIĄZKOWE: Odświeżenie wiersza w kontrolce Dane Tabełaryczne !!! */
12 SELECT * FROM @Elementy@
13 END
14
15
16 ELSE IF @Element_Column=2 OR @Element_Column=3 OR @Element_Column=5
17 /*[2] LOGIKA: PRZELICZANIE W PRZÓD (Cena/Ilość/Stawka VAT -> Netto/Brutto) */
18 BEGIN
19 DECLARE @VAT AS DECIMAL(4, 2)
20
21 /*Pobranie stawki VAT z tabeli CDN.Nazwy */
22 SELECT @VAT = CAST(SUBSTRING(Naz Nazwa1, 0, CHARINDEX(' ', Naz Nazwa1)))
23
```

Kontrolki

^UserId

varchar(900) - Nazwa

^DocumentId

decimal(22, 4) - Cena

^WorkflowId

decimal(22, 4) - Ilość

^Stageld

decimal(22, 4) - Netto

^InitSenderControlName

int - Id Stanowiska (StawkaVAT)

^SenderControlName

decimal(22, 4) - Brutto

Dane_kontrahenta

POS - pozycja (opcjonalny numer wiersza)

Data_wplywu

Składnia:

SELECT

Przykład użycia:

BEGIN TRY

SELECT 'txt1' , 345.324, 345.324, 345.324, 23,

345.324

Zapisz

Zamknij

Kontrolki

Szukaj

ZAKŁADKA 1

Pozycje dokumentu

[Dane tabelaryczne] Elementy

Dane tabelaryczne

OGÓLNE LISTY INICJOWANIE WARTOŚCI OGRANICZENIE EDYCJI

Inicjowanie wartości

Inicjowanie Brak

Kontrolki powiązane Elementy (Elementy)

Dodaj

Wzór na wartość SQL ERP

Ustaw

Zapytanie SQL w kontrolce Dane tabelaryczne na zakładce Inicjowanie wartości

5. Tryb pracy: W kontrolce typu Dane tabelaryczne, na zakładce Ograniczenie edycji kolumnę „Netto” ustawiono w trybie „Do odczytu”.

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 Nazwa Nazwa	☐ Zaznacz wszystko Cena Cena	☐ Zaznacz wszystko Ilość Ilość	☒ Zaznacz wszystko Netto Netto	☐ Zaznacz wszystko StawkaVAT Stawka VAT	☐ Zaznacz wszystko Brutto Brutto
Rejestracja dokumentu	☐	☐	☐	☐	☐	☒	☐	☐
Dekretacja kosztów	☐	☐	☐	☐	☐	☒	☐	☐
Akceptacja kierownika	☐	☐	☐	☐	☐	☒	☐	☐
Księgowość	☐	☐	☐	☐	☐	☒	☐	☐
Odrzucono	☐	☐	☐	☐	☐	☒	☐	☐
Archiwum	☐	☐	☐	☐	☐	☒	☐	☐

Kolumna Netto w trybie „Do odczytu”

Uzupełniona Kontrolka Dane Tabelaryczne

Po kliknięciu ikony „+” (**Dodaj**) na karcie obiegu, **nowy wiersz kontrolki typu Dane Tabelaryczne** został automatycznie zainicjowany wartościami domyślnymi. Kolumny: **Nazwa** i **Ilość** oraz **Stawka VAT** zostały wypełnione wartościami zdefiniowanymi w zapytaniu SQL. Kolumny zostały wstępnie **wypełnione**, co skraca czas uzupełniania danych dla powtarzalnych pozycji. Dodatkowo dzięki rozbudowanemu zapytaniu SQL, dla nowo wprowadzonych wierszy istnieje możliwość **dwukierunkowego przeliczania wartości** (Netto, Brutto, Cena) co, ułatwia bieżącą aktualizację i kontrolę poszczególnych pozycji.

Pozycje dokumentu						
Elementy						
LP	Nazwa	Cena	Ilość	Netto	Stawka VAT	Brutto
1	a-vista		1,00		23%	
2	a-vista		1,00		23%	

Uzupełnione wiersze po dodaniu pozycji ikoną „+”

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

dowiedzieć się więcej?

[Sprawdź Szkolenia Comarch BPM!](#)

Czy wiesz, że w kontrolce Dane tabelaryczne wartości w poszczególnych kolumnach mogą się przeliczać w oparciu o zmianę wartości w innej kolumnie przy pomocy zapytania SQL?

Zastosowanie

Dzięki zastosowaniu tej konfiguracji, proces wprowadzania danych w **Comarch BPM** zyskuje pełną zdolność **automatycznego** kalkulowania wszędzie tam, gdzie użyta jest kontrolka tabelaryczna. Głównym celem jest **eliminacja ręcznych obliczeń**, co bezpośrednio przekłada się na mniejszą liczbę błędów i znacznie **szybszą pracę użytkownika**. Konfiguracja ta umożliwia automatyczne przeliczanie wartości w kolumnach wynikowych (np. **Netto**, **Brutto**) w oparciu o zmianę danych wejściowych (np. **Cena** lub **Ilość**) w tym samym wierszu.

Film obrazujący zastosowanie konfiguracji

<https://pomoc.comarch.pl/bpm/wp-content/uploads/2025/10/filmau>

Zawartość przykładu

- **Definicja Typu Obiegu (Workflow):** Wzorcową ścieżką obiegu dokumentu kosztowego z krokami akceptacji i konfiguracją kontrolki Dane tabelaryczne i wygenerowaniem dokumentu do Comarch ERPXL

[Kliknij tutaj, aby pobrać przykład](#)

Zasada działania

Mechanizm opiera się na ciągłym **monitorowaniu zmian wartości w kontrolce tabelarycznej** i automatycznym uruchamianiu zapytania SQL, które przelicza i aktualizuje wartości w kolumnach wynikowych.

- Kontrolka Dane Tabelaryczne jest skonfigurowana tak, aby **obserwować samą siebie** i reagować na każdą edycję wiersza.
- **Zapytanie SQL**, zaimplementowane na zakładce **Inicjowanie wartości**, jest wyzwalane przy każdej zmianie wartości w kolumnach wejściowych (**Cena**, **Ilość** lub **Stawka VAT**).
- **Przeliczania W Przód: (Cena/Ilość):** Logika **IF** w kodzie SQL jest wyzwalana przy zmianie kolumn wejściowych (np. **Cena** lub **Ilość**). W tym trybie **Netto** i **Brutto** są obliczane na podstawie wartości wejściowych i **stawki VAT**.

- **Przeliczania W Tył: (Brutto):** Logika **ELSE IF** jest wyzwalana przy zmianie kolumny **Brutto**. W tym trybie system przelicza i aktualizuje wartość w kolumnie **Cena** lub **Netto**, traktując Ilość jako stałą, zachowując tym samym wewnętrzną spójność wiersza.
- **Zapytanie SQL** pobiera niezbędne dane (np. wartość **Stawki VAT** z tabeli **CDN.Nazwy**). Następnie, za pomocą komendy **UPDATE**, bezpośrednio przelicza i aktualizuje wartości w kolumnach wynikowych (**Netto** oraz **Brutto**).
- Kolumna wynikowa **Netto** została ustawiona w trybie „Do odczytu”, aby użytkownik nie mógł ręcznie jej edytować.
- Osobna kontrolka typu **Podsumowanie** obserwuje kolumnę wynikową (**Brutto**) i zlicza jej sumę za pomocą prostego zapytania SQL (`select sum (Brutto) from @Elementy@`).

Konfiguracja przykładu

1. **Utworzono kontrolkę Tabelaryczną:** Dodano kontrolkę typu **Dane tabelaryczne** o nazwie (identyfikatorze) **Elementy** i skonfigurowano w niej następujące kolumny:

- **"Nazwa"** (identyfikator kolumny: **Nazwa**) – typ **Tekst**
- **"Cena"** (identyfikator kolumny: **Cena**) – typ **Liczba stałoprzecinkowa**
- **"Ilość"** (identyfikator kolumny: **Ilosc**) – typ **Liczba stałoprzecinkowa**
- **"Netto"** (identyfikator kolumny: **Netto**) – typ **Liczba stałoprzecinkowa**
- **"Stawka VAT"** (identyfikator kolumny: **StawkaVAT**) – typ **Lista**
- **"Brutto"** (identyfikator kolumny: **Brutto**) – typ **Liczba stałoprzecinkowa**

Pozycje dokumentu

Elementy

LP | Nazwa

Cena

Ilość

Netto

Stawka VAT

Brutto



Kolumny w kontrolce typu Dane tabelaryczne

2. W kontrolce typu **Dane tabelaryczne** na zakładce **Listy** zdefiniowano zapytanie pobierające:

▪ Stawki VAT:

```
select Naz_gidLp as Id,  
CASE  
  WHEN Naz_Nazwa='A 23.00' THEN '23%'  
  WHEN Naz_Nazwa='B 8.00' THEN '8%'  
  WHEN Naz_Nazwa='C 0.00' THEN '0%'  
  WHEN Naz_Nazwa='D 0.00' THEN 'ZW'  
  WHEN Naz_Nazwa='E 0.00' THEN 'NP'  
  WHEN Naz_Nazwa='F 7.00' THEN '7%'  
  WHEN Naz_Nazwa='G 5.00' THEN '5%'  
END  
from CDN.Nazwy  
where Naz_GIDTyp = 624 AND trim(substring(Naz_Nazwa,9,2))=''
```

Dane tabelaryczne

OGÓLNE

LISTY

INICJOWANIE WARTOŚCI

OGRANICZENIE EDYCJI

Lista 1



Nazwa

Stawka VAT

Inicjowanie

SQL ERP



Rodzaj identyfikatora

Int



[Pokaż](#) [Usuń](#)

Tryb pracy

Lista



Lista Stawka VAT w kontrolce Dane tabelaryczne

3. **Dodano obserwację kontrolki:** W kontrolce typu **Dane tabelaryczne**, na zakładce **Inicjowanie wartości**, dodano powiązanie do samej siebie.

[Dane tabelaryczne] Elementy

Dane tabelaryczne

OGÓLNE LISTY **INICJOWANIE WARTOŚCI** OGRANICZENIE EDYCJI

Inicjowanie wartości

Inicjowanie

Brak

Kontrolki powiązane Elementy (Elementy) 

[Dodaj](#)

Wzór na wartość

SQL ERP

[Ustaw](#)

Powiązanie kontrolki do samej siebie

4. **Dodano zapytanie SQL:** W kontrolce typu **Dane tabelaryczne**, na zakładce **Inicjowanie wartości** w sekcji **Wzór na wartość SQL ERP** dodano zapytanie, które pobiera stawki VAT z bazy XL i na podstawie **Ilości** i **Ceny** wylicza **wartość Netto**, po uzupełnieniu **Stawki VAT** wylicza **wartość Brutto**, a po zmianie wartości **Brutto** wylicza **Cenę** i **wartość Brutto**.

```
/*Deklaracja Stawki VAT */
DECLARE @VAT AS DECIMAL(4, 2)
```

```
/* Pobranie stawki VAT z tabeli CDN.Nazwy na podstawie kolumny
StawkaVAT w bieżącym wierszu */
SELECT @VAT = CAST(SUBSTRING(Naz_Nazwa1, 0, CHARINDEX(' ',
Naz_Nazwa1)) AS DECIMAL(4, 2)) / 100
```

```
FROM CDN.Nazwy
WHERE Naz_GIDTyp = 624
AND Naz_GIDLp = (
SELECT [StawkaVAT]
FROM @Elementy@
WHERE POS = @Elementy_Row@
)
```

```
/* JEŚLI ZMIENIONO KOLUMNY: 2 (Cena), 3 (Ilość), 5 (Stawka
VAT) */
```

```
IF @Elementy_Column@=2 OR @Elementy_Column@=3 OR
@Elementy_Column@=5
BEGIN
/* PRZELICZANIE W PRZÓD (Cena/Ilość/Stawka VAT ->
Netto/Brutto) */
```

```
/*Wyliczenie i aktualizacja netto i brutto */
```

```
UPDATE @Elementy@
SET
[Netto] = (Ilosc * Cena),
[Brutto] = (Ilosc * Cena) * (@VAT + 1)
WHERE POS = @Elementy_Row@
```

```
/* !!! OBOWIĄZKOWE: Odświeżenie wiersza !!! */
```

```
SELECT * FROM @Elementy@
END
```

```
/* JEŚLI ZMIENIONO KOLUMNĘ: 6 (Brutto) */
```

```
IF @Elementy_Column@=6
BEGIN
/* PRZELICZANIE W TYŁ (Brutto -> Cena/Netto) */
```

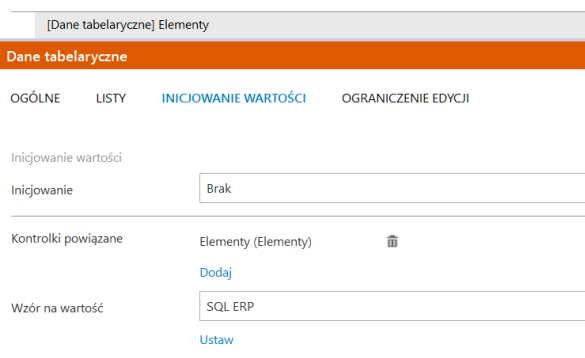
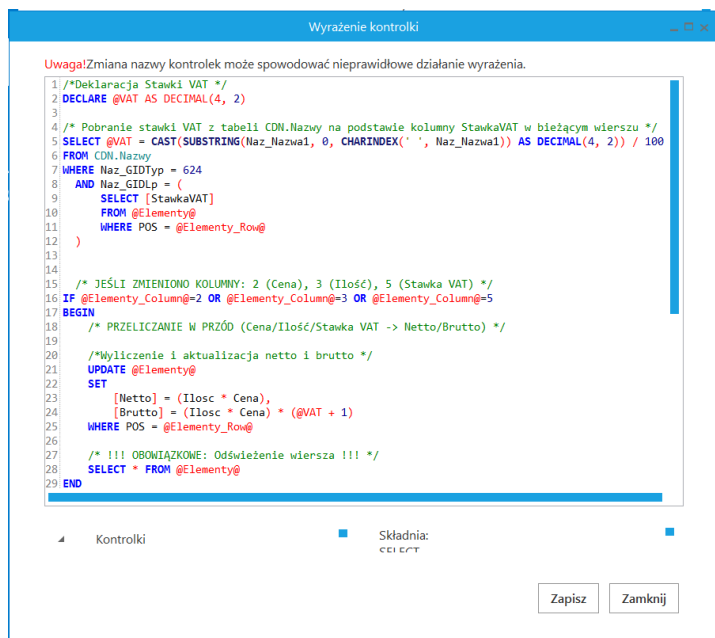
```
/* Aktualizacja ceny i netto (Ilość stała) */
```

