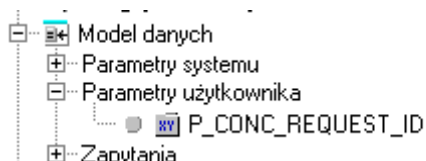


Spis treści

Uruchomienie raportu w środowisku Oracle applications	1
Pisanie do pliku dziennika.....	2
srw – drukowanie tekstu tylko na określonej stronie	2
Formatowanie kwot.....	2
Opis konta księgowego	3
MPK	3
Czytanie standardowych komunikatów	3
filtry na wielocłonowce (lokalizacje, klucz, kategorie)	4
Aby wymusić zmienny porządek sortowania	5
Aby właściwie zostały posortowane dane w grupie	6
Komunikat „brak rekordów”	7
Zmienna etykieta	9
Przerwanie wykonania raportu.....	9
Odpowiednik form_ddl	9
Dodanie ramki otaczającej ramkę	9
Wydruk ramki powtarzanej od nowej strony	9
dynamiczna zmiana zapytania (warunek where, zmiana nazwa tabeli)	10
Łączenie zapytań SQL.....	11
Sygnal 11/ Signal 11 podczas próby utworzenia XML	12
Zakończenie działania raportu z ostrzeżeniem / błędem	12
Diagnostyka raportu	13

Uruchomienie raportu w środowisku Oracle applications

1. w beforeReport wstaw:
`srw.user_exit('FND SRWINIT');`
2. w afterReport wstaw:
`srw.user_exit('FND SRWEXIT');`
3. Raport musi mieć parametr P_CONC_REQUEST_ID.
Parametr ten jest inicjowany przez `srw.user_exit('FND SRWINIT')`



4. Jeżeli generujesz raport w formacie PDF, nie możesz używać dowolnej czcionki (symptom błędu: urwany plik pdf). Użyj: Courier new

Pisanie do pliku dziennika

```
srw.message ('100', 'BeforeReport Trigger +');  
100 i 'BeforeReport Trigger +' - dowolne stringi
```

srw – drukowanie tekstu tylko na określonej stronie

```
function B_REPORT_DATE1FormatTrigger return boolean is  
  pn number(10);  
begin  
  srw.get_page_num(pn);  
  if pn=1 then  
    return (TRUE);  
  else  
    return (FALSE);  
  end if;  
RETURN NULL; end;
```

Formatowanie kwot

```
function BOOK_LIKWIDACJE_I_DSPFormula return VARCHAR2 is  
begin  
  SRW.REFERENCE(:BOOK_LIKWIDACJE_I);  
  SRW.USER_EXIT('FND FORMAT_CURRENCY  
  CODE=":X_CURRENCY_CODE"  
  DISPLAY_WIDTH="14"  
  AMOUNT=":BOOK_LIKWIDACJE_I"  
  DISPLAY=":BOOK_LIKWIDACJE_I_DSP");  
  RETURN(:BOOK_LIKWIDACJE_I_DSP);  
end;
```

uwaga: mimo, że można podać parametr "minimalna precyzja", to kwota jest zawsze wyświetlana z precyzją, z jaką została zapisana. Użyj funkcjo Round(.,2) jeżeli chcesz zaokrąglić kwotę.

Uwaga dotycząca XML Publisher

Nie używaj Oracle Reports do budowania pliku XML. Zamiast tego używaj XDO data template. Uzasadnienie: Oracle Reports generuje kwoty rozdzielając

części całkowite i ułamkowe przecinkiem i layout nie potrafi sformatować tej wartości do liczby

Opis konta księgowego

```
SRW.USER_EXIT('FND FLEXIDVAL
              CODE="GL#"
              APPL_SHORT_NAME="SQLGL"
              DATA=":ACCOUNT_ALL"
              NUM=":STRUCT_NUM"
              DISPLAY="ALL"
              IDISPLAY="ALL"
              DESCRIPTION=":ACCOUNT_DESC");
```

