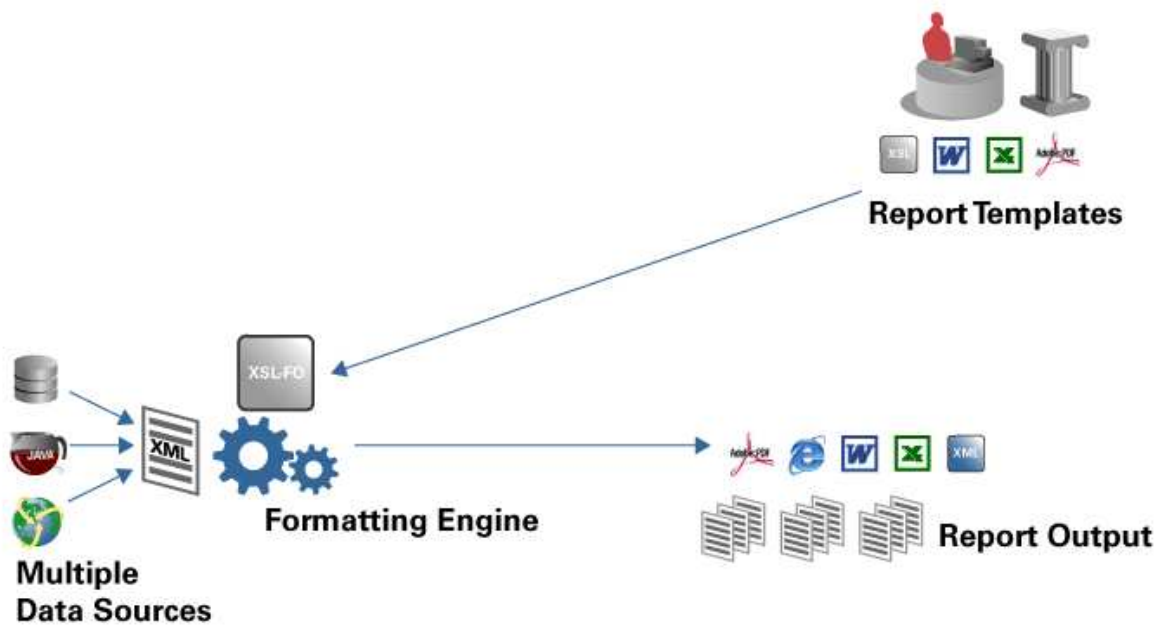


Index

Co to jest XML Publisher	1
Skąd ściągnąć	2
Documentation	2
Przykłady raportów	2
Własne	2
Z dokumentacji.....	2
Szablon w formacie xls-fo a nie rtf	3
Szablon w formacie xls a nie rtf	3
Tabela krzyżowa	3
EBS: Jak zarejestrować raport w Aplikacji	3
EBS: Jak zarejestrować raport w Aplikacji – plik sterujący XML.....	5
jeżeli ... to	7
wartości NULL	7
Ukrywanie sekcji bez danych w pliku XML.....	8
Wstawianie daty bieżącej	8
Ustawianie numeru na pierwszej stronie	9
Niezależne numerowanie stron dokumentów.....	9
Liczba porządkowa.....	9
Podział strony (łamanie strony)	9
Checkbox.....	10
Oracle Reports => XML Publisher.....	11
Odwołanie do elementu w grupie nadrzędnej	11
Odwołanie do elementu w dowolnej grupie.....	11
Odwołanie do elementu bez łamania wiersza	11
Kolorowanie komórki / formatowanie warunkowe	11
Raport dla Word i raport dla Excel.....	11
Formatowanie kwot.....	12
Jak zapobiec wycinaniu zer wiodących przez excel.....	12
Problem z polskimi znakami / strona kodowa	12
java.sql.SQLException: ORA-01009: brak obowiązkowego parametru.....	13
Parametry raportu nie działają.....	14
Komunikat „Brak danych”	14
Adobe Reader nie może otworzyć 'pa014841.fdf'	14
Uruchomienie zlecenia współbieżnego XML Publisher.....	15
Zakończenie działania raportu z ostrzeżeniem / z błędem	15
Upload pliku *.rtf dla XMLP/XLF	15
EBS: Złote rady	16
EBS: Diagnostyka błędów / debug	16
EBS: Out of memory.....	17
EBS: błąd “errmsg:został przerwany przez sygnał 11” (signal)	17
EBS: XDO layout/ data template zablokowane przed edycją	18
Format liczb i kwot w pliku XML	18
EBS: translations (xlf)	18
EBS: lexicals.....	20
Summaries (podsumowania).....	20
EBS: Zmiana ID parametrów na nazwy	21
Displaying Images in BI Publisher Reports (Repost)	21
Multilingual BI Publisher Reports for Siebel CRM.....	23
BI Publisher Debugging.....	26

Co to jest XML Publisher



Skąd ściągnąć

<http://edelivery.oracle.com>

Oracle Business Intelligence (10.1.3) Media Pack for Microsoft Windows (32-bit)
 Oracle Business Intelligence Publisher Desktop 10.1.3.4.1 for Microsoft Windows

Platform: Microsoft Windows(32bit)

Documentation

<http://www.oracle.com/technetwork/middleware/bi-publisher/documentation/xmlpdocs-084437.html>

first look: <http://www.rittman.net/archives/001385.html>

Userguide :

C:\Program Files\Oracle\XML Publisher Desktop\Template Builder for Word\doc\XDO5.6UserGuide.pdf

<http://xdo.us.oracle.com>

<http://blogs.oracle.com/xmlpublisher/newsItems/departments/xmlpForEBusinessSuite>

<http://www.w3.org/TR/xsl/>

Przykłady raportów

Własne

Zajrzyj do katalogu examples

Z dokumentacji

<http://www.oracle.com/technology/products/applications/publishing/index.html>

Szablon w formacie xls-fo a nie rtf

Zob. PPL/ZAW007

Szablon w formacie xls a nie rtf

Idea: group name is stored here:



Ask Wojtek Pietrasik for details.

Znalazłem trochę dokumentacji do szablonów Excelowych w XMLP (BIP), dostępna tu:

http://docs.oracle.com/cd/E21764_01/bi.1111/e13881/T527073T571887.htm

wersję PDF można ściągnąć z powyższej strony.

