

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 wygenerowaniem dokumentu do Comarch ERPXL

[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

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 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
17 ELSE IF @Element_Column=2 OR @Element_Column=3 OR @Element_Column=5
18 /*[2] LOGIKA: PRZELICZANIE W PRZÓD (Cena/Ilość/Stawka VAT -> Netto/Brutto) */
19 BEGIN
20 DECLARE @VAT AS DECIMAL(4, 2)
21
22 /*Pobranie stawki VAT z tabeli CDN.Nazwy */
23 SELECT @VAT = CAST(SUBSTRING(Naz Nazwa1, 0, CHARINDEX(' ', Naz Nazwa1)))
```

Kontrolki

^UserId

^DocumentId

^WorkflowId

^StageId

^InitSenderControlName

^SenderControlName

Dane_kontrahenta

Data_wplywu

Składnia:

SELECT

varchar(900) - Nazwa

decimal(22, 4) - Cena

decimal(22, 4) - Ilość

decimal(22, 4) - Netto

int - Id Stanowiska (StawkaVAT)

decimal(22, 4) - Brutto

POS - pozycja (opcjonalny numer wiersza)

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!](#)