MPK

```
srw.user_exit('FND FLEXSQL
              CODE="GL#"
              NUM=":P_CHART_OF_ACCOUNTS_ID"
              APPL_SHORT_NAME="SQLGL"
              OUTPUT=":P_MPK"
              TABLEALIAS="gcc"
              MODE="SELECT"
              DISPLAY="FA_COST_CTR");
```

result: gcc.segment3

Czytanie standardowych komunikatów

zob. select * from FND_NEW_MESSAGES where language_code = 'PL'

```
DECLARE
  l_message FND_NEW_MESSAGES.message_text%TYPE;
BEGIN
  FND_MESSAGE.set_name('CE', 'CE_FC_NO_HEADER');
  l_message:=FND_MESSAGE.get;
  srw.message('2', l_message);
END;
```

filtry na wielocłonowce (lokalizacje, klucz, kategorie)

```
if :P_CAT_FROM is NOT NULL then
  srw.user_exit('FND FLEXSQL CODE="CAT#" NUM=101
    APPL_SHORT_NAME="OFA"
    OUTPUT=":CP_WHERE_KATEGORIA" TABLEALIAS="FC"
    MODE="WHERE" DISPLAY="ALL"
    OPERATOR="BETWEEN"
    OPERAND1=":P_CAT_FROM"
    OPERAND2=":P_CAT_TO");
```

```
:CP_WHERE_KATEGORIA := ' AND ( '||NVL(:CP_WHERE_KATEGORIA,'1=1')||' )';
end if;
```

-- Klucz OD - DO

```
if :P_KEY_FROM is NOT NULL then
  srw.user_exit('FND FLEXSQL CODE="KEY#" NUM=101
    APPL_SHORT_NAME="OFA"
    OUTPUT=":CP_WHERE_KLUCZ" TABLEALIAS="AK"
    MODE="WHERE" DISPLAY="ALL"
    OPERATOR="BETWEEN"
    OPERAND1=":P_KEY_FROM"
    OPERAND2=":P_KEY_TO");
```

```
:CP_WHERE_KLUCZ := ' AND ( '||NVL(:CP_WHERE_KLUCZ,'1=1')||' )';
end if;
```

-- LOCATION OD - DO

```
if :P_LOC_FROM is NOT NULL then
  :CP_LOC_TABLES := ' ,FA_LOCATIONS lc '||
    ' , FA_DISTRIBUTION_HISTORY DH ';
```

```
srw.user_exit('FND FLEXSQL CODE="LOC#" NUM=101
  APPL_SHORT_NAME="OFA"
  OUTPUT=":CP_WHERE_LOCATION" TABLEALIAS="LC"
  MODE="WHERE" DISPLAY="ALL"
  OPERATOR="BETWEEN"
  OPERAND1=":P_LOC_FROM"
  OPERAND2=":P_LOC_TO");
```

```
:CP_WHERE_LOCATION := ' and lc.location_id = dh.location_id '||
```

```
' and dh.book_type_code = buki.book_type_code '||  
' and dh.asset_id = plan.asset_id '||  
' and dh.date_ineffective is null '||  
      ' AND ( '||NVL(:CP_WHERE_LOCATION,'1=1')||' )';  
end if;
```

Aby wymusić zmienny porządek sortowania

Jeżeli w zapytaniu są grupy, wówczas klauzula **order by** zapytania nie jest brana pod uwagę (następuje resortowanie na kliencie zgodnie z definicją porządku sortowania w grupie).

1. dodaj kolumnę **SORT** jak w przykładzie poniżej (w zależności od parametru p_porzadkowanie w kolumnie tej znajdzie się nazwa, trx_date lub invoice_no). Pamiętaj, że wszystkie kolumny muszą być typu char.
2. Dodaj kolumnę **SORT** do grupy i TYLKO tej kolumnie w grupie nadaj właściwość „kolejność przerwania = rosnący” (właściwość jest oznaczana na diagramie jako strzałka na żółtym tle).

Przeczytaj też koniecznie sekcję „Aby właściwie zostały posortowane dane w grupie”.

Podobnie można obsłużyć parametr rosnąco-malejąco. Wówczas trzeba utworzyć dwie kolumny: sort_asc oraz sort_desc, przy czym w przypadku sortowania malejącego kolumna sort_asc powinna przyjmować wartość stałą.

The screenshot shows a query builder window with several tables: O_SETUP, G_SETUP, O_DETAIL, G_Customer, G_Invoice, and G_DETAIL. The G_Invoice table has a field named SORT circled in red. Below the table list, the SQL text is displayed, with the word SORT circled in red in the query:

```

select
decode (:p_porzadkowanie, 'Typ transakcji',to_char(arpt_sql_func_util.get_trx_type_details(ct.cust_trx_type_id,'NAME'),'Numer faktury',ps.trx_number,'Data
faktury',to_char(ps.trx_date,'yyyy-mm-dd')) SORT)
cust.account_id customer_id,
substrb(party.party_name,1,50) customer_name,
cust.account_number customer_number,
SUBSTR(arpt_sql_func_util.get_trx_type_details(ct.cust_trx_type_id,'NAME'),1,15) Name,
ps.customer_trx_id,
:p_porzadkowanie INVOICE_NO,
SUBSTR(ps.trx_number,1,10) Invoice_No,

```

Buttons for 'Query Builder...', 'Importuj zapytanie SQL...', 'Połącz...', 'OK', 'Anuluj', and 'Pomoc' are visible at the bottom.