```
UPDATE @Elementy@
SET
[Cena] = [Brutto] / (@VAT + 1), -- Obliczenie Ceny Netto z
Brutto
[Netto] = [Ilosc] * ([Brutto] / (@VAT + 1)) -- Obliczenie
Wartości Netto
WHERE POS = @Elementy_Row@
```

```

/* !!! OBOWIĄZKOWE: Odświeżenie wiersza !!! */
SELECT * FROM @Elementy@
END

```



Zapytanie SQL w kontrolce Dane tabelaryczne na zakładce Inicjowanie wartości

5. Tryb pracy: W kontrolce typu **Dane tabelaryczne**, na zakładce **Ograniczenie edycji** kolumnę „**Netto**” ustawiono w trybie „Do odczytu”.

Dane tabelaryczne

OGÓLNE

LISTY

INICJOWANIE WARTOŚCI

OGRANICZENIE EDYCJI

Zaznaczenie opcji spowoduje zablokowanie funkcjonalności lub edycji poniższych parametrów.

	☐ Zaznacz wszystko	☐ Zaznacz wszystko	☐ Zaznacz wszystko	☐ Zaznacz wszystko	☐ Zaznacz wszystko	☒ Zaznacz wszystko	☐ Zaznacz wszystko	☐ Zaznacz wszystko
Etap	Zablokuj dodanie wierszy	Zablokuj usuwanie wierszy	Pozycja Pozycja	Cena Cena	Ilość Ilość	Netto Netto	StawkaVAT Stawka VAT	Brutto Brutto
Rejestracja	☐	☐	☐	☐	☐	☒	☐	☐
Weryfikacja	☐	☐	☐	☐	☐	☒	☐	☐
Koniec	☐	☐	☐	☐	☐	☒	☐	☐

Kolumna Netto w trybie „Do odczytu”

6. **Utworzono kontrolkę liczbową:** Dodano kontrolkę typu **Liczba stałoprzecinkowa** o nazwie (identyfikatorze) **Podsumowanie**, w której dodano powiązanie do kontrolki **Elementy** i skonfigurowano w niej zapytanie, które zlicza wartość z kolumny **Brutto** z kontrolki tabelarycznej:

```
select sum (Brutto) from @Elementy@
```

The screenshot displays the configuration interface for a 'Liczba stałoprzecinkowa' (Fixed-point number) control. The interface is divided into two main panels.

Left Panel (Wyrażenie kontrolki):

- Uwaga!** Zmiana nazwy kontrolki może spowodować nieprawidłowe działanie wyrażenia.
- SQL Query:** `1 select sum (Brutto) from @Elementy@`
- Kontrolki:** A list of controls including `^DocumentCor` and `^UserId`.
- Składnia:** `SELECT decimal (22,4) decimal – wartość stałoprzecinkowa. Przykład: BEGIN TRY`
- Buttons:** `Zapisz` and `Zamknij`.

Right Panel (Liczba stałoprzecinkowa):

- Information:** Kontrolkę można uzupełnić liczbą złożoną z maksymalnie 18 cyfr przed przecinkiem i 6 miejsc po przecinku.
- Fields:**
 - Nazwa wyświetlana:** Podsumowanie
 - Nazwa (identyfikator):** Podsumowanie
 - Liczba miejsc po przecinku, do której będzie zaokrąglana wartość:** 2
 - Prezentuj na całej szerokości:** ☐
 - Ustaw kontrolkę na początku wiersza:** ☐
 - Pokaż na liście:** ☐
- Inicjowanie wartości:**
 - Inicjowanie:** Brak
 - Dokument:**
 - Pole:**
- Zmiana wartości (Obserwator):**
 - Kontrolki powiązane:** Elementy (Elementy) [trash icon]
 - Dodaj:**
 - Wzór na wartość:** SQL OD
 - Pokaż Usuń:**

Kontrolka Liczbowa Podsumowanie

Uzupełniona Kontrolka Dane Tabelaryczne

Po wprowadzeniu wartości w kontrolce typu **Dane tabelaryczne** w kolumnach **Cena**, **Ilość** lub **Stawka VAT** kontrolka typu **Dane Tabelaryczne** została **automatycznie zaktualizowana**. Dla każdej pozycji, system automatycznie przeliczył i uzupełnił wartości w kolumnach **Netto** oraz **Brutto** w tym samym wierszu. Możliwe jest przeliczanie wartości **w przód** (Cena/Ilość → Netto/Brutto) oraz **w tył** (Brutto → Cena/Netto). Wartości w kolumnach

wynikowych (**Netto**, **Brutto**) są widoczne, ale kolumna **Netto** jest zablokowana do ręcznej edycji (tryb „Do odczytu”), co gwarantuje poprawność danych. Kontrolka **Podsumowanie** wyświetla poprawną sumę końcową z kolumny **Brutto** kontrolki typu Dane tabelaryczne.

Karta obiegu

Elementy

LP	Pozycja	Cena	Ilość	Netto	Stawka VAT	Brutto
1	T1	3,00	4,00	12,00	23%	14,76
2	T2	6,00	8,00	48,00	23%	59,04
3	T3	90,00	4,00	360,00	8%	388,80

+

✕

Podsumowanie

462,60

Kontrolka typu Dane tabelaryczne z uzupełnionymi kolumnami

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

[Sprawdź Szkolenia Comarch BPM!](#)

Czy wiesz, że po wybraniu z listy dokumentu PZ w kontrolce Dane tabelaryczne można pobrać do poszczególnych kolumn informacje z tego dokumentu?

Zastosowanie

Dzięki tej konfiguracji, proces obsługi **dokumentów w Comarch BPM** zyskuje zaawansowaną zdolność automatycznego pobierania danych z dokumentów **ERPXL**. Wdrożenie tego wzorca znacznie **przyspiesza procesy** zakupowe, ponieważ eliminuje konieczność ręcznego wyszukiwania i przepisywania danych z dokumentów (w przykładzie dokument Przyjęcia Zewnętrzne (PZ)). System automatycznie pobiera **kluczowe informacje** nagłówkowe (**Kontrahent, Wartość Netto**) z wybranego dokumentu PZ, uzupełniając **kontrolkę tabelaryczną** i wprowadzając **dane** bezpośrednio do obiegu.

Film obrazujący zastosowanie konfiguracji

<https://pomoc.comarch.pl/dms/wp-content/uploads/2025/06/pz2.mp4>

Zawartość przykładu 

- **Definicja Typu Obiegu (Workflow):** Wzorcowa ścieżka obiegu dokumentu z konfiguracją kontrolki typu Dane tabelaryczne.

[Kliknij tutaj, aby pobrać przykład](#)

Zasada działania

Mechanizm opiera się na **obserwowaniu kolumny typu Lista** (lista wyboru dokumentu PZ) i uruchamianiu złożonego zapytania SQL, które **pobiera** powiązane **dane nagłówkowe** z tabel **Comarch ERP XL**.

- W kontrolce typu **Dane Tabelaryczne** zdefiniowano kolumnę **ListaPZ** (typ: Lista) służącą do wyboru dokumentu **PZ** oraz kolumny wynikowe: **PZ** (typ: Dokument elektroniczny), **Kontrahent** (typ: Kontrahent) i **Netto** (typ: Liczba stałoprzecinkowa).
- Na zakładce **Listy** zdefiniowano zapytanie pobierające wszystkie **dokument PZ** (o typie Trn_GIdTyp=1489) z tabeli **cdn.TraNag**.
- Na zakładce **Inicjowanie wartości** dodano obserwację kontrolki przez samą siebie.
- Zapytanie **SQL** (Wzór na wartość SQL ERP) jest wyzwalane przy każdej zmianie kolumny **ListaPZ**.
- Zapytanie **SQL** pobiera **GIDNumer** wybranego dokumentu PZ i na jego podstawie odpytuje tabelę **cdn.TraNag** oraz **cdn.KntKarty** (dla Kontrahenta).
- Polecenie **UPDATE @DokumentyPZ@ SET** aktualizuje kolumny wynikowe w wierszu.
- Osobna kontrolka typu **SumaNettoPZ** obserwuje kolumnę (**Netto**) i zlicza jej sumę za pomocą prostego zapytania SQL (select sum (Netto) from @DokumentyPZ@).

Konfiguracja przykładu

1. Utworzono kontrolkę Tabelaryczną: Dodano kontrolkę typu **Dane tabelaryczne** o nazwie (identyfikatorze) **DokumentyPZ** i skonfigurowano w niej następujące kolumny:

- **"ListaPZ"** (identyfikator kolumny: **ListaPZ**) – typ **Lista**
- **"PZ"** (identyfikator kolumny: **PZ**) – typ **Dokument elektroniczny**
- **"Kontrahent"** (identyfikator kolumny: **Kontrahent**) – typ **Kontrahent**
- **"Netto"** (identyfikator kolumny: **Netto**) – typ **Liczba stałoprzecinkowa**

Karta obiegu					
DokumentyPZ					
LP	ListaPZ	PZ	Kontrahent	Netto	
 					

Kolumny w kontrolce Dane tabelaryczne

2. W kontrolce typu **Dane tabelaryczne** na zakładce **Listy** zdefiniowano zapytanie pobierające:

- **Listę dokumentów PZ z Comarch ERPXL:**

```
select trn_gidnumer,cdn.numerdokumentu(trN_GIDTyp,TrN_SpiTyp,
Trn_TRNTyp,TrN_TrNNumer,TrN_TrNRok, TrN_TrNSeria, TrN_trNMiesiac) from cdn.TraNag where
Trn_GIdTyp=1489
```

[Dane tabelaryczne] DokumentyPZ

Dane tabelaryczne

OGÓLNE **LISTY** INICJOWANIE WARTOŚCI OGRANICZENIE EDYCJI

Lista 1 

Nazwa

ListaPZ

Inicjowanie

SQL ERP 

Rodzaj identyfikatora

Int 

[Pokaż](#) [Usuń](#)

Tryb pracy

Lista 

[Dodaj](#)

Lista pobierająca dokumenty PZ w kontrolce typu Dane tabelaryczne

3. Dodano obserwację kontrolki: W kontrolce typu **Dane tabelaryczne**, na zakładce **Inicjowanie wartości**, dodano powiązanie do samej siebie.

Dane tabelaryczne

OGÓLNE

LISTY

INICJOWANIE WARTOŚCI

OGRANICZENIE EDYCJI

Inicjowanie wartości

Inicjowanie

Brak

Kontrolki powiązane

DokumentyPZ (DokumentyPZ)



Dodaj

Wzór na wartość

SQL ERP

Ustaw

Powiązanie kontrolki do samej siebie

4. Dodano zapytanie SQL: W kontrolce typu **Dane tabelaryczne**, na zakładce **Inicjowanie wartości** w sekcji **Wzór na wartość SQL ERP** dodano zapytanie, które automatycznie podpinia dokument elektroniczny w kolumnie **PZ**, oraz wczytuje kluczowe dane nagłówkowe (**Kontrahent** i **Netto**) bezpośrednio z wybranego dokumentu. Logika ta polega na trzykrotnym użyciu instrukcji **UPDATE @DokumentyPZ@ SET** w jednym bloku kodu. Każda instrukcja, oparta o połączenia (JOIN) z tabelami nagłówkowymi ERP XL (**cdn.TraNag**, **cdn.KntKarty**), pobiera i aktualizuje inną wartość w bieżącym wierszu tabeli.

```
IF @DokumentyPZ_Column@ = 1
```

BEGIN

/*[1] Zapisanie GID dokumentu PZ do kolumny PZ_ID w celu dalszych zapytań*/

UPDATE @DokumentyPZ@ SET [PZ_ID] =

(

SELECT [ListaPZ] FROM @DokumentyPZ@ WHERE POS = @DokumentyPZ_Row@

)

WHERE POS = @DokumentyPZ_Row@

/* [2] Pobranie Kontrahenta (Knt_GidNumer) z tabel nagłówkowych ERP XL*/

UPDATE @DokumentyPZ@ SET [Kontrahent] =

(

SELECT Knt_GidNumer FROM cdn.TraNag

JOIN cdn.kntkarty ON TrN_KntNumer = Knt_GIDNumer

WHERE TrN_GIDNumer = [ListaPZ]

)

WHERE POS = @DokumentyPZ_Row@

/*[3] Pobranie Wartości Netto (TrN_NettoP) z tabel TraNag/TraElem*/

UPDATE @DokumentyPZ@ SET [Netto] = (

SELECT TOP 1 TrN_NettoP

FROM cdn.TraNag

```
JOIN cdn.TraElem ON TrE_GIDNumer = TrN_GIDNumer
```

```
WHERE TrN_GIDNumer = [ListaPZ]
```

```
)
```

```
WHERE POS = @DokumentyPZ_Row@
```

```
/*!!! OBOWIĄZKOWE: Odświeżenie wiersza w kontrolce Dane  
tabelaryczne !!!*/
```

```
SELECT * FROM @DokumentyPZ@
```

```
END
```

Wyrażenie kontrolki

Uwaga! Zmiana nazwy kontrolki może spowodować nieprawidłowe działanie wyrażenia.

```
1 IF @DokumentyPZ_Column@ = 1
2 BEGIN
3     /*[1] Zapisanie GID dokumentu PZ do kolumny PZ_ID w celu dalszych zapytań*/
4     UPDATE @DokumentyPZ@ SET [PZ_ID] =
5     (
6         SELECT [ListaPZ] FROM @DokumentyPZ@ WHERE POS = @DokumentyPZ_Row@
7     )
8     WHERE POS = @DokumentyPZ_Row@
9
10    /* [2] Pobranie Kontrahenta (Knt_GIDNumer) z tabel nagłówekowych ERP XL*/
11    UPDATE @DokumentyPZ@ SET [Kontrahent] =
12    (
13        SELECT Knt_GIDNumer FROM cdn.TraNag
14        JOIN cdn.kntkarty ON TrN_KntNumer = Knt_GIDNumer
15        WHERE TrN_GIDNumer = [ListaPZ]
16    )
17    WHERE POS = @DokumentyPZ_Row@
18
19    /*[3] Pobranie Wartości Netto (TrN_NettoP) z tabel TraNag/TraElem*/
20    UPDATE @DokumentyPZ@ SET [Netto] = (
21        SELECT TOP 1 TrN_NettoP
22        FROM cdn.TraNag
23        JOIN cdn.TraElem ON TrE_GIDNumer = TrN_GIDNumer
24        WHERE TrN_GIDNumer = [ListaPZ]
25    )
26    WHERE POS = @DokumentyPZ_Row@
27
28    /*!!! OBOWIĄZKOWE: Odświeżenie wiersza w kontrolce Dane tabelaryczne !!!*/
29    SELECT * FROM @DokumentyPZ@
30 END
31
```

Kontrolki

Szukaj

ZAKŁADKA 1

Karta obiegu

[Dane tabelaryczne] DokumentyPZ

[Liczba stałoprzecinkowa] Suma Netto PZ

Dane tabelaryczne

OGÓLNE LISTY INICJOWANIE WARTOŚCI OGRANICZENIE EDYCJI

Inicjowanie wartości

Inicjowanie Brak

Kontrolki powiązane DokumentyPZ (DokumentyPZ)

Dodaj

Wzór na wartość SQL ERP

Ustaw

Zapisz Zamknij

Składnia:
SELECT
int - Id Stanowiska (ListaPZ)
int - int - PZ_TYPE, PZ_ID
int - ID (Kontrahent)
decimal(22, 4) - Netto

Zapytanie SQL w kontrolce typu Dane tabelaryczne na zakładce Inicjowanie wartości

5.Tryb pracy: W kontrolce typu Dane tabelaryczne, na zakładce

Ograniczenie edycji kolumnę „Kontrahent” oraz „Netto” ustawiono w trybie „Do odczytu”.

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 ListaPZ ListaPZ	☐ Zaznacz wszystko PZ PZ	☒ Zaznacz wszystko Kontrahent Kontrahent	☒ Zaznacz wszystko Netto Netto
Wprowadzenie	☐	☐	☐	☐	☒	☒
Weryfikacja	☐	☐	☐	☐	☒	☒
Archiwizacja	☐	☐	☐	☐	☒	☒

Kolumna Netto w trybie „Do odczytu”

6.Utworzono kontrolkę liczbową: Dodano kontrolkę typu **Liczba stałoprzecinkowa** o nazwie (identyfikatorze) **SumaNettoPZ**, w której dodano powiązanie do kontrolki **DokumentyPZ** i skonfigurowano w niej zapytanie, które zlicza wartość z kolumny **Netto** z kontrolki tabelarycznej:

```
select sum (Netto) from @DokumentyPZ@
```

Wyrażenie kontrolki

Uwaga!
Zmiana nazwy kontrolki może spowodować nieprawidłowe działanie wyrażenia.

