Wikiprint Book

Title: Zaawansowane dostosowanie eDokumenty dla bystrzaków

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

Version: 13

Date: 12/06/25 04:39:19

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 4
3.5.1 {DOC_ID}, {PRC_ID} 4
3.5.2 {procedures.NAZWA_WLASNOSCI} 4
3.5.3 {documents.dscrpt} 4
3.6 Kto może wykonać dane zadanie workflow? 4
3.6.1 {stages.orgarr} 4
3.6.2 {stages.allow_} 4
4 Najważniejsze obiekty bazy danych 4
5 Integracja z FK 5
6 Własne zakładki i moduły 5
7 Rejestry 5

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)

Globalne stałe:

```
'{ENT_ID}'                => SysContext::$ent_id,
'{USR_ID}'                => UserRights::getUsrAccAsString(),
```

```
'{ORUNID}'                => UserRights::getOrgAccAsString(),
'{LOGGED_ORUNID}'        => SysContext::$usr_info['orunid'][0],
'{LOGGED_USR_ID}'        => SysContext::$usr_info['usr_id'],
'{ACTIVE_ORUNID}'        => SysContext::$usr_info['orunid'][0],
'{ACTIVE_USR_ID}'        => SysContext::$usr_info['usr_id']
```

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*.

3.6 Kto może wykonać dane zadanie workflow?

Aby wykonać zadanie worflow pracownik musi kliknąć w przycisk Załatwione. Przycisk jest widoczny dla pracowników znajdujących się na stanowiskach do których przypisano etap (patrz - stages.orgarr), jak również pokazuje się osobom posiadającym prawo do stanowiska dla którego jest zadanie - ale pod warunkiem że pracownik ma przywilej "Przywilej do wykonywania zadań workflow na podrzędnych jednostkach".

3.6.1 {stages.orgarr}

Dla każdego etapu dostępna jest własność stages.orgarr która przechowuje tablicę identyfikatorów z tabeli organization_units (orunid). Przypisanie stanowisk można zrobić za pomocą komponentu graficznego lub korzystać z zakładki przypisania i dynamicznie ustawiać tą własność:

```
Kiedy: START
Własność: {stages.orgarr}
Wartość: SELECT ARRAY(SELECT o.orunid FROM processes p INNER JOIN orgtree_view o ON p.rspuid = o.usr_id WHERE prc_id = {PR
```

3.6.2 {stages.allow_}

Jeśli dany etap może być w ramach ustalonych zasad wykonywany przez dodatkowe osoby (np. w razie nieplanowanej nieobecności) wówczas te reguły można zapisać we własności {stages.allow_}. Własność podobnie jak {stages.orgarr} przechowuje tablicę identyfikatorów jednostki organizacyjnej (organization_units.orunid).
UWAGA! Jednostki tak przypisane muszą mieć dostęp do kontekstu\nośnika procesu (dokument,sprawa,itp.), gdyż {stages.allow_} nie nadaje żadnych uprawnień, tylko i wyłącznie pozwala "kliknąć" w "Załatwione". Bez tego przypisania przycisk "Załatwione" się nie pojawia.

4 Najważniejsze obiekty bazy danych

Relacja	Co przechowuje
documents	Dokumenty
processes	Sprawa
events	Zdarzenia
fk_elements	Pozycje
vatnote	Faktury
demand	Zapotrzebowanie
fk_order_documents	Zamówienie
organization_units	Struktura organizacyjna
users	Użytkownicy
stages	Instancje etapów workflow
procedures	Instancje procedur
registers	Dzienniki (drzewko)
regofpapers	Dziennik pism

Słowniki

Nazwa tabeli	Opis
types_of_vcosts	RK - rodzaje kosztów
places_of_vcosts	MPK - miejsca powstawania kosztów
types_of_processes_states	Statusy
types_of_accountants_doc	Typy dokumentów księgowych

5 Integracja z FK

6 Własne zakładki i moduły

7 Rejestry

Rejestry