Aby właściwie zostały posortowane dane w grupie

Nawet jeśli określisz porządek sortowania dla ostatniej grupy (w przykładzie poniżej po lewej stronie to pola C_INTEREST_BATCH_ID, C_INVOICE_NUMBER itd.), to nie zostaną one wzięte pod uwagę (dane nie zostaną posortowane).

Zwróć uwagę, że jeżeli istnieje tylko jedna grupa, to jest ona ostatnią grupą, czyli problem dotyczy wielu raportów.

Aby ominąć ten problem utwórz sztucznie grupę nadrzędną lub podrzędną (nie trzeba wprowadzać żadnych zmian w layoutcie).

W przykładzie poniżej po prawej stronie utworzono sztuczną grupę nadrzędną.

G_c_invoice_date	
31	c_invoice_date

G_JGZZ_II_LINES	
789	c_interest_batch_id
A	CF_invoice_date
A	c_invoice_number
31	c_due_date
A	CF_due_date
31	c_payment_date
A	CF_payment_date
789	c_outstanding_amount
789	c_days_late
789	c_original_amount
789	c_days_overdue
789	c_interest_charged
A	c_class
789	c_days_of_interest
789	c_interest_rate
789	C_credits

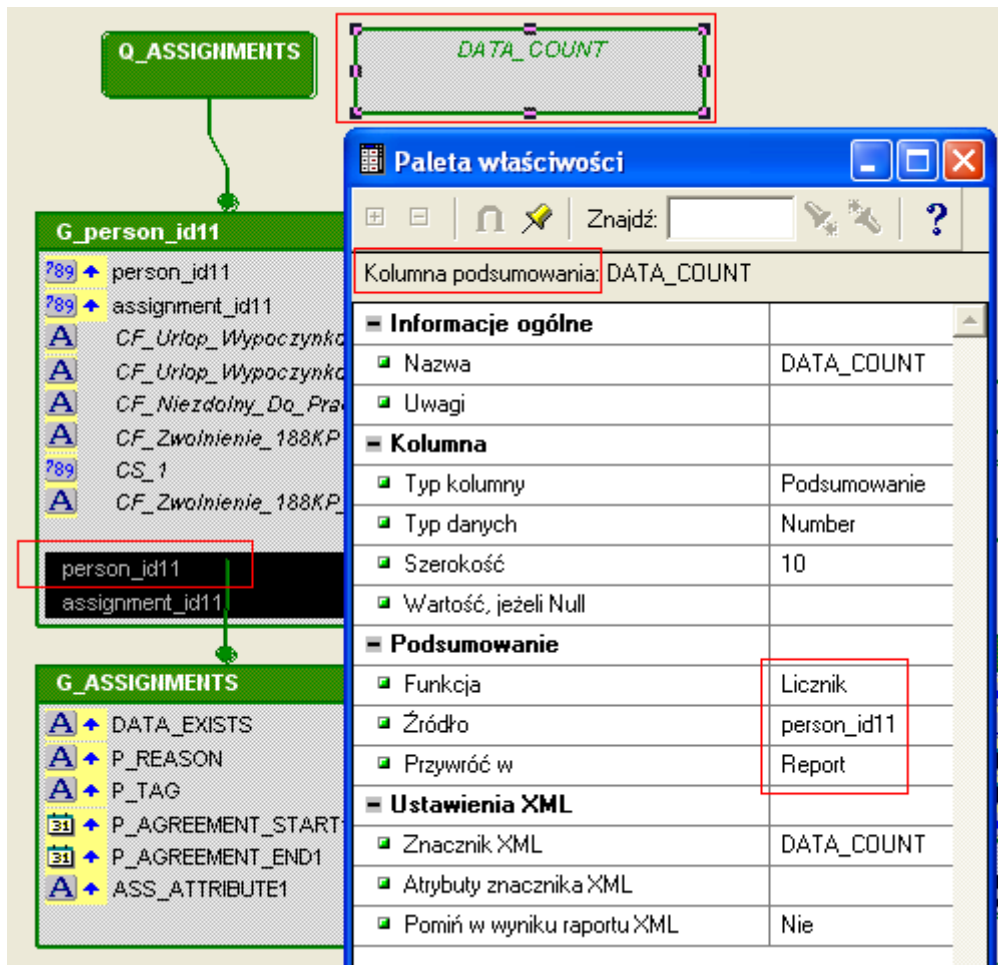
G_c_invoice_date	
31	c_invoice_date

G_c_interest_batch_id	
789	c_interest_batch_id

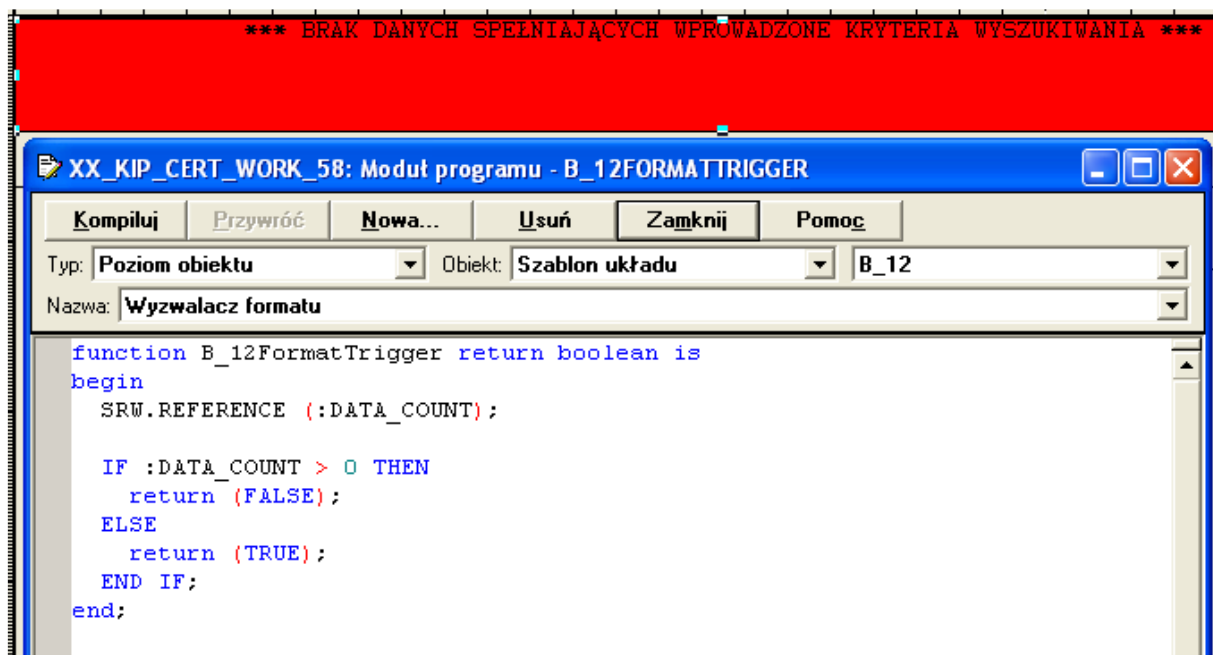
G_JGZZ_II_LINES	
A	CF_invoice_date
A	c_invoice_number
31	c_due_date
A	CF_due_date
31	c_payment_date
A	CF_payment_date
789	c_outstanding_amount
789	c_days_late
789	c_original_amount
789	c_days_overdue
789	c_interest_charged
A	c_class
789	c_days_of_interest
789	c_interest_rate
789	C_credits
789	C_number_overdue

Komunikat „brak rekordów”

Dodaj kolumnę podsumowania, zliczającą rekordy.



Dodaj Tekst jak na rysunku poniżej. Zdefiniuj wyzwalacz określający widoczność w sposób pokazany poniżej.