```
1 select sum (Netto) from @DokumentyPZ@
```

Kontrolki

- ^Document
- ^UserId
- ^Document
- ^Workflow
- ^Stageld

Składnia:

```
SELECT decimal
(22,4)
decimal – wartość
stałoprzecinkowa.
Przykład:
BEGIN TRY
SELECT 1000.2588
END TRY
BEGIN CATCH
```

Zapisz

Zamknij

[Liczba stałoprzecinkowa] Suma Netto PZ

Liczba stałoprzecinkowa

Kontrolkę można uzupełnić liczbą złożoną z maksymalnie 18 cyfr przed przecinkiem i 6 miejsc po przecinku

Nazwa wyświetlana

SumaNettoPZ

Nazwa (identyfikator)

SumaNettoPZ

Liczba miejsc po przecinku, do której będzie zaokrąglana wartość

2

Prezentuj na całej szerokości

☐

Ustaw kontrolkę na początku wiersza

☐

Pokaż na liście

☐

Inicjowanie wartości

Inicjowanie

Brak

Dokument

Pole

Zmiana wartości (Obserwator)

Kontrolki powiązane

DokumentyPZ (DokumentyPZ)

Dodaj

Wzór na wartość

SQL OD

Pokaż

Usuń

Kontrolka Liczbowa Suma Netto PZ

Uzupełniona Kontrolka Dane Tabelaryczne

Po wybraniu z listy dokumentu **PZ** w kontrolce typu **Dane tabelaryczne**, wiersz został automatycznie zaktualizowany. W efekcie, w drugiej kolumnie podpiął się **dokument elektroniczny PZ** oraz zostały uzupełnione kluczowe dane nagłówkowe: **Kontrahent** i **Wartość Netto**. Kontrolka **SumaNettoPZ** wyświetla poprawną sumę końcową z kolumny **Netto** wszystkich podpiętych dokumentów **PZ**.

Karta obiegu

DokumentyPZ

LP	ListaPZ	PZ	Kontrahent	Netto
1	PZ-1/25	  PZ-1/25	  [COMARCH] COMARCH SPÓŁKA AKCYJNA	70,00
2	PZ-2/25	  PZ-2/25	  [ALOZA] Aloza S.A	80,00



Suma Netto PZ

150,00

Kontrolka typu Dane tabelaryczne z uzupełnionymi kolumnami

Dokument - Przyjęcie zewnętrzne [PZ-2/25] (rozliczony) - zostanie zmieniony

Ogólne Kontrahent Kaucje Nagłówek Księgowość VAT Intrastat Atrybuty Załączniki ☐ Do bufora

Kontrahent ALOZA Warszawa **Netto:** 80,00
Aloza S.A. **Brutto:** 98,40
Kaucje: 0,00

Kraj wysyłki PL Polska

Transakcja: Krajowa

Dostawa:

Rodzaj transportu: 3 Transport drogowy

☐ Płatność:

Lp.	Towar	Ilość	Jm.	Cena netto	Cena brutto	Wartości	
						Netto	Brutto
☐	00001 TOWAR2	10,000	szt.	8,0000 PLN	9,8400 PLN	80,00 PLN	98,40 PL

Filtr:

Docelowy MAG ☐ Aktualizuj kaucje

Dokument PZ podpisany w kontrolce Dane tabelarycznej

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

[Sprawdź Szkolenia Comarch BPM!](#)

[Powrót do początku artykułu](#)

Czy wiesz, że po dodaniu jednego lub wielu dokumentów PZ w kontrolce Dane tabelaryczne można do osobnej kontrolki Dane Tabelaryczne do poszczególnych kolumn pobrać informacje z tych dokumentów?

Dodanie dokumentu PZ w kontrolce Dane tabelaryczne o nazwie „PZ z ERP XL” spowoduje uzupełnienie kontrolki „ElementyPZ” informacjami z dowiązanego dokumentu.

<https://pomoc.comarch.pl/dms/wp-content/uploads/2025/06/pznew.mp4>

Konfiguracja przykładu – przykład dotyczy Comarch DMS we

współpracy z Comarch ERP XL:

1. W definicji typu obiegu w kontrolce typu Dane tabelaryczne o nazwie "PZ z ERPXL" (identyfikator kontrolki: PZ) zdefiniowano kolumnę:

- "PZ" (identyfikator kolumny: PZ) – typ **Dokument elektroniczny**

<< ZAKŁADKA 1

Karta obiegu
^

[Tekst] Test

[Dane tabelaryczne] PZ z ERPXL

[Dane tabelaryczne] ElementyPZ

Dane tabelaryczne

Nazwa wyświetlana	PZ z ERPXL
Nazwa (identyfikator)	PZ
Prezentuj na całej szerokości	☒
Ustaw kontrolkę na początku wiersza	☐
Liczba prezentowanych wierszy	 3 ▲ ▼
Kolumny	
_____ Kolumna 1 _____	
Dokument elektroniczny	▼ ▲ 🗑️
Nazwa kolumny	PZ
Identyfikator kolumny	PZ
Nazwa spółki	Domyślna spółka ▼
Typ	Przyjęcie zewnętrzne ▼

[Dodaj](#)

Konfiguracja kontrolki elektronicznej w Danych Tabelarycznych

2. W definicji typu obiegu w kontrolce typu Dane tabelaryczne o nazwie "ElementyPZ" (identyfikator kontrolki: ElementyPZ) zdefiniowano kolumny:

- "Ilość" (identyfikator kolumny: Ilosc) – typ Liczba

stałoprzecinkowa

- "JM" (identyfikator kolumny: JM) – typ **Lista**
- "Cena" (identyfikator kolumny: Cena) – typ **Liczba stałoprzecinkowa**
- "Netto" (identyfikator kolumny: Netto) – typ **Liczba stałoprzecinkowa**

3. W **kontrolce typu Dane tabelaryczne** o nazwie **"ElementyPZ"** na zakładce **Listy** dodano listę o nazwie „Lista miar” i zdefiniowano dla niej zapytanie pobierające jednostki miary z Comarch ERPXL:

```
SELECT Naz_GIDLp,Naz_Nazwa from cdn.Nazwy where Naz_GIDTyp=144
```

The screenshot displays the application's configuration interface. On the left, the 'Zapytanie SQL' (SQL Query) editor shows a query: `SELECT Naz_GIDLp,Naz_Nazwa from cdn.Nazwy where Naz_GIDTyp=144`. Below the editor, a 'Kontrolki' (Controls) list includes fields like ^UserId, ^DocumentId, ^ParentDocumentId, and ^WorkflowId. To the right, the 'Dane tabelaryczne' (Table Data) section is active, showing the 'LISTY' (Lists) tab. Under 'ZAKŁADKA 1' (Tab 1), there's a 'Karta obiegu' (Flow Card) with a '[Dane tabelaryczne] PZ z ERPXL' and '[Dane tabelaryczne] ElementyPZ' entry. The 'LISTY' tab shows a list named 'Lista 1'. The configuration for 'Lista 1' includes: 'Nazwa' (Name) set to 'Lista miar', 'Inicjowanie' (Initiation) set to 'SQL ERP', 'Rodzaj identyfikatora' (Identifier Type) set to 'Int', and 'Nazwa spółki' (Company Name) set to 'Domyślna spółka'. The 'Tryb pracy' (Work Mode) is set to 'Lista'. Buttons for 'Zapisz' (Save) and 'Zamknij' (Close) are visible at the bottom of the SQL editor.

Definiowanie zapytania dla listy „Lista miar”

4. W definicji typu obiegu w kontrolce typu Dane tabelaryczne o nazwie "ElementyPZ" (identyfikator kontrolki: ElementyPZ) w kolumnie "JM" (identyfikator kolumny: JM) w polu „Listy” wybrano listę „Lista miar”

Dane tabelaryczne

OGÓLNE

LISTY

INICJOWANIE WARTOŚCI

OGRANICZENIE EDYCJI

Nazwa wyświetlana

ElementyPZ

Nazwa (identyfikator)

ElementyPZ

Prezentuj na całej szerokości

☒

Ustaw kontrolkę na początku wiersza

☐

Liczba prezentowanych wierszy

3

Kolumny

Liczba stałoprzecinkowa

Nazwa kolumny

Ilość

Identyfikator kolumny

Ilosc

Liczba miejsc po przecinku, do której będzie zaokrąglana wartość

2

Kolumna 1

Lista

Nazwa kolumny

JM

Identyfikator kolumny

JM

Listy

Lista miar

Kolumna 2

Kolumna 3

Liczba stałoprzecinkowa

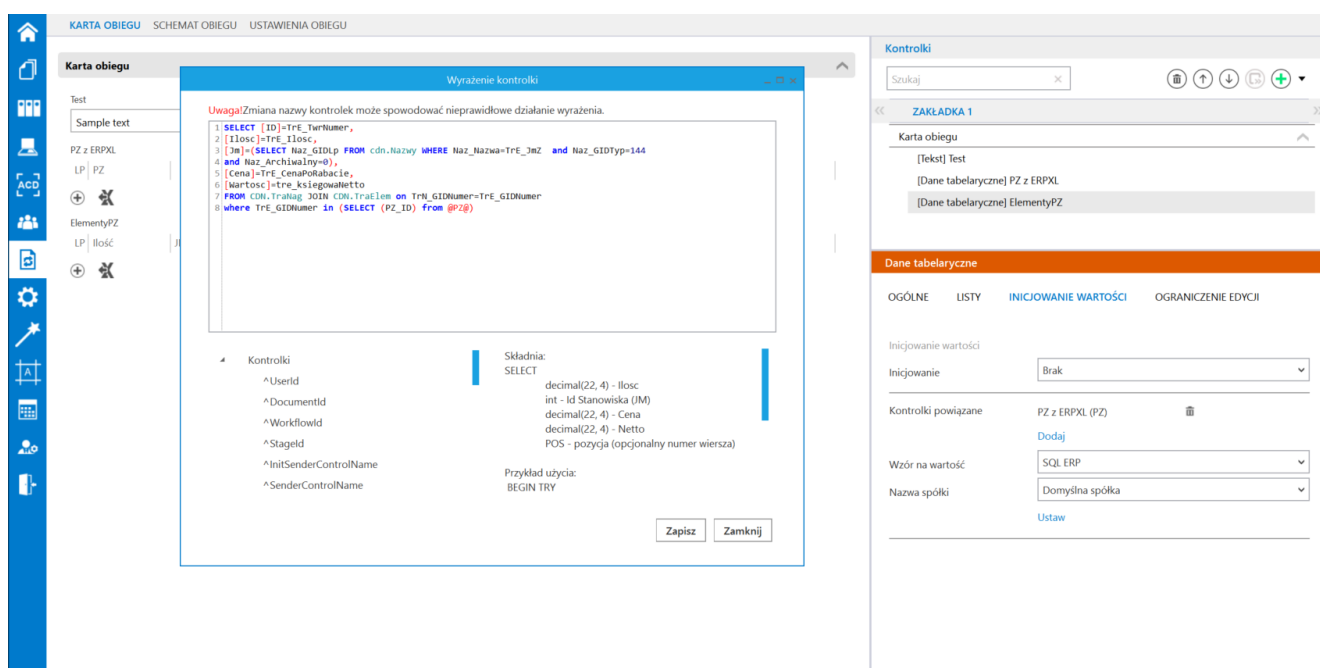
Nazwa kolumny

Cena

Wybór listy „Lista miar” dla kolumny „JM”

5. W **kontrolce typu Dane tabelaryczne** o nazwie **"ElementyPZ"** na zakładce **Inicjowanie wartości** w polu „Kontrolki powiązane” dodano obserwację **kontrolki typu Dane tabelaryczne o nazwie "PZ z ERPXL"** oraz w polu „Wzór na wartość” wybrano **„SQL ERP”** i zdefiniowano zapytanie:

```
SELECT [ID]=TrE_TwrNumer,  
[Ilosc]=TrE_Ilosc,  
[Jm]=(SELECT Naz_GIDLp FROM cdn.Nazwy WHERE Naz_Nazwa=TrE_JmZ  
and Naz_GIDTyp=144  
and Naz_Archiwalny=0),  
[Cena]=TrE_CenaPoRabacie,  
[Wartosc]=tre_ksiegowaNetto  
FROM CDN.TraNag JOIN CDN.TraElem on TrN_GIDNumer=TrE_GIDNumer  
where TrE_GIDNumer in (SELECT (PZ_ID) from @PZ@)
```



Zapytanie w kontrolce Dane Tabelaryczne – zakładka Inicjowanie wartości

6. **Zapisano zmiany**, klikając w przycisk



[Zapisz].

Po wprowadzonych powyżej zmianach, **gdy na dokumencie Comarch DMS zostaje dodany dokument PZ z Comarch ERPXL do kontrolki typu Dane tabelaryczne o nazwie "PZ z ERPXL", wówczas informacje z tego dokumentu zostają pobrane do kontrolki typu Dane tabelaryczne o nazwie „ElementyPZ”.**

DT_OBSERWACJA_Elekt/7/7/2021

Dokumenty Spółka : Domyślna spółka Data utworzenia : 13-07-2021

Karta obiegu

Test

PZ z ERPXL

LP	PZ
1	PZ-11/21

ElementyPZ

LP	Towar	Ilość	JM	Cena	Netto
1	[W3] w	1,00	tona	87,50	87,50
2	[USLUK	1,00	szt.	100,00	100,00

Dodanie dokumentu PZ i uzupełnienie elementów w Danach Tabelarycznych

Dokumenty

Spółka : Domyślna spółka

Data utworzenia : 13-07-2021

Karta obiegu

Test

PZ z ERPXL

LP | PZ

1 PZ-11/21

2 PZ-10/21

ElementyPZ

LP	Towar	Ilość	JM	Cena	Netto
1	[ZŁOM	1,00	szt.	1234,00	1234,00
2	[T1] T1	1,00	szt.	408,39	408,39
3	[W3] w	1,00	tona	87,50	87,50

Dodanie kolejnego dokumentu PZ w Danych Tabelarycznych

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](#)

Czy wiesz, że kontrolkę typu Dane Tabełaryczne można automatycznie wypełnić pozycjami dokumentu zaczytanego przez Comarch OCR przy współpracy z Comarch ERP XL?

Zastosowanie

Dzięki zastosowaniu tej konfiguracji, **proces obiegu faktur kosztowych zyskuje pełną automatyzację**. Głównym celem jest eliminacja ręcznego wprowadzania pozycji dokumentów – dane zaczytane przez Comarch OCR są automatycznie mapowane i wypełniają kontrolkę tabelaryczną na kacie obiegu. W efekcie **skracamy czas obsługi faktury**, minimalizujemy błędy przepisywania i zapewniamy, że proces akceptacji bazuje na **poprawnych i spójnych danych** przed ich zaksięgowaniem.

Zawartość przykładu

- **Definicja Typu Obiegu (Workflow):** Wzorcowa ścieżka obiegu dokumentu kosztowego z krokami akceptacji, konfiguracją kontrolki Dane tabelaryczne i wygenerowaniem dokumentu do Comarch ERPXL
- **Punkt konfiguracyjny ACD:** Wzorcowa konfiguracja, która jest odpowiedzialna za pobranie ID dokumentu OCR, Załącznika oraz wszystkich kluczowych danych

nagłówkowych (NIP, Kontrahent, Dane kontrahenta, Data zakupu, Data wystawienia, Termin płatności, Forma płatności, Waluta) oraz inicjację dokumentu obiegowego.

[Kliknij tutaj, aby pobrać przykład](#)

Zasada działania

Mechanizm opiera się na **pobieraniu danych z bazy Comarch BPM, które zostały zapisane po przetworzeniu ich przez Comarch OCR w punkcie ACD.**

- Konfiguracja wykorzystuje **ID dokumentu OCR** (zapisane w dedykowanej kontrolce typu **Liczba całkowita**) jako identyfikator do pobrania danych. Kontrolka tabelaryczna **obserwuje to pole**, reagując na jego wypełnienie.
- Kluczowym elementem jest **zapytanie SQL** umieszczone na zakładce **Inicjowanie Wartości** kontrolki tabelarycznej. Zapytanie jest filtrowane za pomocą parametru **@ID_OCR@**, dzięki czemu pobiera ono wiersze z właściwego dokumentu OCR z tabeli **do.OC_Document_Items** w bazie BPM.
- Na podstawie zwróconych wierszy z tabeli **do.OC_Document_Items**, kontrolka **Dane tabelaryczne** jest automatycznie wypełniana pozycjami faktury (Nazwa, Cena, Ilość, Jednostka miary, Netto, Stawka VAT, Brutto, Kwota VAT).
- W zapytaniu SQL zawarta jest niezbędna **logika CASE WHEN**, która konwertuje wartości tekstowe (np. Stawki VAT oraz Jednostki miary) na identyfikatory liczbowe, których BPM wymaga do poprawnego działania mechanizmów systemowych i wyświetlania danych.

Konfiguracja przykładu

1. **Utworzono kontrolkę ID_OCR** : Na karcie obiegu dodano kontrolkę typu **Liczba całkowita** o nazwie (identyfikatorze) **ID_OCR**, do której będzie zapisywane ID OCR-owanego dokumentu.

Techniczne

[Liczba całkowita] ID_OCR

Liczba całkowita

i

Kontrolkę można uzupełnić liczbą z zakresu od -2 147 483 648 do 2 147 483 647

Nazwa wyświetlana

ID_OCR

Nazwa (identyfikator)

ID_OCR

Prezentuj na całej szerokości

☐

Ustaw kontrolkę na początku wiersza

☒

Pokaż na liście

☐

Kontrolka Liczba całkowita ID_OCR

2. **Utworzono kontrolkę Tabelaryczną**: Dodano kontrolkę typu **Dane tabelaryczne** o nazwie (identyfikatorze) **Elementy** i skonfigurowano w niej następujące kolumny:

- **"Nazwa"** (identyfikator kolumny: **Nazwa**) – typ Tekst
- **"Cena"** (identyfikator kolumny: **Cena**) – typ Liczba stałoprzecinkowa
- **"Ilość"** (identyfikator kolumny: **Ilosc**) – typ Liczba stałoprzecinkowa
- **"Jednostka miary"** (identyfikator kolumny: **JM**) – typ

Lista

- **"Netto"** (identyfikator kolumny: Netto) – typ Liczba stałoprzecinkowa
- **"Stawka VAT"** (identyfikator kolumny: StawkaVAT) – typ Lista
- **"Brutto"** (identyfikator kolumny: Brutto) – typ Liczba stałoprzecinkowa
- **"Kwota VAT"** (identyfikator kolumny: KwotaVAT) – typ Liczba stałoprzecinkowa

Pozycje dokumentu

Elementy

LP	Nazwa	Cena	Ilość	Jednostka miary	Netto	Stawka VAT	Brutto	Kwota VAT
+ X								

Kolumny w kontrolce Dane tabelaryczne

3. W kontrolce typu **Dane tabelaryczne** na zakładce **Listy** zdefiniowano zapytanie pobierające:

- **Stawki VAT:**

```
select Naz_gidLp as Id,
CASE
WHEN Naz_Nazwa='A 23.00' THEN '23%'
WHEN Naz_Nazwa='B 8.00' THEN '8%'
WHEN Naz_Nazwa='C 0.00' THEN '0%'
WHEN Naz_Nazwa='D 0.00' THEN 'ZW'
WHEN Naz_Nazwa='E 0.00' THEN 'NP'
WHEN Naz_Nazwa='F 7.00' THEN '7%'
WHEN Naz_Nazwa='G 5.00' THEN '5%'
END
from CDN.Nazwy
where Naz_GIDTyp = 624 AND trim(substring(Naz_Nazwa,9,2))= ''
```

[Dane tabelaryczne] Elementy

Dane tabelaryczne

OGÓLNE LISTY INICJOWANIE WARTOŚCI OGRANICZENIE EDYCJI

Lista 1 

Nazwa

Stawka VAT

Inicjowanie

SQL ERP 

Rodzaj identyfikatora

Int 

[Pokaż](#) [Usuń](#)

Tryb pracy

Lista 

Lista Stawka VAT w kontrolce Dane tabelaryczne

▪ **Jednostki miary:**

```
SELECT Naz_GIDLp,Naz_Nazwa from cdn.Nazwy where Naz_GIDTyp=144  
;
```

[Dane tabelaryczne] Elementy

Dane tabelaryczne

OGÓLNE

LISTY

INICJOWANIE WARTOŚCI

OGRANICZENIE EDYCJI

Lista 1

Nazwa

JM

Inicjowanie

Jednostki miary

Rodzaj identyfikatora

Int

Pokaż

Usuń

Tryb pracy

Lista

Dodaj

Lista Jednostka miary w kontrolce Dane tabelaryczne

4.Dodano obserwację kontrolki: W kontrolce typu **Dane tabelaryczne**, na zakładce **Inicjowanie wartości**, dodano powiązanie do kontrolki **ID_OCR**.

[Dane tabelaryczne] Elementy

[Etykieta] Podsumowanie

Dane tabelaryczne

OGÓLNE

LISTY

INICJOWANIE WARTOŚCI

OGRANICZENIE EDYCJI

Inicjowanie wartości

Inicjowanie

Brak



Kontrolki powiązane

ID_OCR (ID_OCR)



Dodaj

Wzór na wartość

SQL ERP



Ustaw

Powiązanie do kontrolki ID_OCR

5.Zapytanie SQL: W kontrolce typu **Dane tabelaryczne**, na zakładce **Inicjowanie wartości** w sekcji **Wzór na wartość SQL** OD dodano zapytanie, które pobiera dane z tabel **do.OC_Document_Items** oraz **do.OC_Documents** na podstawie przekazanego parametru **@ID_OCR@** (ID dokumentu OCR). W zapytaniu SQL konieczne jest użycie logiki CASE WHEN do konwersji wartości tekstowych **jednostek miary** ('szt') i **stawek VAT** (np. '23.00') na odpowiednie identyfikatory liczbowe akceptowane przez Comarch BPM.

```
If @^SenderControlName@ = 'ID_OCR'
```

```
Begin
```

Select

ODI_Name, /*Nazwa*/

ODI_NetUnitPrice, /*Cena netto*/

ODI_Count, /*Ilość*/

CASE

WHEN ODI_Unit='szt.' THEN 1

WHEN ODI_Unit='l' THEN 2

WHEN ODI_Unit='kg' THEN 3

Else 0

END, /*Jednostka miary*/

ODI_NetValue, /* Wartość Netto*/

CASE

WHEN ODI_VatRate='23.00' THEN 1

WHEN ODI_VatRate='8.00' THEN 2

WHEN ODI_VatRate='0.00' THEN 3

WHEN ODI_VatRate='5.00' THEN 7

Else 0

END,

ODI_GrossValue, /*Wartość Brutto*/

[ODI_GrossValue]-[ODI_NetValue], -- kwota VAT

1

```

from do.OC_Document_Items I

join do.OC_Documents D on D.OCD_Id=I.ODI_DocumentId

where ODI_DocumentId=@ID_OCR@

END

```

The screenshot displays the 'Wyrażenie kontrolki' (Control Expression) window. The left pane contains a SQL query that calculates VAT and net value based on document items. The right pane shows the 'Kontrolki' (Controls) configuration for 'Dane tabelaryczne' (Table Data). The configuration includes a search bar, a tab for 'ZAKŁADKA 1', and a section for 'Dane tabelaryczne' with options for 'OGÓLNE', 'LISTY', 'INICJOWANIE WARTOŚCI', and 'OGRANICZENIE EDYCJI'. The 'INICJOWANIE WARTOŚCI' (Initiate Values) section is active, showing a dropdown for 'Inicjowanie' (Initiation) set to 'Brak' (None). Below this, there are fields for 'Kontrolki powiązane' (Associated Controls) and 'Wzór na wartość' (Value Formula), both set to 'ID_OCR (ID_OCR)' and 'SQL OD' respectively.

Zapytanie SQL w kontrolce Dane tabelaryczne na zakładce Inicjowanie wartości

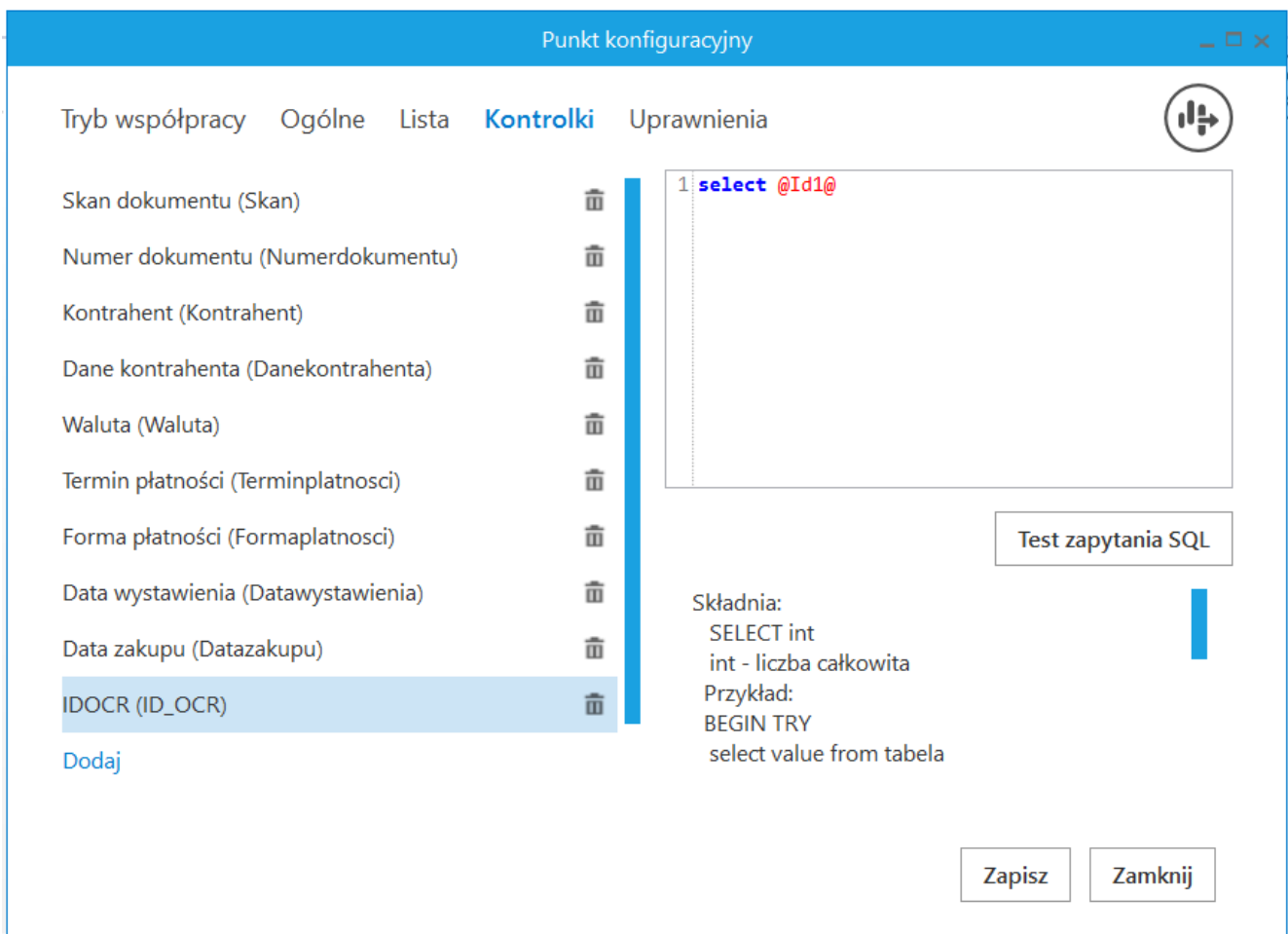
Uwaga

Należy upewnić się, że identyfikatory liczbowe (ID) **stawek VAT** i **jednostek miary** użyte w zapytaniu SQL są zgodne z tymi, które aktualnie obowiązują w bazie Comarch ERP XL współpracującej z Comarch BPM.

6.Punkt ACD: W punkcie konfiguracyjnym ACD (Współpraca z Comarch OCR), wykorzystywanym do inicjowania obiegu, na

zakładce **Kontrolki** dodano kontrolkę **ID_OCR** i przypisano jej odpowiednie zapytanie:

```
select @Id1@
```



Kontrolka ID_OCR w punkcie ACD

Uzupełniona Kontrolka Dane Tabelaryczne

Po pomyślnym zainicjowaniu dokumentu z punktu **ACD**, kontrolka typu **Dane Tabelaryczne** została automatycznie wypełniona. Dla każdej pozycji dokumentu (faktury) odczytanej przez Comarch OCR i zapisanej w tabeli **do.OC_Document_Items**, na formularzu BPM pojawił się **oddzielny, kompletny wiersz** z uzupełnionymi

danymi.

Pozycje dokumentu

Elementy

LP	Nazwa	Cena	Ilość	Jednostka miary	Netto	Stawka VAT	Brutto	Kwota VAT
1	TOWAR1	60,00	4,00	szt.	240,00	23%	295,20	55,20
2	TOWAR2	30,00	7,00	szt.	210,00	23%	258,30	48,30
3	TOWAR3	80,00	7,00	szt.	560,00	23%	688,80	128,80
4	TOWAR4	90,00	8,00	kg	720,00	8%	777,60	57,60

Podsumowanie

Wartość Netto

1 730,00

Wartość Brutto

2 019,90

Techniczne

ID_OCR

5

Kontrolka Dane tabelaryczne uzupełniona pozycjami odczytanym przez Comarch OCR

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

[Sprawdź Szkolenia Comarch BPM!](#)

[Powrót do początku artykułu](#)

Czy wiesz, że można wypełnić kontrolkę typu Dane tabelaryczne pozycjami dokumentu zaczytanego przez Comarch OCR oraz przeliczyć wartości przy współpracy z Comarch ERP XL?

Zastosowanie

Dzięki zastosowaniu tej zintegrowanej konfiguracji, proces obiegu faktur kosztowych w **Comarch BPM** zyskuje **pełną automatyzację i spójność danych**. Głównym celem jest:

- **Eliminacja ręcznego wprowadzania pozycji dokumentów:** Dane zaczytane przez **Comarch OCR** są automatycznie mapowane i wypełniają kontrolkę tabelaryczną na karcie obiegu.
- **Automatyczne kalkulowanie wartości:** Kontrolka tabelaryczna uzyskuje pełną zdolność automatycznego przeliczania wszędzie tam, gdzie zmienione zostaną dane wejściowe (np. Cena, Ilość, Stawka VAT, Brutto).
- **Minimalizacja błędów i skrócenie czasu obsługi faktury:** Proces akceptacji bazuje na poprawnych i spójnych danych, które są automatycznie kalkulowane i aktualizowane, zanim dokument zostanie zaksięgowany w **ERP XL**.

Zawartość przykładu

- **Definicja Typu Obiegu (Workflow):** Wzorcowy ścieżka obiegu dokumentu kosztowego z krokami akceptacji, konfiguracją kontrolki Dane tabelaryczne i wygenerowaniem dokumentu do Comarch ERPXL
- **Punkt konfiguracyjny ACD:** Wzorcowy konfiguracja, która jest odpowiedzialna za pobranie ID dokumentu OCR, Załącznika oraz wszystkich kluczowych danych nagłówkowych (NIP, Kontrahent, Dane kontrahenta, Data zakupu, Data wystawienia, Termin płatności, Forma płatności, Waluta) oraz inicjację dokumentu obiegowego.

[Kliknij tutaj, aby pobrać przykład](#)

Zasada działania

Mechanizm opiera się na **dwóch trybach działania** zintegrowanych w jednej kontrolce **Dane Tabelaryczne („Elementy”)**:

1. Pobranie pozycji z Comarch OCR (Wyzwalacz: ID_OCR):

- Po przetworzeniu dokumentu przez Comarch OCR, punkt ACD zapisuje **ID dokumentu OCR** do dedykowanej kontrolki typu Liczba całkowita o nazwie **ID_OCR**.
- Kontrolka tabelaryczna **„Elementy”** obserwuje to pole, reagując na jego wypełnienie.
- **Zapytanie SQL** (umieszczone na zakładce Inicjowanie Wartości) jest filtrowane za pomocą parametru **@ID_OCR@** i pobiera wiersze z tabeli **do.OC_Document_Items** w bazie BPM.
- Na podstawie zwróconych wierszy kontrolka **DT** jest

automatycznie wypełniana pozycjami faktury.

- W zapytaniu OCR używana jest logika CASE WHEN do konwersji wartości tekstowych (**Stawki VAT, Jednostki miary**) na **identyfikatory liczbowe**, wymagane przez BPM.

2. Automatyczne Przeliczanie (Wyzwalacz: Edycja wiersza DT):

- Kontrolka **DT** jest skonfigurowana tak, aby **obserwować samą siebie**.
 - Zapytanie SQL jest wyzwalane przy każdej zmianie wartości w kolumnach wejściowych (**Cena, Ilość lub Stawka VAT**).
 - **Przeliczanie W Przód**: Logika IF w kodzie SQL jest wyzwalana przy zmianie Ceny, Ilości lub Stawki VAT, obliczając **Netto** i **Brutto**.
 - **Przeliczanie W Tył**: Logika ELSE IF jest wyzwalana przy zmianie kolumny **Brutto**, przeliczając i aktualizując wartość **Ceny** lub **Netto** (przy zachowaniu stałej Ilości).
 - Zapytanie to wykorzystuje komendę **UPDATE** do bezpośredniego przeliczania i aktualizacji wartości w kolumnach wynikowych (**Netto, Brutto**)
- Osobna kontrolka **WartoscNetto** obserwuje kolumnę wynikową (**Netto**) i zlicza jej sumę za pomocą prostego zapytania SQL (select sum (Netto) from @Elementy@).
- Osobna kontrolka **WartoscBrutto** obserwuje kolumnę wynikową (**Brutto**) i zlicza jej sumę za pomocą prostego zapytania SQL (select sum (Brutto) from @Elementy@).

Konfiguracja przykładu

1. Utworzono kontrolkę **ID_OCR**: Na karcie obiegu dodano kontrolkę typu **Liczba całkowita** o nazwie (identyfikatorze) **ID_OCR**, do której będzie zapisywane ID OCR-owanego dokumentu.

Techniczne



[Liczba całkowita] ID_OCR

Liczba całkowita



Kontrolkę można uzupełnić liczbą z zakresu od -2 147 483 648 do 2 147 483 647

Nazwa wyświetlana

ID_OCR

Nazwa (identyfikator)

ID_OCR

Prezentuj na całej szerokości

☐

Ustaw kontrolkę na początku wiersza

☒

Pokaż na liście

☐

Kontrolka Liczba całkowita ID_OCR

2. Utworzono kontrolkę Tabelaryczną: Dodano kontrolkę typu **Dane tabelaryczne** o nazwie (identyfikatorze) **Elementy** i skonfigurowano w niej następujące kolumny:

- **"Nazwa"** (identyfikator kolumny: **Nazwa**) – typ **Tekst**
- **"Cena"** (identyfikator kolumny: **Cena**) – typ **Liczba stałoprzecinkowa**
- **"Ilość"** (identyfikator kolumny: **Ilosc**) – typ **Liczba stałoprzecinkowa**
- **"Jednostka miary"** (identyfikator kolumny: **JM**) – typ **Lista**
- **"Netto"** (identyfikator kolumny: **Netto**) – typ **Liczba stałoprzecinkowa**
- **"Stawka VAT"** (identyfikator kolumny: **StawkaVAT**) – typ **Lista**
- **"Brutto"** (identyfikator kolumny: **Brutto**) – typ **Liczba**

stałoprzecinkowa

Pozycje dokumentu

Elementy

LP	Nazwa	Cena	Ilość	Jednostka miary	Netto	Stawka VAT	Brutto
----	-------	------	-------	-----------------	-------	------------	--------



Kolumny w kontrolce Dane tabelaryczne

3.W kontrolce typu **Dane tabelaryczne** na zakładce **Listy** zdefiniowano zapytanie pobierające:

▪ Stawki VAT:

```
select Naz_gidLp as Id,
CASE
  WHEN Naz_Nazwa='A 23.00' THEN '23%'
  WHEN Naz_Nazwa='B 8.00' THEN '8%'
  WHEN Naz_Nazwa='C 0.00' THEN '0%'
  WHEN Naz_Nazwa='D 0.00' THEN 'ZW'
  WHEN Naz_Nazwa='E 0.00' THEN 'NP'
  WHEN Naz_Nazwa='F 7.00' THEN '7%'
  WHEN Naz_Nazwa='G 5.00' THEN '5%'
END
from CDN.Nazwy
where Naz_GIDTyp = 624 AND trim(substring(Naz_Nazwa,9,2))= ''
```

Dane tabelaryczne

OGÓLNE

LISTY

INICJOWANIE WARTOŚCI

OGRANICZENIE EDYCJI

Lista 1



Nazwa

Stawka VAT

Inicjowanie

SQL ERP



Rodzaj identyfikatora

Int



[Pokaż](#) [Usuń](#)

Tryb pracy

Lista



Lista Stawka VAT w kontrolce Dane tabelaryczne

▪ Jednostki miary:

```
SELECT Naz_GIDLp,Naz_Nazwa from cdn.Nazwy where Naz_GIDTyp=144  
;
```

[Dane tabelaryczne] Elementy

Dane tabelaryczne

OGÓLNE

LISTY

INICJOWANIE WARTOŚCI

OGRANICZENIE EDYCJI

Lista 1

Nazwa

JM

Inicjowanie

Jednostki miary

Rodzaj identyfikatora

Int

Pokaż

Usuń

Tryb pracy

Lista

Dodaj

Lista Jednostka miary w kontrolce Dane tabelaryczne

4.Dodano obserwację kontrolki: W kontrolce **Dane tabelaryczne**, na zakładce **Inicjowanie wartości**, dodano powiązanie do kontrolki **ID_OCR** oraz **do samej siebie**.

[Dane tabelaryczne] Elementy

Dane tabelaryczne

OGÓLNE LISTY INICJOWANIE WARTOŚCI OGRANICZENIE EDYCJI

Inicjowanie wartości

Inicjowanie

Brak

Kontrolki powiązane

ID_OCR (ID_OCR)

Elementy (Elementy)

[Dodaj](#)

Wzór na wartość

SQL OD

[Ustaw](#)

Powiązanie do kontrolki ID_OCR i samej siebie

5. Zintegrowane zapytanie SQL do obsługi obu trybów: W kontrolce typu **Dane tabelaryczne**, na zakładce **Inicjowanie wartości** w sekcji **Wzór na wartość SQL ERP** dodano zapytanie, które pobiera dane z tabel **do.OC_Document_Items** oraz **do.OC_Documents** na podstawie przekazanego parametru **@ID_OCR@** (ID dokumentu OCR). W zapytaniu SQL konieczne jest użycie logiki CASE WHEN do konwersji wartości tekstowych **jednostek miary** ('szt') i **stawek VAT** (np. '23.00') na odpowiednie identyfikatory liczbowe akceptowane przez Comarch BPM. Dodano również zapytanie, które na podstawie **Iłości** i **Ceny** wylicza **wartość Netto**, po zmianie **Stawki VAT** wylicza **wartość Brutto**, a po zmianie wartości **Brutto** wylicza **Cenę** i **wartość Netto**.

Uwaga

Należy upewnić się, że identyfikatory liczbowe (ID) **stawek VAT** i **jednostek miary** użyte w zapytaniu SQL są zgodne z tymi, które aktualnie obowiązują w bazie Comarch ERP XL współpracującej z Comarch BPM.

Uwaga

W zapytaniu przed nazwą tabel **do.OC_Document_Items** oraz **do.OC_Documents** należy wskazać właściwą nazwę bazy **Comarch BPM**.

Ta sekcja jest wywoływana tylko raz, podczas inicjowania obiegu z OCR:

```
If @^SenderControlName@ = 'ID_OCR'
```

```
Begin
```

```
Select
```

```
ODI_Name, /*Nazwa*/
```

```
ODI_NetUnitPrice, /*Cena netto*/
```

```
ODI_Count, /*Ilość*/
```

```
CASE
```

```
WHEN ODI_Unit='szt.' THEN 1
```

```
WHEN ODI_Unit='l' THEN 2
```

```
WHEN ODI_Unit='kg' THEN 3
```

```
Else 0
```

```
END, /*Jednostka miary*/
```

```
ODI_NetValue, /* Wartość Netto*/
```

CASE

WHEN ODI_VatRate='23.00' THEN 1

WHEN ODI_VatRate='8.00' THEN 2

WHEN ODI_VatRate='0.00' THEN 3

WHEN ODI_VatRate='5.00' THEN 7

Else 0

END,

ODI_GrossValue, /*Wartość Brutto*/

[ODI_GrossValue]-[ODI_NetValue], -- kwota VAT

1

from [BPM_Przykłady].do.OC_Document_Items I /*Zmienić na właściwą nazwę bazy BPM*/

join [BPM_Przykłady].do.OC_Documents D on
D.OCD_Id=I.ODI_DocumentId /*Zmienić na właściwą nazwę bazy BPM*/

where ODI_DocumentId=@ID_OCR@

END

Ta sekcja jest wywoływana przy każdej modyfikacji danych w kontrolce „Elementy” i wymaga użycia ELSE po sekcji OCR:

ELSE

If @^InitSenderControlName@ ='Elementy'

Begin

/*Deklaracja Stawki VAT */

```
DECLARE @VAT AS DECIMAL(4, 2)
```

```
/* Pobranie stawki VAT z tabeli CDN.Nazwy na podstawie kolumny  
StawkaVAT w bieżącym wierszu */
```

```
SELECT @VAT = CAST(SUBSTRING(Naz_Nazwa1, 0, CHARINDEX(''  
'', Naz_Nazwa1)) AS DECIMAL(4, 2)) / 100
```

```
FROM CDN.Nazwy
```

```
WHERE Naz_GIDTyp = 624
```

```
AND Naz_GIDLp = (
```

```
    SELECT [StawkaVAT]
```

```
    FROM @Elementy@
```

```
    WHERE POS = @Elementy_Row@
```

```
)
```

```
/* JEŚLI ZMIENIONO KOLUMNY: 2 (Cena), 3 (Ilość), 6 (Stawka  
VAT) */
```

```
IF @Elementy_Column@=2 OR @Elementy_Column@=3 OR @Elementy_Col  
umn@=6
```

```
BEGIN
```

```
    /* PRZELICZANIE W PRZÓD (Cena/Ilość/Stawka VAT ->  
Netto/Brutto) */
```

```
    /*Wyliczenie i aktualizacja netto i brutto */
```

```
    UPDATE @Elementy@
```

```
    SET
```

```
        [Netto] = (Ilosc * Cena),
```

```
        [Brutto] = (Ilosc * Cena) * (@VAT + 1)
```

```
    WHERE POS = @Elementy_Row@
```

```
    /* !!! OBOWIĄZKOWE: Odświeżenie wiersza !!! */
```

```
    SELECT * FROM @Elementy@
```

```
END
```

```
/* JEŚLI ZMIENIONO KOLUMNĘ: 7 (Brutto) */
```

```
IF @Elementy_Column@=7
```

```
BEGIN
```

```
    /* PRZELICZANIE W TYŁ (Brutto -> Cena/Netto) */
```

```
    /* Aktualizacja ceny i netto (Ilość stała) */
```

```
    UPDATE @Elementy@
```

```
    SET
```

```

        [Cena] = [Brutto] / (@VAT + 1),          /*Obliczenie
Ceny Netto z Brutto*/
        [Netto] = [Ilosc] * ([Brutto] / (@VAT + 1)) /*Oblicze
nie Wartości Netto*/
WHERE POS = @Elementy_Row@

/* !!! OBOWIĄZKOWE: Odświeżenie wiersza !!! */
SELECT * FROM @Elementy@

END
END

```

The image shows two parts of a software interface. On the left is the 'Wyrażenie kontrolki' (Control Expression) dialog box, which contains a SQL query for calculating net price. On the right is the 'Kontrolki' (Controls) panel, showing the 'Dane tabelaryczne' (Table Data) tab and the 'Inicjowanie wartości' (Value Initialization) section.

Wyrażenie kontrolki (Control Expression):

```

Uwaga! Zmiana nazwy kontrolki może spowodować nieprawidłowe działanie wyrażenia.