Odniesienie tych dziwnych tagów (np. <?ref:xdo0003?>) generowanych w polach przez najnowszą wersję Oracle BI Publisher Desktop (11.1.1.5.0 for Windows), to znalazłem takie info na metalinku (ID 1138602.1):

Note: The Template Builder for Excel 11.1.1.5.0 is **not** included in the above listed patches. For true Excel Template design ([Creating Excel Templates](#)) it is required to:

1. Uninstall the BI Publisher Desktop 10.1.3.2.1
2. Install Oracle BI Publisher Desktop 11.1.1.5.0 for Windows ([download](#))
3. Use the Template Builder for Excel 11.1.1.5.0 for creating Excel templates

Oracle E-Business Suite 11i in Release 12 customers **must not** use the Template Builder for Word as part of Oracle BI Publisher Desktop 11.1.1.5.0 for RTF layout template design as it is not backwards compatible.

Wynikałoby z tego, że do tworzenia template'ów Excel potrzebny jest BIP Desktop 11.1.1.5.0 ale jeśli chcemy tworzyć RTF szablony na potrzeby EBSa to nie możemy go używać

Tabela krzyżowa

PPL.E4A2_RCP_05_Lista_obecnosci_do_podpisu_RCP

EBS: Jak zarejestrować raport w Aplikacji

Responsibility „XML Publisher Administrator”

1. Create concurrent program with **XML output**

If you can use XML template – select XDODTEXE executable (more in the next section)

If not – select executable that generates XML output (any executable program - for example PLSQL package).

2. Test whether xml file with data is correctly created (view output).
3. Create *.rtf template

4. Register xml and *.rtf templates. Join fields according to description:

Concurrent program
| short name
| code
Data definition
| Name
| data definition
Template
* File
* language
* region

Remarks:

! When you register template both Language and **Region** must be entered

! Select **XML output** for conc request

ORACLE XML Publisher

[Strona startowa](#) [Wylógowanie](#)

[Szablony](#) [Definicje danych](#)

[Definicje danych](#) >

Przegląd definicji danych: XXPRTINV_XML

[Aktualizuj](#) [E](#)

Nazwa: XXPRTINV_XML
Aplikacja: Rozszerzenia
Schemat XML:
Podgląd danych:

Kod: XXPRTINV_XML
Data początkowa: 22-wrz-2006
Data końcowa:
Szablon danych:

Opis:

Ten ekran omiń jeżeli generujesz plik za pomocą pakietu lub reports (nie używasz XDODTEXE)

Pliki

Schemat XML

Szablon danych **DATA_TEMPLATE_XTEXT_XXZAW005.xml**

Podgląd danych

Plik sterujący burstingu

Szablony >

Przegląd szablonu: Faktura - wydruk graficzny

Ogólne

Nazwa	Faktura - wydruk graficzny	Kod	XXPRTINV_XML	<input type="button" value="Aktualizuj"/>
Aplikacja	Rozszerzenia	Definicja danych	XXPRTINV_XML	<input type="button" value=""/>
Typ	RTF	Data początkowa	22-wrz-2006	
Domyślny plik	invoice.rtf	Data końcowa		
Domyślny plik - język	Polski	Szablon podrzędny	Nie	
Domyślny plik - terytorium	POLSKA			
Opis	<input type="text"/>			

Important in R12 !

Authorization: Administracja systemu, współbieżne -> programy

Select * from po_headers returns 0 records ? Tryb jednostki operacyjnej has to be set.

This property of concurrent program is not available on forms form.

Uaktualnianie programu współbieżnego: XX_KAS_ZA_ZAO_026 - Windows Internet Explorer

http://lamignat-pl.pl.oracle.com:8026/OA_HTML/OA.jsp?page=/oracle/apps/fnd/cp/program/webui/Create

Uaktualnianie adnotacji

Włączone Tak Nie

Nazwa skrótnowa **XX_KAS_ZA_ZAO_026**

Opis

* Nazwa programu Raport historii zamówie

Aplikacja Rozszerzenia

Typ programu

Plik wykonywalny Parametry Niezgodności **Zlecenie** Ustawienia dotyczące strony Statystyka

Ustawienia zlecenia

Typ

Funkcja MLS

Element podsumowania działań

Element przyrostowy

Opcje pliku wykonywalnego

Odświeżanie portletu

Tryb jednostki operacyjnej **Pojedyncza**

Preferencje wyjścia

open issues:

Multilanguage – see “Administracja systemu” responsibility

EBS: Jak zarejestrować raport w Aplikacji – plik sterujący XML

Wszystkie uwagi jak w punkcie poprzednim.

Oprócz tego :

1. Umieść na serwerze plik sterujący (zastępujący pakiet lub oracle reports).

Definicje danych >
 Przegląd definicji danych: XXFA_VAT_YEAR_DN Aktualizuj

Nazwa	XXFA_VAT_YEAR_DN	Kod	XXFA_VAT_YEAR_DM
Aplikacja	Rozszerzenia	Data początkowa	10-gru-2007
Schemat XML		Data końcowa	
Podgląd danych		Szablon danych	count_corr_year_report_VAT.xml

Opis: Raport roczny odliczeń i korekt VAT

[Powrót do definicji danych](#) Aktualizuj

[Szablony](#) | [Definicje danych](#) | [Administracja](#) | [Strona startowa](#) | [Wylogowanie](#) | [Preferencje](#) | [Pomoc](#) | [Strona personalizacji](#) | [Diagnostyka](#)

Copyright (c) 2006, Oracle. All rights reserved.
 Informacje na temat tej strony.



count_corr_year_report_VAT.xml

Do edycji pliku możesz użyć darmowego narzędzia XML Notepad.

2. Wpisz nazwę programu wykonywalnego dla programu współbieżnego : XDODTEXE

3. Uwagi dotyczące pliku sterującego:

- Parametry zdefiniowane w pliku XML muszą odpowiadać parametrom programu współbieżnego – nazwy i typy i wielkość liter
- Możesz uruchomić procedurę z pakietu w bazie danych. Musi być to funkcja zwracająca wartość boolean (false=błąd).
- Jeżeli używasz pakietu w bazie danych: Parametry te muszą być także zdefiniowane w specyfikacji pakietu o nazwie określonej w zmiennej defaultPackage.

```

<?xml version="1.0" encoding="utf-8" ?>
<!-- Created 2007-12-07 8Maszczyk -->
- <dataTemplate name="count_corr_year_report_VAT" description="Raport roczny odliczeń i korekt VAT" version="1.0" defaultPackage="XXFA_VAT_PKG" >
- <parameters>
  <parameter name="PS_FINANCIAL_YEAR" dataType="character" />
  <parameter name="PS_ASSET_NUM_FROM" dataType="character" />
  <parameter name="PS_ASSET_NUM_TO" dataType="character" />
  <parameter name="PS_REPORT_TYPE" dataType="character" />
</parameters>
- <dataQuery>
- <sqlStatement name="Q_LOCATION">
  - </sqlStatement>
  