Zmienna etykieta

W polu tekstowym (etykieta) można robić odwołania do innych obiektów

Przykład: etykieta odwoła się do pola tekstowego o nazwie F_TOP1, gdy wpiszesz w tekście etykiety
"Konta &F_ROP1"

Przerwanie wykonania raportu

```
SRW.MESSAGE(001,'Wystąpił błąd podczas wczytywania danych. Realizacja  
raportu zostaje przerwana.');
```

```
RAISE SRW.PROGRAM_ABORT;
```

Odpowiednik form_ddl

```
srw.do_sql('alter session set sql_trace=true');
```

Dodanie ramki otaczającej ramkę

1. Dodaj na układzie ramkę w ten sposób, żeby w całości otaczała ramkę podrzędną
w razie potrzeby zmniejsz na chwilę rozmiary ramki podrzędnej.
2. Uczyń aktywnym okno układu, zaznacz nowododaną ramkę
3. Naciskaj razy klawisz F8. Po każdym naciśnięciu obiekt ramka będzie schodził coraz niżej. Zakończ naciskanie klawisza F8, gdy ramka znajdzie się na samym dole.

Wydruk ramki powtarzanej od nowej strony

1. dla WSZYSTKICH elementów na raporcie, w tym również dla ramki powtarzanej ustaw następujące właściwości:
drukuj obiekt = pierwsza strona
pionowa elastyczność = zmienny
2. Dla ramki powtarzanej:
maksymalna liczba rekordów na stronie = 1
łamane strony po = tak
pozostałe właściwości domyślnie (kierunek drukowania = w dół, liczba rekordów szweców = 0, ...)

Ramka powtarzana nie musi posiadać ramki nadrzędnej, nie musi także posiadać ramek podrzędnych.

dynamiczna zmiana zapytania (warunek where, zmiana nazwa tabeli)

1. w zapytaniu wstaw

```
select *
FROM XXFA_BALANCES_REPORT bal
     ,FA_ADDITIONS_B ad
     ,FA_CATEGORIES_B ca
     ,GL_CODE_COMBINATIONS cc
     ,FA_ASSET_KEYWORDS ak
     &CP_LOC_TABLES
WHERE      ad.ASSET_ID = bal.ASSET_ID
          AND ca.CATEGORY_ID = bal.CATEGORY_ID
          AND cc.CODE_COMBINATION_ID = bal.DISTRIBUTION_CCID
          AND ad.ASSET_KEY_CCID = ak.CODE_COMBINATION_ID
          AND bal.REQUEST_ID = :P_CONC_REQUEST_ID
---> to      &CP_WHERE_ASSET_NBR
```

2. ustaw CP_WHERE_ASSET_NBR na odpowiednią wartość w zdarzeniu beforeReport

```
function BeforeReport return boolean is
begin
  IF :P_NR_ST IS NOT NULL THEN
    :CP_WHERE_ASSET_NBR := ' AND AD.ASSET_NUMBER LIKE
UPPER(''||:P_NR_ST||') ';
  END IF;
end;
```

Uwaga:

W taki sam sposób można parametryzować np. nazwy tabel, np:

```
select *
FROM XXFA_BALANCES_REPORT bal
     ,FA_ASSET_KEYWORDS ak
     &CP_LOC_TABLES
WHERE ad.ASSET_ID = bal.ASSET_ID
      AND ca.CATEGORY_ID = bal.CATEGORY_ID
      AND cc.CODE_COMBINATION_ID = bal.DISTRIBUTION_CCID
```

```

AND ad.ASSET_KEY_CCID = ak.CODE_COMBINATION_ID
AND bal.REQUEST_ID = :P_CONC_REQUEST_ID
&CP_WHERE_ASSET_NBR

```

```

if :P_LOC_FROM is NOT NULL then
  :CP_LOC_TABLES := ',FA_LOCATIONS lc';

```

```

srw.user_exit('FND FLEXSQL CODE="LOC#" NUM=101
  APPL_SHORT_NAME="OFA"
  OUTPUT=":CP_WHERE_LOCATION" TABLEALIAS="LC"
  MODE="WHERE" DISPLAY="ALL"
  OPERATOR="BETWEEN"
  OPERAND1=":P_LOC_FROM"
  OPERAND2=":P_LOC_TO");