1 If @*SenderControlName@ = 'ID_OCR'
2 Begin
3 /*Mapowanie danych z OCR*/
4 Select
5 ODI_Name, /*Nazwa*/
6 ODI_NetUnitPrice, /*Cena netto*/
7 ODI_Count, /*Ilość*/
8 CASE
9 WHEN ODI_Unit='szt.' THEN 1
10 WHEN ODI_Unit='l' THEN 2
11 WHEN ODI_Unit='kg' THEN 3
12 Else 0
13 END, /*Jednostka miary*/
14
15 ODI_NetValue, /*Wartość Netto*/
16 CASE
17 WHEN ODI_VatRate='23.00' THEN 1
18 WHEN ODI_VatRate='8.00' THEN 2
19 WHEN ODI_VatRate='0.00' THEN 3
20 WHEN ODI_VatRate='5.00' THEN 7
21 Else 0
22 END,
23 ODI_GrossValue, /*Wartość Brutto*/
24 1 /*LP wiersza*/
25 From [BPM.Przyklady].do_OC_Document_Items I /*Zmienić na właściwą nazwę bazy BPM*/
26 Join [BPM.Przyklady].do_OC_Documents D on D.ODC_Id=I.OOI_DocumentId /*Zmienić na właściwą nazwę bazy BPM*/
27 Where ODI_DocumentId=@ID_OCR@
28 end
29
30 ELSE
31 If @*InitSenderControlName@ = 'Elementy'
32 Begin
33 /*Deklaracja Stawki VAT */
34 DECLARE @VAT AS DECIMAL(4, 2)
35
36 /* Pobranie stawki VAT z tabeli CDN.Nazwy na podstawie kolumny StawkaVAT w bieżącym wierszu */
37 SELECT @VAT = CAST(SUBSTRING(Naz_Nazwa1, 0, CHARINDEX(' ', Naz_Nazwa1)) AS DECIMAL(4, 2)) / 100
38 FROM CDN.Nazwy

```

Kontrolki (Controls) Panel:

- ZAKŁADKA 1** (TAB 1)
- Pozycje dokumentu** (Document Positions)
- [Dane tabelaryczne] Elementy** (Table Data Elements)
- Dane tabelaryczne** (Table Data)
- OGÓLNE** (General)
- LISTY** (Lists)
- INICJOWANIE WARTOŚCI** (Value Initialization)
- OGRANICZENIE EDYCJI** (Edit Restriction)
- Inicjowanie wartości** (Value Initialization)
- Inicjowanie** (Initialization): Brak (None)
- Kontrolki powiązane** (Related Controls): ID_OCR (ID_OCR), Elementy (Elementy)
- Wzór na wartość** (Value Formula): SQL ERP
- Ustaw** (Set)

Zapytanie SQL w kontrolce Dane tabelaryczne na zakładce Inicjowanie wartości

- **Tryb pracy:** W kontrolce typu **Dane tabelaryczne**, na zakładce **Ograniczenie edycji** kolumnę „Netto” ustawiono w trybie „Do odczytu”.

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 Nazwa Nazwa	☐ Zaznacz wszystko Cena Cena	☐ Zaznacz wszystko Ilość Ilość	☐ Zaznacz wszystko JM Jednostka miary	☒ Zaznacz wszystko Netto Netto	☐ Zaznacz wszystko StawkaVAT Stawka VAT	☐ Zaznacz wszystko Brutto Brutto
Rejestracja dokumentu	☐	☐	☐	☐	☐	☐	☒	☐	☐
Dekretacja kosztów	☐	☐	☐	☐	☐	☐	☒	☐	☐
Akceptacja kierownika	☐	☐	☐	☐	☐	☐	☒	☐	☐
Księgowość	☐	☐	☐	☐	☐	☐	☒	☐	☐
Odrzucono	☐	☐	☐	☐	☐	☐	☒	☐	☐
Archiwum	☐	☐	☐	☐	☐	☐	☒	☐	☐

Kolumna Netto w trybie „Do odczytu”

- **Utworzono kontrolkę liczbową:** Dodano kontrolkę typu **Liczba stałoprzecinkowa** o nazwie (identyfikatorze) **WartoscNetto**, w której dodano powiązanie do kontrolki **Elementy** i skonfigurowano w niej zapytanie, które zlicza wartość z kolumny **Netto** z kontrolki tabelarycznej:

```
select sum (Netto) from @Elementy@
```

Wyrażenie kontrolki

Uwaga!
Zmiana nazwy kontrolki może spowodować nieprawidłowe działanie wyrażenia.

```
1 SELECT SUM(Netto) FROM @Elementy@
```

Kontrolki

^Document
^UserId
^DocumentI
^Workflow
^Stageld

Składnia:

SELECT decimal
(22,4)
decimal – wartość
stałoprzecinkowa.
Przykład:
BEGIN TRY
SELECT 1000.2588
END TRY
BEGIN CATCH
SELECT 0

Zapisz

Zamknij

[Liczba stałoprzecinkowa] WartoscNetto

Liczba stałoprzecinkowa

Kontrolkę można uzupełnić liczbą złożoną z maksymalnie 18 cyfr przed przecinkiem i 6 miejsc po przecinku

Nazwa wyświetlana

WartoscNetto

Nazwa (identyfikator)

WartoscNetto

Liczba miejsc po przecinku, do której będzie zaokrąglana wartość

2

Prezentuj na całej szerokości

☐

Ustaw kontrolkę na początku wiersza

☐

Pokaż na liście

☒

Inicjowanie wartości

Inicjowanie

Brak

Dokument

Pole

Zmiana wartości (Obserwator)

Kontrolki powiązane

Elementy (Elementy)

Dodaj

Usuń

Wzór na wartość

SQL OD

Kontrolka Liczbowa WartoscNetto

- **Utworzono kontrolkę liczbową:** Dodano kontrolkę typu **Liczba stałoprzecinkowa** o nazwie (identyfikatorze) **WartoscBrutto**, w której dodano powiązanie do kontrolki **Elementy** i skonfigurowano w niej zapytanie, które zlicza wartość z kolumny Brutto z kontrolki tabelarycznej:

```
select sum (Brutto) from @Elementy@
```

Wyrażenie kontrolki

Uwaga!
Zmiana nazwy kontrolki może spowodować nieprawidłowe działanie wyrażenia.

```
1 SELECT SUM(Brutto) FROM @Elementy@
```

Kontrolki

^Document
^UserId
^DocumentI
^Workflowc
^Stageld

Składnia:

SELECT decimal
(22,4)
decimal – wartość
stałoprzecinkowa.
Przykład:
BEGIN TRY
SELECT 1000.2588
END TRY
BEGIN CATCH
SELECT 0

Zapisz
Zamknij

[Liczba stałoprzecinkowa] WartoscBrutto

Liczba stałoprzecinkowa

i Kontrolkę można uzupełnić liczbą złożoną z maksymalnie 18 cyfr przed przecinkiem i 6 miejsc po przecinku

Nazwa wyświetlanaWartoscBrutto

Nazwa (identyfikator)WartoscBrutto

Liczba miejsc po przecinku, do której będzie zaokrąglana wartość2

Prezentuj na całej szerokości

Ustaw kontrolkę na początku wiersza

Pokaż na liście

Inicjowanie wartości

InicjowanieBrak

Dokument

Pole

Zmiana wartości (Obserwator)

Kontrolki powiązaneElementy (Elementy)

Dodaj

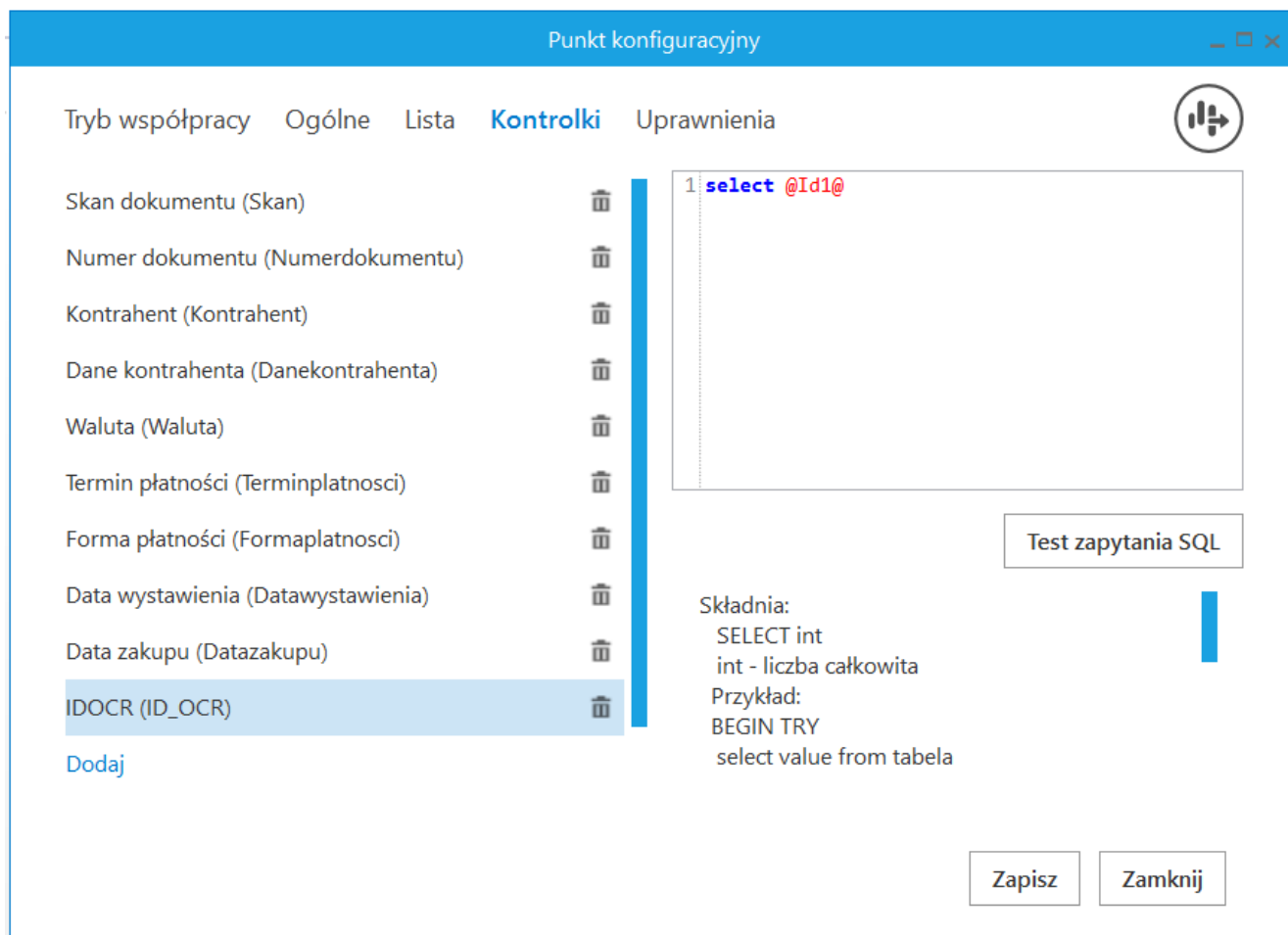
Wzór na wartośćSQL OD

PokażUsuń

Kontrolka Liczbowa WartoscBrutto

- **Punkt ACD:** W punkcie konfiguracyjnym ACD (Współpraca z Comarch OCR), wykorzystywanym do inicjowania obiegu, na zakładce **Kontrolki** dodano kontrolkę **ID_OCR** i przypisano jej odpowiednie zapytanie:

```
select @Id1@
```



Kontrolka ID_OCR w punkcie ACD

Uzupełniona Kontrolka Dane Tabelaryczne

Po pomyślnym zainicjowaniu dokumentu z punktu ACD, kontrolka typu **Dane Tabelaryczne** została automatycznie wypełniona **pozycjami odczytanymi** przez **OCR**. Dla każdej pozycji, system automatycznie przeliczył i uzupełnił wartości w kolumnach **Netto** i **Brutto** w tym samym wierszu. W przypadku nieprawidłowości w kwotach możliwe jest przeliczanie wartości **w przód** (**Cena/Ilość** -> **Netto/Brutto**) oraz **w tył** (**Brutto** -> **Cena/Netto**). Kolumna **Netto** jest zablokowana do ręcznej edycji (tryb „Do odczytu”), co gwarantuje poprawność danych. Kontrolka **WartoscNetto** wyświetla poprawną sumę końcową z kolumny **Netto** kontrolki Dane tabelaryczne. Kontrolka **WartoscBrutto** wyświetla poprawną sumę końcową z kolumny **Brutto**

kontrolki typu Dane tabelaryczne.

Pozycje dokumentu

Elementy

LP	Nazwa	Cena	Ilość	Jednostka miary	Netto	Stawka VAT	Brutto
1	Towar1	80,00	4,00	szt.	320,00	23%	393,60
2	Towar2	80,00	8,00	szt.	640,00	23%	787,20
3	Towar3	80,00	7,00	szt.	560,00	23%	688,80
4	Towar4	90,00	8,00	szt.	720,00	8%	777,60

Podsumowanie

WartoscNetto

2 240,00

WartoscBrutto

2 647,20

Techniczne

ID_OCR

5

Kontrolka Dane tabelaryczne uzupełniona pozycjami odczytanymi przez Comarch OCR oraz podsumowanie kwot

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

[Sprawdź Szkolenia Comarch BPM!](#)

[Powrót do początku artykułu](#)

Czy wiesz, że można wypełnić kontrolkę typu Dane Tabełaryczne pozycjami dokumentu pobranego z KSeF oraz przeliczyć wartości przy współpracy z Comarch ERP XL?

Zastosowanie

Dzięki zastosowaniu tej zintegrowanej konfiguracji, proces obiegu faktur kosztowych w **Comarch BPM** zyskuje **pełną automatyzację i spójność danych**. Dane są pobierane bezpośrednio z systemu KSeF (poprzez BPM), a kontrolka tabelaryczna jest automatycznie wypełniana.

Głównym celem jest:

- **Eliminacja ręcznego wprowadzania pozycji dokumentów:** Dane z faktury KSeF (pobrane do tabeli `do.KS_DocumentItems`) są automatycznie mapowane i wypełniają kontrolkę tabelaryczną na karcie obiegu.
- **Automatyczne kalkulowanie wartości:** Kontrolka tabelaryczna uzyskuje zdolność automatycznego przeliczania wszędzie tam, gdzie zmienione zostaną dane wejściowe (np. Cena, Ilość, Stawka VAT) przez użytkownika.
- **Minimalizacja błędów i skrócenie czasu obsługi faktury:** Proces akceptacji bazuje na poprawnych i spójnych

danych, które są automatycznie kalkulowane i aktualizowane, zanim dokument zostanie zaksięgowany w ERP XL.

Zawartość przykładu

- **Definicja Typu Obiegu (Workflow):** Wzorcową ścieżką obiegu dokumentu kosztowego z krokami akceptacji, konfiguracją kontrolki Dane tabelaryczne i wygenerowaniem dokumentu do Comarch ERPXL
- **Punkt konfiguracyjny ACD:** Wzorcową konfiguracją, która jest odpowiedzialna za pobranie ID dokumentu KSeF, Załącznika oraz wszystkich kluczowych danych nagłówkowych (Numer dokumentu, Kontrahent, Data zakupu, Data wystawienia, Data wpływu Termin płatności, Forma płatności, Waluta, Wartość Netto, Wartość Brutto, Numer KSeF) oraz inicjację dokumentu obiegowego. W przypadku kontrolki **Kontrahent** w zapytaniu przed nazwą tabeli CDN.KntKarty należy wskazać właściwą nazwę bazy **Comarch ERXL**.

[Kliknij tutaj, aby pobrać przykład](#)

Zasada działania

Mechanizm opiera się na **dwóch trybach działania** zintegrowanych w jednej kontrolce **Dane Tabelaryczne (Elementy)**, wykorzystując dane z bazy Comarch BPM, które zostały zapisane po przetworzeniu faktury z **KSeF** w punkcie **ACD**.

- Konfiguracja wykorzystuje **ID dokumentu KSeF** (zapisane w dedykowanej kontrolce typu Liczba całkowita o nazwie np. **ID_KSeF**) jako identyfikator do pobrania danych.

- Kontrolka tabelaryczna obserwuje to pole (**ID_KSeF**), reagując na jego wypełnienie.
- Kluczowym elementem jest **zapytanie SQL** umieszczone na zakładce **Inicjowanie Wartości** kontrolki typu dane tabelarycznej.
- Zapytanie jest filtrowane za pomocą parametru **@ID_KSeF@**, dzięki czemu pobiera ono wiersze z właściwego dokumentu **KSeF** z tabeli **do.KS_DocumentItems** w bazie BPM.
- W zapytaniu SQL zawarta jest niezbędna **logika konwersji** (CTE z ROW_NUMBER), która mapuje stawki VAT z KSeF na identyfikatory liczbowe (**Naz_GIDLp**) z tabeli CDN.Nazwy, których BPM wymaga do poprawnego wyświetlania danych w kolumnie typu **Lista (Stawka VAT)**.
- Dodatkowo, kontrolka **Dane Tabelaryczne (Elementy)** obserwuje sama siebie (**@^SenderControlName@ = 'Elementy'**), co aktywuje logikę automatycznego przeliczania wartości w trybie edycji:
- Zapytanie monitoruje zmiany w kolumnach wejściowych, takich jak **Cena** lub **Ilość** (oraz **Stawka VAT**).
- W przypadku edycji tych pól, skrypt automatycznie pobiera **stawkę VAT**, a następnie przelicza i aktualizuje wartości **Netto (Ilość * Cena)** oraz **Brutto (Netto * (1 + VAT))** w bieżącym wierszu.

Konfiguracja przykładu

1. **Utworzono kontrolkę ID_KSeF:** Na karcie obiegu dodano kontrolkę typu **Liczba całkowita** o nazwie (identyfikatorze) **ID_KSeF**, do której będzie zapisywane **ID dokumentu KSeF**.

Liczba całkowita

 Kontrolkę można uzupełnić liczbą z zakresu od -2 147 483 648 do 2 147 483 647

Nazwa wyświetlana ID_KSeF

Nazwa (identyfikator) ID_KSeF

Prezentuj na całej szerokości ☐

Ustaw kontrolkę na początku wiersza ☐

Pokaż na liście ☐

Inicjowanie wartości

Inicjowanie Brak

Dokument

Pole

Zmiana wartości (Obserwator)

Kontrolki powiązane [Dodaj](#)

Wzór na wartość Wyrażenie

[Ustaw](#)

Kontrolka Liczba całkowita ID_KSeF

2. **Utworzono kontrolkę Tabelaryczną:** Dodano kontrolkę typu **Dane tabelaryczne** o nazwie (identyfikatorze) **Elementy** i skonfigurowano w niej następujące kolumny:

- **"Nazwa"** (identyfikator kolumny: **Nazwa**) – typ **Tekst**
- **"Cena"** (identyfikator kolumny: **Cena**) – typ **Liczba stałoprzecinkowa**
- **"Ilość"** (identyfikator kolumny: **Ilosc**) – typ **Liczba stałoprzecinkowa**
- **"Netto"** (identyfikator kolumny: **Netto**) – typ **Liczba stałoprzecinkowa**
- **"Stawka VAT"** (identyfikator kolumny: **StawkaVAT**) – typ

Lista

- **"Brutto"** (identyfikator kolumny: **Brutto**)– typ **Liczba stałoprzecinkowa**

Pozycje dokumentu

Elementy

LP	Nazwa	Cena	Ilość	Netto	Stawka VAT	Brutto
----	-------	------	-------	-------	------------	--------



Kolumny w kontrolce typu Dane tabelaryczne

3. W kontrolce typu **Dane tabelaryczne** o nazwie (identyfikatorze) **Elementy** na zakładce **Listy** zdefiniowano zapytanie pobierające:

▪ Stawki VAT:

```
select Naz_gidLp as Id,  
CASE  
  WHEN Naz_Nazwa='A 23.00' THEN '23%'  
  WHEN Naz_Nazwa='B 8.00' THEN '8%'  
  WHEN Naz_Nazwa='C 0.00' THEN '0%'  
  WHEN Naz_Nazwa='D 0.00' THEN 'ZW'  
  WHEN Naz_Nazwa='E 0.00' THEN 'NP'  
  WHEN Naz_Nazwa='F 7.00' THEN '7%'  
  WHEN Naz_Nazwa='G 5.00' THEN '5%'  
END  
from CDN.Nazwy  
where Naz_GIDTyp = 624 AND trim(substring(Naz_Nazwa,9,2))=''
```

[Dane tabelaryczne] Elementy

Dane tabelaryczne

OGÓLNE

LISTY

INICJOWANIE WARTOŚCI

OGRANICZENIE EDYCJI

Lista 1



Nazwa

Stawka VAT

Inicjowanie

SQL ERP



Rodzaj identyfikatora

Int



[Pokaż](#) [Usuń](#)

Tryb pracy

Lista



Lista Stawka VAT w kontrolce typu Dane tabelaryczne

▪ Jednostki miary:

```
SELECT Naz_GIDLp,Naz_Nazwa from cdn.Nazwy where Naz_GIDTyp=144  
;
```

[Dane tabelaryczne] Elementy

Dane tabelaryczne

OGÓLNE

LISTY

INICJOWANIE WARTOŚCI

OGRANICZENIE EDYCJI

Lista 1

Nazwa

JM

Inicjowanie

Jednostki miary

Rodzaj identyfikatora

Int

Pokaż

Usuń

Tryb pracy

Lista

Dodaj

Lista Jednostka miary w kontrolce typu Dane tabelaryczne

4. **Dodano obserwację kontrolki:** W kontrolce **Dane tabelaryczne**, na zakładce **Inicjowanie wartości**, dodano powiązanie do kontrolki **ID_KSeF** oraz **do samej siebie**.

[Dane tabelaryczne] Elementy

Dane tabelaryczne

OGÓLNE LISTY INICJOWANIE WARTOŚCI OGRANICZENIE EDYCJI

Inicjowanie wartości

Inicjowanie

Brak

Kontrolki powiązane

Elementy (Elementy)

ID_KSeF (ID_KSeF)

Dodaj

Wzór na wartość

SQL ERP

Ustaw

Dane tabelaryczna, zakładka *Inicjowanie wartości*

5. **Dodano zapytanie SQL:** W kontrolce **Dane tabelaryczne**, na zakładce **Inicjowanie wartości** w sekcji **Wzór na wartość** **SQL ERP** dodano zapytanie, które pobiera dane z tabel **do.KS_DocumentItems** na podstawie przekazanego parametru **@ID_KSeF@** (ID dokumentu KSeF).

W zapytaniu SQL konieczne jest użycie logiki **CASE WHEN** do konwersji wartości pola **KSI_ElementStawkaVAT** (np. 'Stawka23', 'zw') na ujednolicony klucz, który następnie jest mapowany na **identyfikatory liczbowe** (Naz_GIDLp) stawek VAT z tabeli **CDN.Nazwy** z bazy XL. Logika **CTE** z **ROW_NUMBER()** zapewnia unikalność mapowanych ID.

Uwaga

W zapytaniu przed nazwą tabeli **do.KS_DocumentItems** należy

wskazać właściwą nazwę bazy **Comarch BPM**.

- **Ta sekcja jest wywoływana tylko raz, podczas inicjowania obiegu:**

```
IF @^SenderControlName@ = 'ID_KSeF' /*Identyfikator kontrolki  
do której pobierane jest ID dokumentu KSeF */
```

```
BEGIN
```

```
/* Tryb 1: Inicjowanie wierszy z dokumentu KSeF */
```

```
WITH ProcessedDocumentItems AS (
```

```
/* CTE 1: Ujednolicenie stawek VAT, tabela  
KSI_ElementStawkaVAT */
```

```
SELECT
```

```
KSI_ElementNazwa,
```

```
KSI_ElementCenaNetto,
```

```
KSI_ElementIlosc,
```

```
KSI_ElementWartoscNetto,
```

```
KSI_ElementStawkaVAT,
```

```
CASE
```

```
WHEN di.KSI_ElementStawkaVAT LIKE 'Stawka%' THEN  
LTRIM(RTRIM(SUBSTRING(di.KSI_ElementStawkaVAT, 7,  
LEN(di.KSI_ElementStawkaVAT) - 6)))
```

```
WHEN di.KSI_ElementStawkaVAT = 'zw' THEN 'ZW'
```

```

WHEN di.KSI_ElementStawkaVAT = 'np' THEN 'NP'

ELSE di.KSI_ElementStawkaVAT

END AS StandardizedStawkaVAT

FROM

[BAZA_BPM].do.KS_DocumentItems AS di /*Należy wskazać nazwę
bazy BPM*/

WHERE KSI_DocumentId = @ID_KSeF@ /*Identyfikator kontrolki
liczba całkowita, do której pobierane jest ID dokumentu KSeF*/

),

ProcessedNazwyRanked AS (

/* CTE 2: Porządkowanie stawek VAT z tabeli słownikowej
(CDN.Nazwy), w celu zapewnienia unikalności */

SELECT

Naz_GIDLp,

Naz_Nazwa1,

CASE

WHEN LTRIM(RTRIM(n.Naz_Nazwa1)) LIKE '%ZW%' THEN 'ZW'

WHEN LTRIM(RTRIM(n.Naz_Nazwa1)) LIKE '%NP%' THEN 'NP'

WHEN CHARINDEX(' ', LTRIM(RTRIM(n.Naz_Nazwa1))) > 0 THEN
LTRIM(RTRIM(SUBSTRING(n.Naz_Nazwa1, 1, CHARINDEX(' ',
n.Naz_Nazwa1) - 1)))

ELSE LTRIM(RTRIM(n.Naz_Nazwa1))

END AS StandardizedNazwa1_ForJoin,

```

CASE

WHEN LTRIM(RTRIM(n.Naz_Nazwa1)) LIKE '%ZW%' OR
LTRIM(RTRIM(n.Naz_Nazwa1)) LIKE '%NP%' THEN 0.00

WHEN ISNUMERIC(

CASE

WHEN CHARINDEX(' ', LTRIM(RTRIM(n.Naz_Nazwa1))) > 0 THEN
LTRIM(RTRIM(SUBSTRING(n.Naz_Nazwa1, 1, CHARINDEX(' ',
n.Naz_Nazwa1) - 1)))

ELSE LTRIM(RTRIM(n.Naz_Nazwa1))

END

) = 1 THEN CAST(

CASE

WHEN CHARINDEX(' ', LTRIM(RTRIM(n.Naz_Nazwa1))) > 0 THEN
LTRIM(RTRIM(SUBSTRING(n.Naz_Nazwa1, 1, CHARINDEX(' ',
n.Naz_Nazwa1) - 1)))

ELSE LTRIM(RTRIM(n.Naz_Nazwa1))

END

AS DECIMAL(5, 2)) / 100.0

ELSE 0.00

END AS CalculatedVATRate,

/* Numerujemy rekordy dla tego samego klucza łączenia. */

ROW_NUMBER() OVER (PARTITION BY

CASE

```

WHEN LTRIM(RTRIM(n.Naz_Nazwa1)) LIKE '%ZW%' THEN 'ZW'

WHEN LTRIM(RTRIM(n.Naz_Nazwa1)) LIKE '%NP%' THEN 'NP'

WHEN CHARINDEX(' ', LTRIM(RTRIM(n.Naz_Nazwa1))) > 0 THEN
LTRIM(RTRIM(SUBSTRING(n.Naz_Nazwa1, 1, CHARINDEX(' ',
n.Naz_Nazwa1) - 1)))

ELSE LTRIM(RTRIM(n.Naz_Nazwa1))

END

ORDER BY n.Naz_GIDLp ASC) as RN

FROM

CDN.Nazwy AS n

WHERE Naz_GIDTyp = 624

),

ProcessedNazwy AS (

/* CTE 3: Finalne, unikalne stawki */

SELECT

Naz_GIDLp,

StandardizedNazwa1_ForJoin,

CalculatedVATRate

FROM

ProcessedNazwyRanked

WHERE RN = 1 /* Wybieramy tylko jeden (pierwszy) rekord dla
każdej unikalnej stawki */

```