```

Parametry programu współbieżnego

Program: **Raport roczny odliczeń i korekt VAT**
 Aplikacja: **Rozszerzenia**

Domena z konflikt: Grupa zabezpieczeń:

Lp.	Parametr	Opis	Włącz.
10	PS_FINANCIAL_YEAR	Rok podatkowy	<input checked="" type="checkbox"/>
20	PS_ASSET_NUM_FROM	Od numeru środka trwałego	<input checked="" type="checkbox"/>
30	PS_ASSET_NUM_TO	Do numeru środka trwałego	<input checked="" type="checkbox"/>
40	PS_REPORT_TYPE	Tryb przetwarzania raportu	<input checked="" type="checkbox"/>

Zatwierdzenie

Zestaw wartości: **XX_FINANCIAL_YEAR** Opis:
 Domyślny typ: Wartość domyślna:
 Wymagane Włącz zabezpieczenia Zakres:

Wyświetlaj

Rozmiar wyświetlania: **10** Rozmiar opisu: **50**
 Łączna długość opisu: **25** Etykieta: **Rok podatkowy**

Token: **PS_FINANCIAL_YEAR**

```
CREATE OR REPLACE package xxfa_vat_pkg as
```

```
PS_FINANCIAL_YEAR varchar2(4);
PS_ASSET_NUM_FROM varchar2(15);
PS_ASSET_NUM_TO varchar2(15);
PS_REPORT_TYPE varchar2(1);
```

```
function beforeReportTrigger (
  preport_type varchar2 --M/Y
  ,ppreliminary_final_flag varchar2 --P/F
  ,xperiod varchar2
) return boolean;
```

```
ORDER BY xfai.vat_period vat_period
]]>
</sqlStatement>
</dataQuery>
<dataTrigger name="BEFOREREPORT" source="XXFA_VAT_PKG.BEFOREREPORTTRIGGER('Y', :PS_REPORT_TYPE, :PS_FINANCIAL_YEAR)" />
- <datastructure>
- <group name="G_LOCATION" source="Q_LOCATION">
  <element name="LOCATION_CODE" value="LOCATION_CODE" />
</group>
- <group name="G_LOGIN" source="Q_LOGIN">
  <element name="LOGIN" value="LOGIN" />
</group>
```

jeżeli ... to ...

```
<?xdofx:if PLEC='F' then 'Pani' else 'Pan' end if?>
```

wartości NULL

```
<?if:element_name!=?> desired behavior when the element exists and is not null <end if?>
<?if:element_name and element_name=?> desired behavior when the element exists but value is null <end if?>
<?if:not(element_name)?> desired behavior when the element is not present <end if?>
<?if:SHOULDBE_LINES/LINE_SHOULDBE_LINES[1]?> desired behavior when element exists but element has child LINE_SHOULDBE_LINES
<end if?>
```

Ukrywanie sekcji bez danych w pliku XML

Plik XML:

```
<PRZEDMIOT tytul="Przedmiot:" nr="1.">  
<EL>  
<NR>1</NR>  
<OPIS>ZAKUP ART. SPOŻYWCZYCH</OPIS>  
<ILOSC></ILOSC>  
<JM>sztuka</JM>  
</EL>  
</PRZEDMIOT>
```

Layout:

F:PRZEDMIOT

if

nr tytul

Nr	Opis	Ilość	Jednostka
F NR	OPIS	ILOSC	JM E

Etc

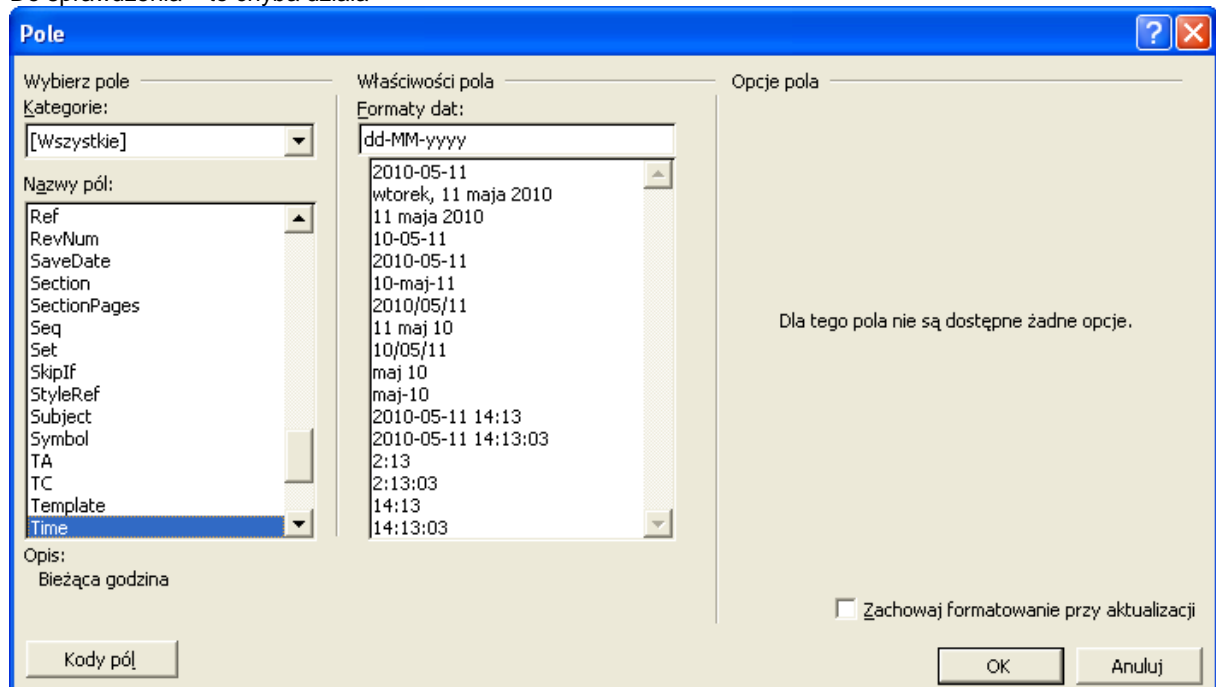
E

Wstawianie daty bieżącej

Report Date: <?format-date:xdoxslt:sysdate_as_xsdformat();'LONG'?>

Nie używaj standardowego pola: Date - nie działa.

Do sprawdzenia – to chyba działa



Więcej:

<http://www.oracle.com/technology/products/xml-publisher/blog/xslttext/10134/oracle/apps/xdo/template/rtf/XSLTFunctions.html>

Ustawianie numeru na pierwszej stronie

<?initial-page-number:10?>

Niezależne numerowanie stron dokumentów

Użyj składnię @section, nic więcej nie trzeba robić.