```

```

:CP_WHERE_LOCATION := ' AND lc.LOCATION_ID =
bal.LOCATION_ID ||
  ' AND ( ||NVL(:CP_WHERE_LOCATION,'1=1')|| )';
end if;

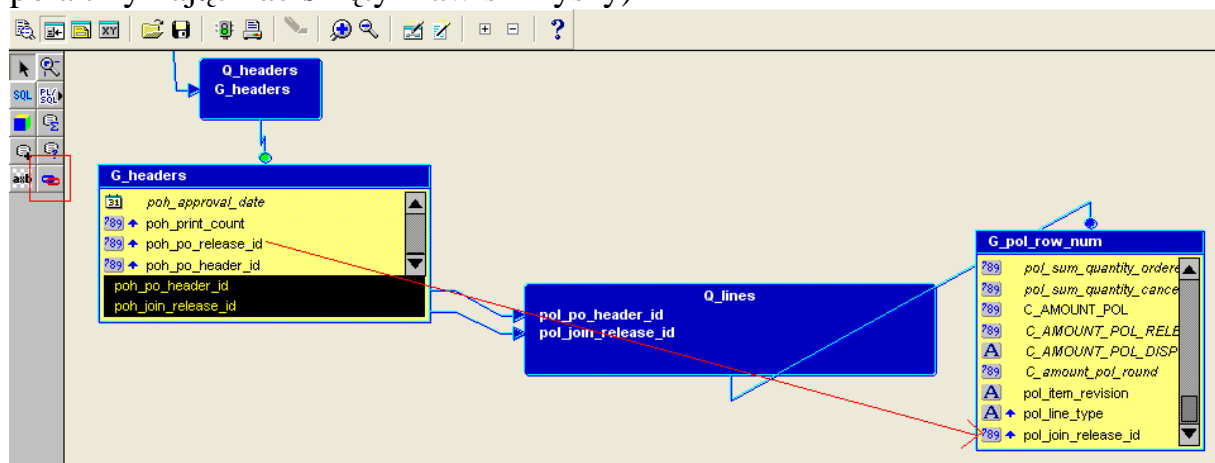
```

Łączenie zapytań SQL

Raport może składać się z kilku zapytań SQL, połączonych w relacji nadrzędny-podrzędny (zapytania mogą też nie być ze sobą połączone)

Są dwie możliwości połączenia zapytań:

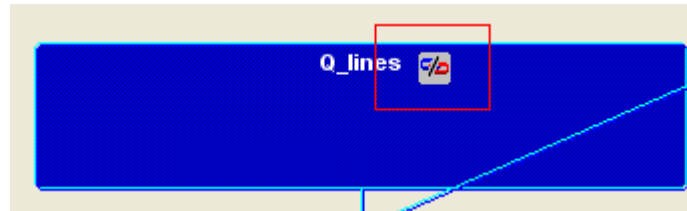
1/ zalecane – przez utworzenie powiązania (kliknij w zaznaczoną ikonę i powiąż pola trzymając naciśnięty klawisz myszy)



2/ Przez wpisanie wprost w zapytaniu podrzędnym powiązania w warunku where np. where pol_join_release_id = :poh_po_release_id

Uwaga:

Nie będziesz mógł powiązać zapytań, jeżeli jedno z zapytań zawiera zapytania zagnieżdżone. Wówczas Query jest oznaczone za pomocą symbolu „otwarte złączenie” :



W takim przypadku „ukryj” przed raportem zagnieżdżenia w widoku w bazie danych.

Sygnal 11/ Signal 11 podczas próby utworzenia XML

Zob. XMLKnowledge.doc

Zakończenie działania raportu z ostrzeżeniem / błędem

```
declare
  res boolean;
begin
  res := fnd_concurrent.set_completion_status('C', 'Komunikat');
  --C = Normal
  --G = Warning
  --E = Error
end;
```

Diagnostyka raportu

Programy współbieżne

Program **PPL - Asygnata zasiłkowa** Włączone

Skrót **XXXKIP67**

Aplikacja **Rozszerzenia Kadry i Płace**

Opis **Asygnata zasiłkowa**

Plik wykonywalny

Nazwa **XXXKIP67** Opcje **RUNDEBUG=YES TRACEFILE=/c**

Metoda **Oracle Reports** Priorytet

Zlecenie

Typ

El. przyrostowy

Funkcja MLS

Użycie w SUZ Dozw. wyłączone wartości

Uruch. osobno Uruch. ponow. przy błęd. sys.