```

)

/* Końcowy SELECT: Zwraca żądany zestaw kolumn */

SELECT

pi.KSI_ElementNazwa,

pi.KSI_ElementCenaNetto,

pi.KSI_ElementIlosc,

pi.KSI_ElementWartoscNetto,

pn.Naz_GIDLp AS 'stawka VAT', /* ID stawki z drugiej tabeli */

pi.KSI_ElementWartoscNetto * (1 + pn.CalculatedVATRate) AS
'Wartość brutto',

NULL AS 'NULL'

FROM

ProcessedDocumentItems AS pi

JOIN

ProcessedNazwy AS pn

ON

pi.StandardizedStawkaVAT = pn.StandardizedNazwa1_ForJoin;

END

```

- Ta sekcja jest wywoływana przy każdej modyfikacji danych w kontrolce „Elementy” i wymaga użycia ELSE po sekcji KSeF:

ELSE

IF @^SenderControlName@ = 'Elementy' /*Identyfikator kontrolki
Dane tabelaryczne*/

BEGIN

/* Tryb 2: Automatyczne przeliczanie wartości w trybie edycji
*/

/* Kolumny 2-Cena, 3-Ilość, 5-Stawka VAT */

IF @Elementy_Column@=2 OR @Elementy_Column@=3 OR
@Elementy_Column@=5

BEGIN

DECLARE @VAT as decimal (4,2)

/* Pobranie stawki VAT z tabeli słownikowej */

SELECT @VAT=CAST(SUBSTRING (Naz_Nazwa1, 0, CHARINDEX (' ',
Naz_Nazwa1)) as decimal (4,2))/100

FROM CDN.Nazwy

WHERE Naz_GIDTyp=624 AND Naz_GIDLp=

(SELECT [Stawka VAT]

FROM @Elementy@

WHERE POS=@Elementy_Row@)

UPDATE @Elementy@

/* Aktualizacja wartości Netto */

SET [Netto] = (Ilosc * Cena),

```
/* Aktualizacja wartości Brutto */
```

```
[Brutto] = (Ilosc * Cena) * (@VAT + 1)
```

```
WHERE POS=@Elementy_Row@
```

```
/* Zwrócenie zaktualizowanej tabeli - Obowiązkowy select */
```

```
SELECT * FROM @Elementy@
```

```
END
```

```
END
```

Wyrażenie kontrolki

Uwaga! Zmiana nazwy kontrolki może spowodować nieprawidłowe działanie wyrażenia.

```
1 IF @SenderControlName = 'ID_KSeF' /*Identyfikator kontrolki do której pobierane jest ID dokumentu KSeF */
2 BEGIN
3     /* Tryb 1: Inicjowanie wierszy z dokumentu KSeF */
4
5     WITH ProcessedDocumentItems AS (
6         /* CTE 1: Ujednolicenie stawek VAT, tabela KSI_ElementStawkaVAT */
7         SELECT
8             KSI_ElementNazwa,
9             KSI_ElementCenaNetto,
10            KSI_ElementIlosc,
11            KSI_ElementWartoscNetto,
12            KSI_ElementStawkaVAT,
13            CASE
14                WHEN di.KSI_ElementStawkaVAT LIKE 'Stawka%' THEN LTRIM(RTRIM(SUBSTRING(di.KSI_ElementStawkaVAT, 11, 10)))
15                WHEN di.KSI_ElementStawkaVAT = 'zw' THEN 'ZW'
16                WHEN di.KSI_ElementStawkaVAT = 'np' THEN 'NP'
17                ELSE di.KSI_ElementStawkaVAT
18            END AS StandardizedStawkaVAT
19        FROM
20            [BPM_Przyklady].do.KS_DocumentItems AS di /*Należy wskazać nazwę bazy BPM*/
21        WHERE KSI_DocumentId = @ID_KSeF@ /*Identyfikator kontrolki liczba całkowita, do której pobierane jest ID dokumentu KSeF */
22    ),
23    ProcessedNazwyRanked AS (
24        /* CTE 2: Porządkowanie stawek VAT z tabeli słownikowej (CDN.Nazwy), w celu zapewnienia unikalności
25        SELECT
26            Naz_GIDLP,
27            Naz_Nazwa1,
28            CASE
29                WHEN LTRIM(RTRIM(n.Naz_Nazwa1)) LIKE '%Z%' THEN 'ZW'
30                WHEN LTRIM(RTRIM(n.Naz_Nazwa1)) LIKE '%NP%' THEN 'NP'
31                WHEN CHARINDEX(' ', LTRIM(RTRIM(n.Naz_Nazwa1))) > 0 THEN LTRIM(RTRIM(SUBSTRING(n.Naz_Nazwa1, 1, CHARINDEX(' ', LTRIM(RTRIM(n.Naz_Nazwa1)))))
32                ELSE LTRIM(RTRIM(n.Naz_Nazwa1))
33            END AS StandardizedNazwa1_ForJoin,
34            CASE
35                WHEN LTRIM(RTRIM(n.Naz_Nazwa1)) LIKE '%Z%' OR LTRIM(RTRIM(n.Naz_Nazwa1)) LIKE '%NP%' THEN (
36                    WHEN ISNUMERIC(LTRIM(RTRIM(SUBSTRING(n.Naz_Nazwa1, 11, 10))) THEN CAST(LTRIM(RTRIM(SUBSTRING(n.Naz_Nazwa1, 11, 10))) AS DECIMAL(22, 4))
37                    ELSE LTRIM(RTRIM(SUBSTRING(n.Naz_Nazwa1, 11, 10)))
38                )
39            END AS StandardizedNazwa1_ForJoin
40        FROM
41            [BPM_Przyklady].do.CDN_Nazwy AS n
42        WHERE n.Naz_GIDLP = @ID_KSeF@
43    )
44    SELECT
45        di.KSI_ElementNazwa,
46        di.KSI_ElementCenaNetto,
47        di.KSI_ElementIlosc,
48        di.KSI_ElementWartoscNetto,
49        di.KSI_ElementStawkaVAT,
50        StandardizedStawkaVAT,
51        StandardizedNazwa1_ForJoin,
52        StandardizedNazwa1_ForJoin
53    FROM
54        ProcessedDocumentItems AS di
55    JOIN
56        ProcessedNazwyRanked AS n
57    ON
58        di.KSI_ElementNazwa = n.StandardizedNazwa1_ForJoin
59    ORDER BY
60        di.KSI_ElementIlosc
```

Kontrolki

- ^Userid
- ^DocumentId

Składnia:

SELECT

varchar(900) - Nazwa
decimal(22,4) - Cena
decimal(22,4) - Ilosc

Zapisz Zamknij

Administrator

Kontrolki

Szukaj

ZAKŁADKA 1

Pozycje dokumentu

[Dane tabelaryczne] Elementy

Dane tabelaryczne

OGÓLNE LISTY INICJOWANIE WARTOŚCI OGRANICZENIE EDYCJI

Inicjowanie wartości

Inicjowanie Brak

Kontrolki powiązane

Elementy (Elementy)

ID_KSeF (ID_KSeF)

Dodaj

Wzór na wartość

SQL ERP

Ustaw

Zapytanie SQL w kontrolce typu Dane tabelaryczne na zakładce Inicjowanie wartości

Dane tabelaryczne

OGÓLNE

LISTY

INICJOWANIE WARTOŚCI

OGROANICZENIE EDYCJI

Zaznaczenie opcji spowoduje zablokowanie funkcjonalności lub edycji poniższych parametrów.

	☐ Zaznacz wszystko Zablokuj dodanie wierszy	☐ Zaznacz wszystko Zablokuj usuwanie wierszy	☐ Zaznacz wszystko Nazwa Nazwa	☐ Zaznacz wszystko Cena Cena	☐ Zaznacz wszystko Ilość Ilość	☒ Zaznacz wszystko Netto Netto	☐ Zaznacz wszystko Stawka VAT Stawka VAT	☒ Zaznacz wszystko Brutto Brutto
Etap								
Rejestracja dokumentu	☐	☐	☐	☐	☐	☒	☐	☒
Dekretacja kosztów	☐	☐	☐	☐	☐	☒	☐	☒
Akceptacja kierownika	☐	☐	☐	☐	☐	☒	☐	☒
Księgowość	☐	☐	☐	☐	☐	☒	☐	☒
Odrzucono	☐	☐	☐	☐	☐	☒	☐	☒
Archiwum	☐	☐	☐	☐	☐	☒	☐	☒

Kolumna Netto i Brutto w trybie „Do odczytu”

6. **Utworzono kontrolkę liczbową:** Dodano kontrolkę typu **Liczba stałoprzecinkowa** o nazwie (identyfikatorze) **WartoscNetto**, w której dodano powiązanie do kontrolki **Elementy** i skonfigurowano w niej zapytanie, które zlicza wartość z kolumny **Netto** z kontrolki tabelarycznej:

```
select sum (Netto) from @Elementy@
```

Wyrażenie kontrolki

Uwaga!
Zmiana nazwy kontrolki może spowodować nieprawidłowe działanie wyrażenia.

```
1 SELECT SUM(Netto) FROM @Elementy@
```

Kontrolki

^Document
^UserId
^DocumentI
^Workflowc
^Stageld

Składnia:

SELECT decimal
(22,4)
decimal – wartość
stałoprzecinkowa.
Przykład:
BEGIN TRY
SELECT 1000.2588
END TRY
BEGIN CATCH
SELECT 0

ZapiszZamknij

[Liczba stałoprzecinkowa] WartoscNetto

Liczba stałoprzecinkowa

i Kontrolkę można uzupełnić liczbą złożoną z maksymalnie 18 cyfr przed przecinkiem i 6 miejsc po przecinku

Nazwa wyświetlanaWartoscNetto

Nazwa (identyfikator)WartoscNetto

Liczba miejsc po przecinku, do której będzie zaokrąglana wartość2

Prezentuj na całej szerokości

Ustaw kontrolkę na początku wiersza

Pokaż na liście

Inicjowanie wartości

InicjowanieBrak

Dokument

Pole

Zmiana wartości (Obserwator)

Kontrolki powiązaneElementy (Elementy)

Dodaj

Wzór na wartośćSQL OD

PokażUsuń

Kontrolka Liczbowa WartoscNetto

7. Utworzono kontrolkę liczbową: Dodano kontrolkę typu **Liczba stałoprzecinkowa** o nazwie (identyfikatorze) **WartoscBrutto**, w której dodano powiązanie do kontrolki **Elementy** i skonfigurowano w niej zapytanie, które zlicza wartość z kolumny Brutto z kontrolki tabelarycznej:

```
select sum (Brutto) from @Elementy@
```

Wyrażenie kontrolki

Uwaga!
Zmiana nazwy kontrolki może spowodować nieprawidłowe działanie wyrażenia.

```
1 SELECT SUM(Brutto) FROM @Elementy@
```

Kontrolki

^Document
^UserId
^DocumentI
^Workflowc
^Stageld

Składnia:

SELECT decimal
(22,4)
decimal – wartość
stałoprzecinkowa.
Przykład:
BEGIN TRY
SELECT 1000.2588
END TRY
BEGIN CATCH
SELECT 0

Zapisz
Zamknij

[Liczba stałoprzecinkowa] WartoscBrutto

Liczba stałoprzecinkowa

i Kontrolkę można uzupełnić liczbą złożoną z maksymalnie 18 cyfr przed przecinkiem i 6 miejsc po przecinku

Nazwa wyświetlanaWartoscBrutto

Nazwa (identyfikator)WartoscBrutto

Liczba miejsc po przecinku, do której będzie zaokrąglana wartość2

Prezentuj na całej szerokości

Ustaw kontrolkę na początku wiersza

Pokaż na liście

Inicjowanie wartości

InicjowanieBrak

Dokument

Pole

Zmiana wartości (Obserwator)

Kontrolki powiązaneElementy (Elementy)

Dodaj

Wzór na wartośćSQL OD

PokażUsuń

Kontrolka Liczbowa WartoscBrutto

8. Punkt ACD: W punkcie konfiguracyjnym ACD (Współpraca z KSeF), na zakładce Kontrolki, dodano kontrolkę **ID_KSeF** i przypisano jej zapytanie, które pobiera ID dokumentu KSeF:

```
select @Id1@
```

Punkt konfiguracyjny

Tryb współpracy
Ogólne
Lista
Kontrolki
Uprawnienia

Skan dokumentu (Skan)		1 select @Id1@ 2 3 4 5 6 7 8 9 10 Test zapytania SQL Składnia: SELECT int int - liczba całkowita Przykład: BEGIN TRY select value from tabela where Id = @Id1@ END TRY BEGIN CATCH DECLARE @ErrorMessage varchar(max) = 'W zapytaniu wystąpił błąd : ' + ERROR_MESSAGE();
Numer dokumentu (Numerdokumentu)		
Data wystawienia (Datawystawienia)		
Kontrahent (Kontrahent)		
Data zakupu (Datazakupu)		
Wartość Netto (WartoscNetto)		
Wartość Brutto (WartoscBrutto)		
Waluta (Waluta)		
Forma płatności (Formaplatnosci)		
Termin płatności (Terminplatnosci)		
Data wpływu (Datawpływu)		
Numer KSeF (NumerKSeF)		
IDKSeF (ID_KSeF)		

Dodaj

Zapisz
Zamknij

Kontrolka ID_KSeF w punkcie ACD

Uzupełniona kontrolka typu Dane Tabełaryczne

Po pomyślnym zainicjowaniu dokumentu z punktu ACD, **kontrolka typu Dane Tabełaryczne** została automatycznie wypełniona **pozycjami odczytanymi** z KSeF.

Dla każdej pozycji, system automatycznie przeliczył i uzupełnił wartości w kolumnach **Netto** i **Brutto** w tym samym wierszu. W przypadku nieprawidłowości w kwotach możliwe jest

przeliczanie wartości **Netto/Brutto** poprzez zmianę **Ceny/Ilości** lub **Stawki VAT**.

Kolumna **Netto** oraz **Brutto** jest zablokowana do ręcznej edycji (tryb „Do odczytu”), co gwarantuje poprawność danych.

Kontrolka **Wartość Netto** wyświetla poprawną sumę końcową z kolumny **Netto** kontrolki typu Dane tabelaryczne.

Kontrolka **Wartość Brutto** wyświetla poprawną sumę końcową z kolumny **Brutto** kontrolki Dane tabelaryczne.

Pozycje dokumentu

Elementy						
LP	Nazwa	Cena	Ilość	Netto	Stawka VAT	Brutto
1	Towar 2	67,89	100,00	6 789,00	8%	7 332,12

Podsumowanie

Wartość Netto	Wartość Brutto	ID_KSeF
6 789,00	7 332,12	1

Kontrolka typu Dane tabelaryczne uzupełniona pozycjami odczytanymi z KSeF oraz podsumowanie kwot

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

[Sprawdź Szkolenia Comarch BPM!](#)