Jeżeli jest zagnieżdżonych kilka sekcji to @section trzeba dopisać na każdym poziomie zagnieżdżenia.

Zob. wydruk faktury PPL

Liczba porządkowa

1/ Użyj standardowej funkcjonalności numerowania kolejnych wierszy

2/ Czasem standardowa funkcjonalność numerowania kolejnych wierszy w komórce nie działa. Aby obejść problem użyj zmiennej:

set_lp

F_EMP lp	lp++	E_EMP
-------------	------	-------

3/ You can also use PL/SQL function within group name section. Solution 2 and 3 are equivalent.

```
<group name="G_DETAILS" dataType="varchar2" source="Q_MAIN">  
  <element name="CF_TRANS_VAT_OR_OFFSET" dataType="number" value="CF_trans_VAT_or_Offset" />  
  ...  
  <element name="CF_SEQ_NO" dataType="number" value="Xxswm_jg_zz_summary_ap_pkg.CF_seq_no" />  
</group>
```

```
...  
FUNCTION CF_seq_no RETURN NUMBER  
IS  
BEGIN  
  G_SEQ_NO := G_SEQ_NO + 1;  
  RETURN (G_SEQ_NO);  
END CF_seq_no;  
...
```

Podział strony (łamanie strony)

Sposób 1. Wada: nie numeruje niezależnie dokumentów

```
MAIN LOOP  
PAGE_BR  
END MAIN LOOP
```

```
<?if: position() mod 3 = 0?>  
  PAGE BREAK APPLIED  
  <?split-by-page-break:?>  
<?end if?>
```

Sposób 2 (@section). Wada: czasami wstawia pustą stronę

```
MAIN LOOP  
END MAIN LOOP
```

```
When you use @section with the BI Publisher commands for-each or for-each-group (for example: <?for-each@section:ELEMENT_NAME?>), and the input data file has no data, then an empty or invalid PDF output document may be generated for that for-each loop. To prevent this from happening, enter the following in your RTF template:  
1. At the end of your RTF template, add a section break  
2. On the last page (the new section page), add the command <?if@section:not(ELEMENT_NAME)?>No Data Found<?end if?>  
where ELEMENT_NAME is the same data element that you are using in your for-each@section loop.  
Now if no data exists for ELEMENT_NAME, a valid PDF will be generated with the text "No Data Found".
```

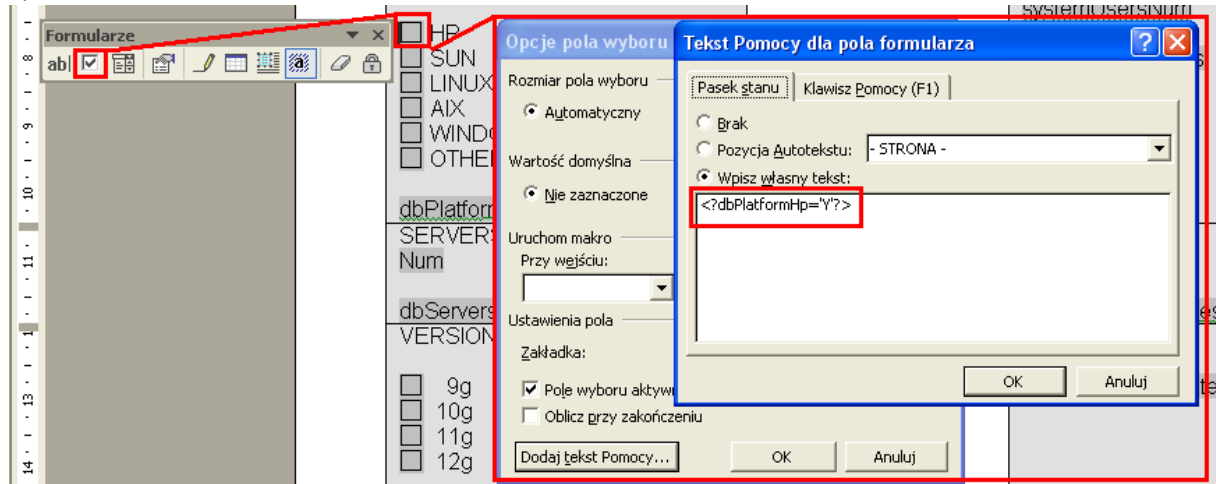
Sposób 3. Wada: czasami wstawia pustą stronę (sposób 2 jest lepszy niż 3)

```
<?split-by-page-break:?>
```

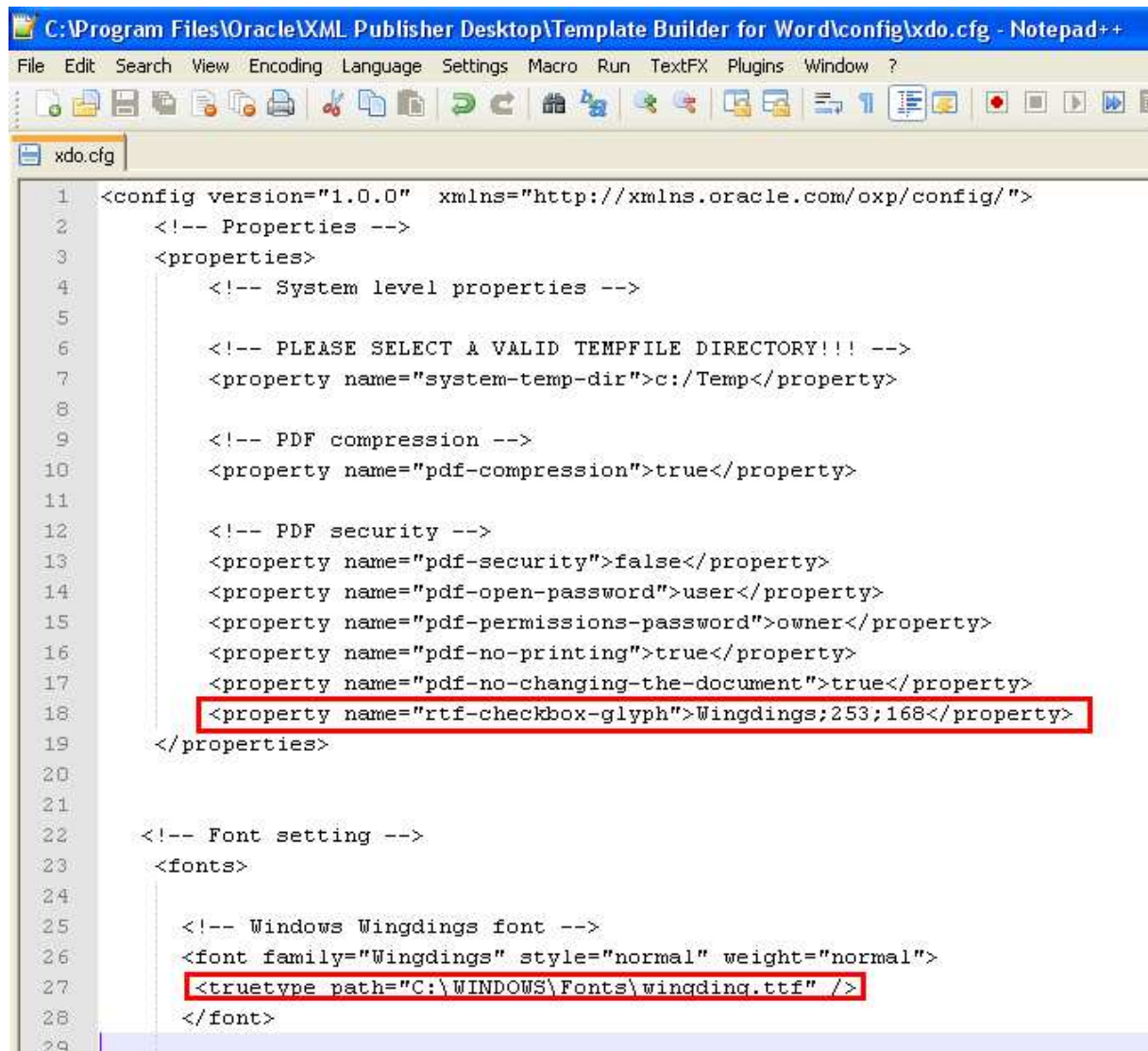
Sposób 4. Wstawienie sekcji. Wada: zawsze wstawia pustą stronę