Włącz śledzenie Podlega NLS

Wyjście

Format. **Tekst**

Zapisz

Drukuj

Kolumny **180**

Wiersze **66**

Styl **Wszerz**

Wymagany styl

Drukarka

Kopiuj dg... Kontrola sesji Niezgodności Parametry

RUNDEBUG=YES TRACEFILE=/dout/TST4/XXXKIP67.trc
TRACEMODE=TRACE_REPLACE TRACE_OPTS=(TRACE_ALL)

Plik tworzy się na serwerze aplikacyjnym (nie tworzy się na serwerze bazy danych)

Przykładowy plik dziennika:

```
DZIENNIK :
:      Raport: /tstu03/tst3/tst3appl/xxkip/11.5.0/reports/PL/XXXKIP67.rdf
      Zalogowany do serwera:
      Użytkownik:

DZIENNIK :
      Zalogowany do serwera: TST3
      Użytkownik: APPS

09:31:21  ERR REP-1305: Ostrzeżenie: Ramka 'M_4' przecina ale nie obejmuje obiektu
'B_36'.
09:31:21  ERR REP-1305: Ostrzeżenie: Ramka 'M_5' przecina ale nie obejmuje obiektu
'B_33'.
09:31:21  ERR REP-1319: Ostrzeżenie: Niejawny algorytm kotwicz+cy nie może
jednoznacznie
zidentyfikować obiektu kotwicz+cego 'B_15'.
Wybrano 'F_29'
09:31:21  ERR REP-1319: Ostrzeżenie: Niejawny algorytm kotwicz+cy nie może
```

```

jednoznacznie
zidentyfikować obiektu kotwiczacego 'B_16'.
Wybrano 'F_31'
09:31:21 APP ( Kolumna bazy danych P_DATE_EFF
09:31:21 APP ) Kolumna bazy danych P_DATE_EFF
09:31:21 APP ( Kolumna bazy danych P_ASSIGNMENT_ID
09:31:21 APP ) Kolumna bazy danych P_ASSIGNMENT_ID
09:31:21 APP ( Kolumna bazy danych P_absence_id
09:31:21 APP ) Kolumna bazy danych P_absence_id
09:31:21 PLS ( Funkcja: beforereport
09:31:22 MSG MSG-01001: Początek
09:31:22 MSG MSG-01003: :p_eff_date=26-MAR-09
09:31:22 MSG MSG-02000: poz A
09:31:22 MSG MSG-02000: vbusiness_group_id 81
09:31:22 MSG MSG-02000: vsecurity_profile_id 81
09:31:22 MSG MSG-02000: :p_date_eff 26-MAR-09
09:31:22 MSG MSG-02000: :p_assignment_id 12142
09:31:22 MSG MSG-02000: :p_absence_id 14129
09:32:15 MSG MSG-02000: Dane pobrane
09:32:15 MSG MSG-02000: Koniec
09:32:15 PLS ) Funkcja: beforereport
09:32:15 APP ( Ramka
09:32:15 APP . ( Ramka M_1
09:32:15 APP .. ( Ramka powtarzana R_10
09:32:15 APP ... ( Grupa G_PERSON_ID Lokalne przerwanie: 0
Globalne przerwanie: 0
09:32:15 APP .... ( Zapytanie Q_Main
09:32:15 SQL EXECUTE QUERY : select * from xxkip67_tmp
09:32:15 APP .... ) Zapytanie Q_Main
09:32:15 APP ... ) Grupa G_PERSON_ID
09:32:15 APP ... ( Szablon tekstowy B_7
09:32:15 APP ... ) Szablon tekstowy B_7
09:32:15 APP ... ( Szablon tekstowy B_6
09:32:15 APP ... ) Szablon tekstowy B_6
09:32:15 APP ... ( Szablon tekstowy B_5
09:32:15 APP ... ) Szablon tekstowy B_5
09:32:15 APP ... ( Szablon tekstowy B_4
09:32:15 APP ... ) Szablon tekstowy B_4
09:32:15 APP ... ( Szablon tekstowy B_10
09:32:15 APP ... ) Szablon tekstowy B_10
09:32:15 APP ... ( Szablon tekstowy B_9
09:32:15 APP ... ) Szablon tekstowy B_9
09:32:15 APP ... ( Szablon tekstowy B_8
09:32:15 APP ... ) Szablon tekstowy B_8
09:32:15 APP ... ( Pole tekstowe F_41
09:32:15 APP .... ( Kolumna bazy danych REGON_FIRMY
09:32:15 APP .... ) Kolumna bazy danych REGON_FIRMY
09:32:15 APP ... ( Pole tekstowe F_41
09:32:15 APP ... ( Pole tekstowe F_40
09:32:15 APP .... ( Kolumna bazy danych NIP_FIRMY
09:32:15 APP .... ) Kolumna bazy danych NIP_FIRMY
09:32:15 APP ... ( Pole tekstowe F_40
09:32:15 APP ... ( Pole tekstowe F_36
09:32:15 APP .... ( Kolumna bazy danych NUSP_FIRMY
09:32:15 APP .... ) Kolumna bazy danych NUSP_FIRMY
09:32:15 APP ... ( Pole tekstowe F_36
09:32:15 APP ... ( Pole tekstowe F_25
09:32:15 APP .... ( Kolumna bazy danych ADRES_ZAM
09:32:15 APP .... ) Kolumna bazy danych ADRES_ZAM
09:32:15 APP ... ( Pole tekstowe F_25
09:32:15 APP ... ( Pole tekstowe F_28
09:32:15 APP .... ( Kolumna bazy danych SERIA_NR_DOKUMENTU
09:32:15 APP .... ) Kolumna bazy danych SERIA_NR_DOKUMENTU
09:32:15 APP ... ( Pole tekstowe F_28
09:32:15 APP ... ( Szablon tekstowy B_14
09:32:15 APP ... ) Szablon tekstowy B_14
09:32:15 APP ... ( Pole tekstowe F_30
09:32:15 APP .... ( Kolumna bazy danych WYMIAR_CZASU_PRACY

