**Wikiprint Book**

**Title: Zaawansowane dostosowanie eDokumenty dla bystrzaków**

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

**Version: 13**

**Date: 12/03/25 21:49:24**

## Table of Contents

*Zaawansowane dostosowanie eDokumenty dla bystrzaków* 3
*1 Dodatkowe pola i formatki* 3
*2 Raporty* 3
*3 Workflow* 3
*3.1 Warunki i decyzje* 3
*3.2 Komendy* 3
*3.3 Dane wejściowe* 3
*3.4 Przypisania* 3
*3.5 Parametry* 3
*3.5.1 {DOC_ID}, {PRC_ID}* 4
*3.5.2 {procedures.NAZWA_WLASNOSCI}* 4
*3.5.3 {documents.dscrpt}* 4
*4 Najważniejsze obiekty bazy danych* 4
*5 Integracja z FK* 4
*6 Własne zakładki i moduły* 4

# Zaawansowane dostosowanie eDokumenty dla bystrzaków

## 1 Dodatkowe pola i formatki

Jeżeli chcemy wyposażyć eDokumenty w dodatkowe pola na wbudowanych formatkach: dokumentu,

- sprawy,
- kartoteki kontrahenta
- urządzenia
- zdarzenia

to możemy to zrobić poprzez Panel Sterowania i odpowiedni link Cechy <nazwa formatki>. Z utworzeniem cech nie powinno być problemów, wykorzystać można wiele różnych formantów: od pól tekstowych, poprzez listy wyboru aż po specjalizowane komponenty wyboru kontaktu, sprawy czy osoby kontaktowej.

Przykładowo Tworzenie dodatkowych pól w kartotece klienta

## 2 Raporty

Poprzez odpowiednio przygotowane raporty Użytkownicy mogą zyskać dokładnie takie spojrzenie na wprowadzane do systemu dane jakie sobie wyobrazili. Raportów wcale nie trzeba wykonywać w module Raporty. Poprzez mechanizm Menu raportów możemy je podłączać do modułów a co najciekawsze również pod kartoteki klienta i sprawy. Podłączone raporty będą się wówczas wykonywać z parametrem odpowiednim dla formatki {PRC_ID} – dla sprawy, {CONTID} dla kartoteki klienta. Oczywiście w samym raporcie musimy zadeklarować ich obsługę.

Raporty mogą również przyjmować listę zaznaczonych na liście elementów np. {DOC_IDS}.

Więcej: Tworzenie raportów w SQL

## 3 Workflow

### 3.1 Warunki i decyzje

Pozwalają kierować ścieżkami workflow. Decyzje różnią się tym od warunków że są wykonywane przez użytkownika. Warunki są zaś obliczane - najczęściej poprzez odpowiednie zapytania SQL np. *SELECT amount > 1000 FROM vatnote WHERE doc_id = {DOC_ID}*, które zwracając TRUE lub FALSE kierują workflow na odpowiednią ścieżkę.

### 3.2 Komendy

Komendy pozwalają na poszczególnych etapach workflow dokonać automatycznych czynności np. utworzenia dokumentu, uprawnienia użytkowników do sprawy, dodania przypomnienia itp. Pozwalają one również na wykonywanie dowolnie złożonych sprawdzeń (komenda sprawdź warunek SQL). Lista komend jest opisana tutaj: Lista komend

### 3.3 Dane wejściowe

Jeśli podczas przebiegu procedury chcemy pobrać pewne jej parametry np. osoby opisujące fakturę, wówczas możemy zadeklarować dane wejściowe, które mogą przyjmować różne formy np. pól tekstowych, wyboru pracownika, a co najciekawsze również własnych list wyboru napełnianych danymi pobranymi kwerendą SQL.

### 3.4 Przypisania

Pobrane z danych wejściowych dane są widoczne tylko w tym etapie w którym zostały pobrane, jeżeli chcemy je wykorzystać w kolejnych etapach to należy wcześniej zdefiniować własności globalne procedury, a następnie po pobraniu danych przypisać je do zmiennych globalnych. UWAGA! przypisanie danej wejściowej do globalnej należy realizować na KONIEC (zakończenie) etapu, a nie na jego START.

Przypisania również są bardzo często używane do ustawienia osób (a w zasadzie stanowisk) które będą wykonywać dany etap. Do tego celu służy właściwość {stages.orgarr} będąca tablicą przechowującą identyfikatory stanowisk - czyli klucz orunid z tabeli orgtree_view)

### 3.5 Parametry

Ponieważ w workflow nie będziemy chcieli wszystkiego przypisywać na sztywno, możemy posługiwać się parametrami, które w czasie wykonywania warunków, komend i danych wejściowych będą przeszukiwane przez system na okoliczność wystąpienia parametrów.

Parametry są zamkniętymi w wąsach (w przyszłości również poprzedzonymi dolarem) ciągami znaków które mogą przyjąć trzy postacie:

#### 3.5.1 {DOC_ID}, {PRC_ID}

Bardzo często w warunkach, komendach i przypisaniach używa się parametrów pobieranych z aktywnych obiektów podlegających workflow tj. dokumentu {DOC_ID} lub sprawy {PRC_ID}. Na przykład jeśli chcemy sprawdzić czy są wypełnione pole uwagi dokumentu możemy napisać:

```
SELECT fixinf IS NOT NULL FROM documents WHERE doc_id = {DOC_ID}
```

wówczas ciąg znaków {DOC_ID} zostanie zamieniony przed wykonaniem na bazie zapytania na identyfikator dokumentu podlegającego workflow. Na tej samej zasadzie możemy sprawdzić w warunku czy prognozowana wartość sprawy nie jest większa od 4 milionów.

```
SELECT forepa > 4000000 FROM processes WHERE prc_id = {PRC_ID}
```

#### 3.5.2 {procedures.NAZWA_WLASNOSCI}

W ten sposób odwołujemy się w workflow do zmiennych procedury, które zadeklarujemy w jej właściwościach (na liście procedur zaznaczamy procedurę i klikamy ikonkę właściwości).

Do zadeklarowanej w ten sposób własności można następnie przypisać jakąś wartość, a następnie możemy się do niej odwoływać poprzez {procedures.NAZWA_WŁASNOŚCI}.

#### 3.5.3 {documents.dscrpt}

Czasem pomocnym może być również odwołanie się do właściwości obiektu podlegającego workflow (dokumentowi lub sprawie) można to zrobić poprzez formułę {nazwa_obiektu.nazwa_pola_z_bazy}. Działać to będzie dla obiektów *documents* i *processes*.

## 4 Najważniejsze obiekty bazy danych

| Relacja | Co przechowuje | Klucz główny i najważniejsze pola |
|---|---|---|
| documents | Dokumenty | doc_id |
| vatnote | Faktury | vtntid/doc_id |
| demand | Zapotrzebowanie | demnid/doc_id |
| fk_order_documents | Zamówienie | ordrid/doc_id |
| processes | Sprawa | prc_id |
| fk_elements | Pozycje | fkelid |
| organization_units | Struktura organizacyjna | orunid |
| users | Użytkownicy | usr_id |

**Słowniki**

| | |
|---|---|
| types_of_vcosts | RK - rodzaje kosztów |
| places_of_vcosts | MPK - miejsca powstawania kosztów |
| types_of_processes_states | Statusy |

## 5 Integracja z FK

## 6 Własne zakładki i moduły