Checkbox

1/



2/



Oracle Reports => XML Publisher

XML Publisher has internal tool RDF2RTF convertor to migrate oracle report to RTF Template. This can be accessed from XML Publisher website (xdo.us.oracle.com) under Demo=>RDF2RTF tab. To access this you should have access to Oracle network.

Zidentyfikowane błędy w konwerterze RDF2RTF:

- nie przenosi triggerow, przenosi informacje, ze byl trigger
- gubi informacje o triggerach (patrz poprzednie)
- nie radzi sobie z referencjami &nazwa_pola
- błędnie przenosi layout (wzajemne położenie elementów, formatowanie)

Odwołanie do elementu w grupie nadrzędnej

```
<?.././DECREASE_INCREASE_AMOUNT_TEXT?>
```

../ - jeden poziom wyżej

.././ - dwa poziomy wyżej

Odwołanie do elementu w dowolnej grupie

```
<?//DECREASE_INCREASE_AMOUNT_TEXT?>
```

uwaga: jeżeli element występuje w dokumencie wielokrotnie, to zostanie wybrana pierwsza instancja od góry. Czyli można w ten sposób czytać parametry raportu, ale nie można czytać wartości z grup(ramek) powtarzalnych.

Odwołanie do elementu bez łamania wiersza

Użyj słowa kluczowego „@inlines”, np.

FAKTURĘ PRZYGOTOWAŁ: Jan Kowal -->, NR TEL. 604 22 24 58<--

Kolorowanie komórki / formatowanie warunkowe

```
<?if:LINE_NUM=1?><xsl:attribute xdofo:ctx="block" name="background-color">#FF0000</xsl:attribute><?end if?>
```

#FF0000 = RGB format

można stosować mnemoniki aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, and yellow

Więcej na temat kolorów: <http://www.w3.org/TR/xsl/>

Raport dla Word i raport dla Excel

Mimo że raport może być generowany do dowolnego formatu należy pamiętać o następujących właściwościach XML Publisher.

Właściwość	Word	Excel
Tabele zagnieżdżone	Wyświetlane prawidłowo	Powoduje przesunięcie layoutu (excel z definicji nie obsługuje tabel zagnieżdżonych)

Zob. formatowanie kwot		
Zob. jak zapobiec wycinaniu zer wiodących		
Podsumowania		Lepiej nie dodawać podsumowań w XML, bo będą myliły się z danymi. Zamiast tego lepiej wykonać je w excelu adhoc w razie potrzeby

Formatowanie kwot

1/

formatowanie liczb do pdf i excel tak aby przez excel traktowane były jako liczby

- dodać pole (Decimal Format) z wpisem: `<xsl:decimal-format name="Euro"`

`decimal-separator="," grouping-separator=" " xdofo:ctx="begin"/>`

- formatować: `<?if:WARTOSC_BRUTTO!="?><?format-number(WARTOSC_BRUTTO,"# ##0,00","Euro")?><?end if?>`

- dodać dwie spacje przed polem aby wartość była tekstowa

2/

`<?format-number:SUMWARTOSCPERREPORT;'99999999999999990D00'?>`

Nie używaj formatu 999G999G999G999G999G999G990D00 (10 000 000,00), ponieważ w Excelu kwota zostanie potraktowana jako tekst i nie będzie można wykonywać działań

Więcej : zob. XDO5.6UserGuide.pdf, szukaj „Number and Date Formatting”

Jak zapobiec wycinaniu zer wiodących przez excel

1. Wstaw **dwie** spacje przed wartością

2. Wstaw znak ' przed wartością

2. Wstaw wyrażenie

`= "<?ORGANIZATION_CODE?>"`

wadą tej metody jest wyświetlanie w programie Word wartości w postaci `"0123"`

Problem z polskimi znakami / strona kodowa

1. Mimo, że Aplikacja prawidłowo generuje raport, to podczas próby uruchomienia podglądu raportu w programie MS Word, pojawia się komunikat „Select valid XML data” lub podobny ?

Rozwiązanie:

Zmień nagłówek pliku na następujący:

`<?xml version="1.0" encoding="ISO-8859-2"?>`

lub

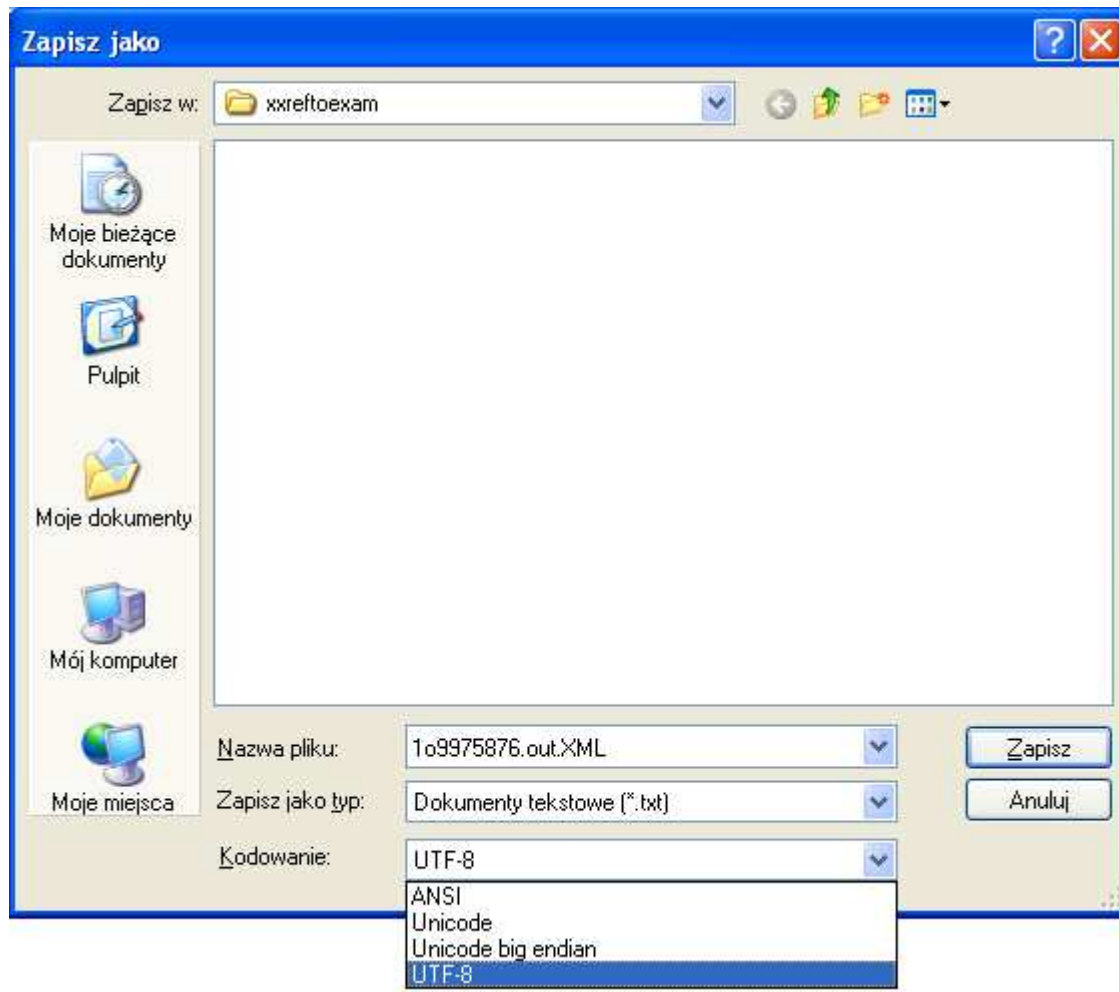
`<?xml version="1.0" encoding="UTF-8"?>`

lub

`<?xml version="1.0" encoding="windows-1250"?>`

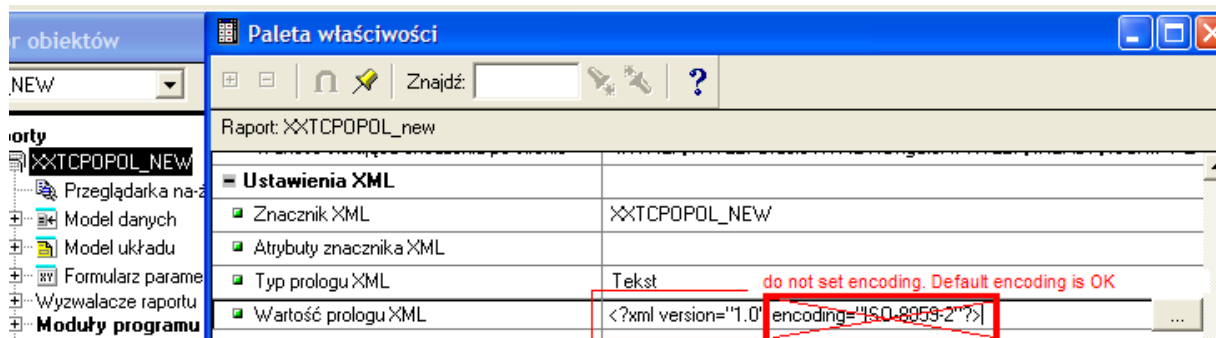
sprawdź eksperymentalnie który pasuje.

Nadal problem ? zapisz plik w standardzie ANSI:



2. Problem z polskimi znakami podczas tworzenia data template w technologii reports ?

Rozwiązanie – nie ustawiaj kodowania na poziomie nagłówka:



java.sql.SQLException: ORA-01009: brak obowiązkowego parametru

źle (ORA-01009)

```
...
<sqlStatement name="Q1"><![CDATA[
/*podstawowe informacje opisujące szablon*/
select
```

```
description template_name
, (select tl.description description
...

```

dobrze

```
...
<sqlStatement name="Q1"><![CDATA[
select
/*podstawowe informacje opisujące szablon*/
description template_name
, (select tl.description description
...

```

Parametry raportu nie działają

1. Istotna jest wielkość liter w parametrach. Zastosuj wszędzie jednolicie małe lub wielkie litery (nazwa tokenu, plik data template)
2. nie używaj składni „select :nazwa_parametru nazwa_parametru from dual”, zamiast tego użyj „select :nazwa_parametru jakaś_inna_nazwa_parametru from dual”
3. Parametry daty deklaruj w pliku jako character i dokonuj konwersji za pomocą funkcji **fn_date.canonical_to_Date (lub deklaruj jako 'date' i nie konwertuj)**

Komunikat „Brak danych”

1/ Jeżeli nie używasz składni @section (jest to bardzo prawdopodobne), generującej każdy dokument od nowej strony, zrób tak:.

```
If
                                End of Report
End

If
                                No Data Fund
End

```

W tym przypadku możesz też zastosować rozwiązanie opisane dla narzędzia Oracle Reports (dodaj kolumnę podsumowania zliczającą liczbę rekordów i uzależnij układ od wartości tej kolumny)

2/ Problem jest niebanalny, gdy używasz składni @section, ponieważ powoduje ona, że żadna strona nie jest wyświetlana, gdy brakuje danych, przy czym XML Publisher ignoruje wszelkie sposoby obejścia tego problemu. Wówczas zastosuj następujące rozwiązanie:

1/ Pobierz dane w wyzwalaczu BeforeReport i zachowaj je w tabeli tymczasowej (użyj global temporary data). Jeżeli Twoje zapytanie zawiera funkcje nie akceptowane przez Oracle Reports (funkcje analityczne, trim, zagnieżdżone polecenia select itd), wówczas użyj funkcji w pakiecie w bazie danych.