```

```

09:32:15 APP .... ) Kolumna bazy danych          WYMIAR_CZASU_PRACY
09:32:15 APP ... ) Pole tekstowe                    F_30
09:32:15 APP ... ( Szablon tekstowy                B_11
09:32:15 APP ... ) Szablon tekstowy                B_11
09:32:15 APP ... ( Pole tekstowe                    F_27
09:32:15 APP .... ( Kolumna bazy danych            NIP
09:32:15 APP .... ) Kolumna bazy danych            NIP
09:32:15 APP ... ) Pole tekstowe                    F_27
09:32:15 APP ... ( Pole tekstowe                    F_26
09:32:15 APP .... ( Kolumna bazy danych            PESEL
09:32:15 APP .... ) Kolumna bazy danych            PESEL
09:32:15 APP ... ) Pole tekstowe                    F_26
09:32:15 APP ... ( Szablon tekstowy                B_12
09:32:15 APP ... ) Szablon tekstowy                B_12
09:32:15 APP ... ( Szablon tekstowy                B_13
09:32:15 APP ... ) Szablon tekstowy                B_13
09:32:15 APP ... ( Pole tekstowe                    F_29
09:32:15 APP .... ( Kolumna bazy danych            DATA_ZATRUDNIENIA
09:32:15 APP .... ) Kolumna bazy danych            DATA_ZATRUDNIENIA
09:32:15 APP ... ) Pole tekstowe                    F_29
09:32:15 APP ... ( Szablon tekstowy                B_15
09:32:15 APP ... ) Szablon tekstowy                B_15
09:32:15 APP ... ( Pole tekstowe                    F_31
09:32:15 APP .... ( Kolumna bazy danych            DATA_ZATRUDNIENIE_USTALO
09:32:15 APP .... ) Kolumna bazy danych            DATA_ZATRUDNIENIE_USTALO
09:32:15 APP ... ) Pole tekstowe                    F_31
09:32:15 APP ... ( Szablon tekstowy                B_16
09:32:15 APP ... ) Szablon tekstowy                B_16
09:32:15 APP ... ( Ramka powtarzana                R_1
09:32:15 APP .... ( Grupa                          G_02   Lokalne przerwanie: 0
Globalne przerwanie: 0
09:32:15 APP ..... ( Zapytanie                      Q_02
09:32:15 SQL          EXECUTE QUERY : select   inf_person_id
, inf_position
, inf_assignment_id g02_assignment_id
, inf_przyczyna g02_prev_employer
, inf_date_start g02_date_start
, inf_date_end g02_date_end
from      xxkip67_inf_wyplaty
WHERE ( inf_position = '02' ) AND ( :PERSON_ID = inf_person_id) order by
inf_date_start

                Zmienna wiązania                Wartość
                -----                -----
                PERSON_ID                12084

(...)

+-----+
| Statystyki profilowania Report Builder |
+-----+

Ł~ CZNY CZAS, KTÓRY UPŁYŃ~Ł:                55.25 sekundy

Czas Reports:                54.33 sekundy (98.33% CAŁO|CI)

Czas ORACLE:                0.91 sekundy ( 1.66% CAŁO|CI)

                UPI:                0.12 sekundy
                SQL:                0.79 sekundy

Ł~ CZNY czas CPU zużyty przez proces:        0.22 sekundy

```