2/ Jeżeli żadne wiersze nie zostały pobrane, to wstaw wiersz z wartością „NO_DATA_FOUND”

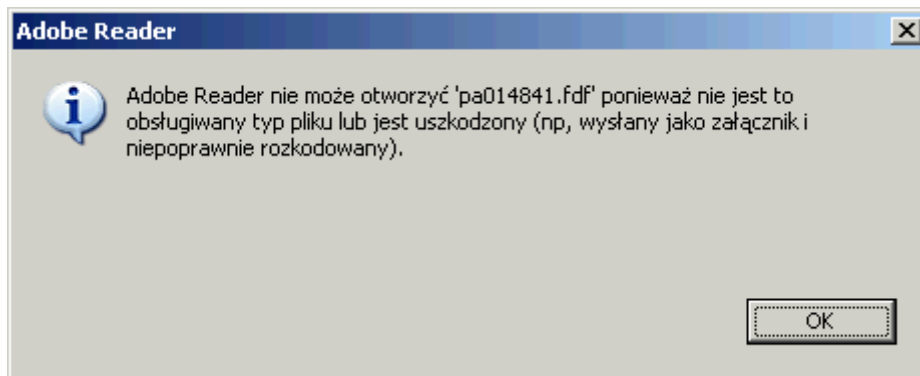
3/ Obsłuż wystąpienie wiersza „NO_DATA_FOUND” w raporcie, przykładowo:

```
NO_DATA_FOUND { Nie znaleziono żadnych danych do wydrukowania }
```

Przykład: raport XXAPBAL

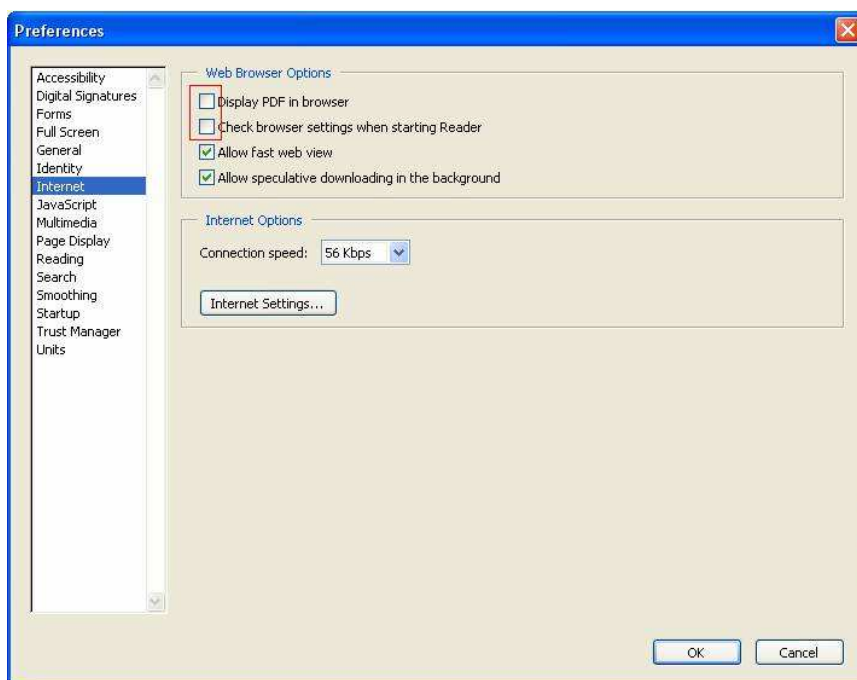
Adobe Reader nie może otworzyć 'pa014841.fdf'

Symptom:



Solution:

Problem polegający na występowaniu komunikatu (rys powyżej) podczas próby wyświetlenia pliku wyników zlecenia współbieżnego można rozwiązać, zmieniając dwa ustawienia w oknie właściwości programu Adobe Reader, zgodnie z rysunkiem poniżej.



Uruchomienie zlecenia współbieżnego XML Publisher

Zob. forms_OraApp.doc / sekcja Uruchomienie zlecenia współbieżnego (bez okna uruchamiania)

Zakończenie działania raportu z ostrzeżeniem / z błędem

`fnd_concurrent.set_completion_status` nie odnosi efektu (Zob. plik nt. Oracle Reports, sekcja o tym samym tytule).

Użyj `raise_application_error(-20000, 'Komunikat o błędzie');`

Upload pliku *.rtf dla XMLP/XLF

Zob. Wykonaj za pomocą programu fndloader.exe

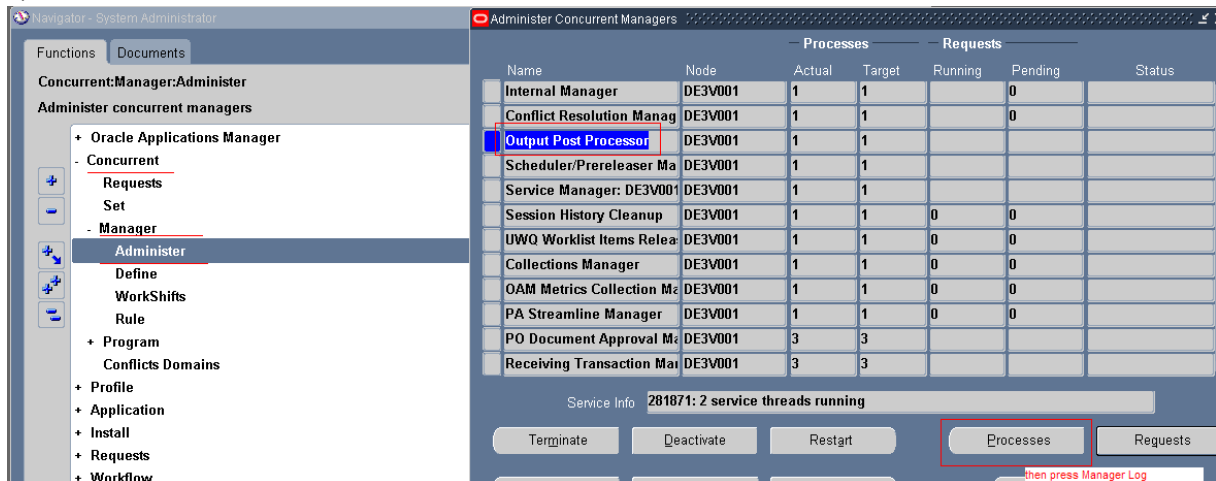
Xlf: <http://killamsetty.blogspot.com/2009/02/how-to-use-xdoloader.html>

EBS: Złote rady

- ➔ 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

EBS: Diagnostyka błędów / debug

1/



		Processes		Requests		
Name	Node	Actual	Target	Running	Pending	Status
Internal Manager	DE3V001	1	1		0	
Conflict Resolution Manag	DE3V001	1	1		0	
Output Post Processor	DE3V001	1	1			
Scheduler/Prereleaser Ma	DE3V001	1	1			
Service Manager: DE3V001	DE3V001	1	1			
Session History Cleanup	DE3V001	1	1	0	0	
UWQ Worklist Items Relea	DE3V001	1	1	0	0	
Collections Manager	DE3V001	1	1	0	0	
OAM Metrics Collection Ma	DE3V001	1	1	0	0	
PA Streamline Manager	DE3V001	1	1	0	0	
PO Document Approval Ma	DE3V001	3	3			
Receiving Transaction Ma	DE3V001	3	3			

Service Info 281871: 2 service threads running

Buttons: Terminate, Deactivate, Restart, Processes, Requests

2/

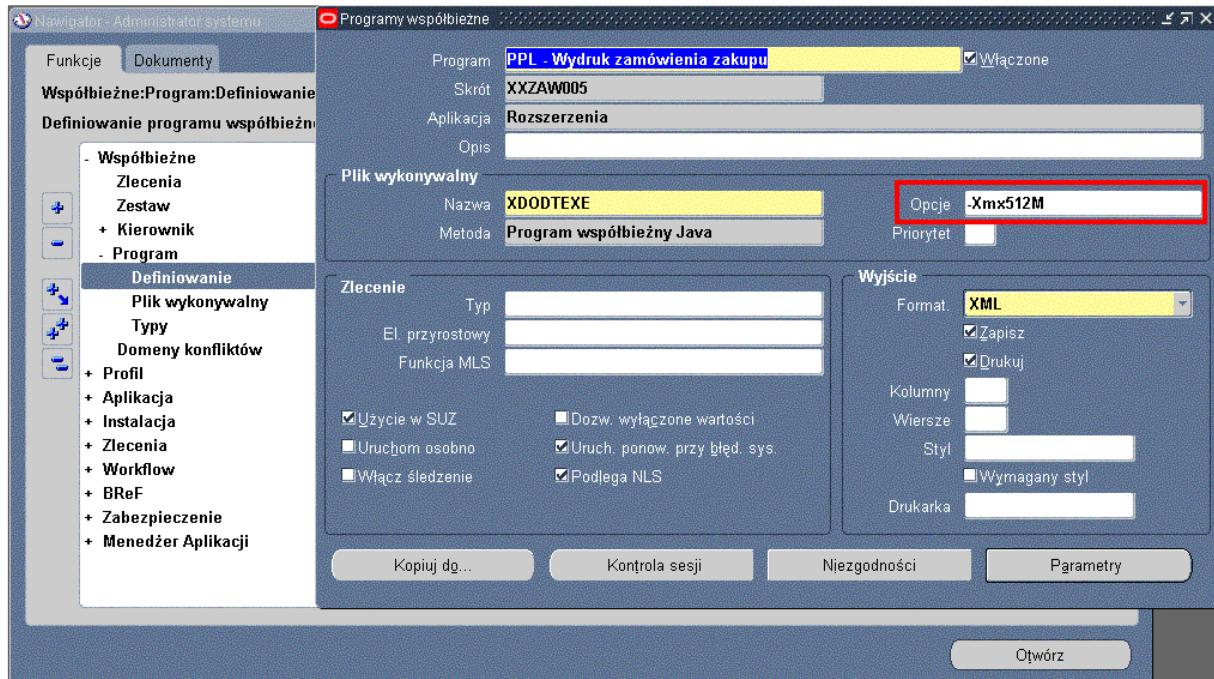
Wstaw do data template. Debug będzie w logu zlecenia

```
<dataTemplate name="XXE3PAY0119"
  description="PPL - Dodatek stazowy"
  defaultPackage="XXE3PAY0119_PKG">
```

```
<properties>
  <property name="include_parameters" value="true" />
  <property name="include_null_element" value="true" />
  <property name="xml_tag_case" value="upper" />
  <property name="db_fetch_size" value="500" />
  <property name="scalable_mode" value="off" />
  <property name="include_rowsettag" value="true" />
  <property name="debug_mode" value="on" />
</properties>
```

```
<parameters>
  <parameter name="P_DATA_OD" dataType="date" />
  <parameter name="P_DATA_DO" dataType="date" />
  <parameter name="P_ZMIANA_UPRAWNIEN" dataType="character" />
  <parameter name="P_PERSON_ID" dataType="character" />
  <parameter name="P_ORG_STRUCTURE_ID" dataType="character" />
  <parameter name="P_ORG_STRUCTURE_VERSION_ID" dataType="character" />
  <parameter name="P_ORG_ID" dataType="character" />
  <parameter name="P_CZY_PODRZEDNE" dataType="character" />
  <parameter name="P_PROCENT_PODATKU" dataType="number" />
  <parameter name="P_SORTOWANIE" dataType="character" />
</parameters>
```


EBS: Out of memory



-Xmx512M

EBS: błąd “ormsg:został przerwany przez sygnał 11” (signal)

Symptom

Plik rdf nie generuje plik XML, komunikat w pliku dziennika

stat_low = 8B

stat_high = 0

ormsg:został przerwany przez sygnał 11

Rozwiązanie

1. Zmień wyjście programu współbieżnego na TEXT, uruchom raport. Jeżeli raport nadal się nie generuje, zajrzyj do logu, prawdopodobnie zobaczysz teraz komunikaty o błędach zamiast „sygnał 11”, popraw błędy.

2. Problem prawdopodobnie związany jest z polami wyliczonymi. Usuń z raportu wszystkie pola wyliczane. Uruchom raport ponownie, żeby sprawdzić, czy problem dotyczy pól wyliczanych. Czytaj dalej.

3. Nadal problem? Przyczyną są polskie znaki w nazwie pola, w znaczniku XML lub w nagłówku procedury. Usuń polskie znaki.

Paleta właściwości	
Kolumna formuły: skl_data_ostatniej_wplaty	
Informacje ogólne	
Nazwa	skl_data_ostatniej_wplaty
Uwagi	
Kolumna	
Typ kolumny	Wzór
Typ danych	Date
Szerokość	20
Wartość, jeżeli Null	
Wartość dla maski formatu równej NULL	
Kolejność przerwania	Brak
Wypełniacz/Formuła	
Wzór PL/SQL	
Ustawienia XML	
Znacznik XML	SKL_DATA_OSTATNIEJ_WPLATY
Atrybuty znacznika XML	

4. Nadal problem? Zwiększ szerokości dla kolumn formuł. Gdy wynik zapytania zwraca polskie znaki Reports potrzebuje więcej znaków na ich zapisanie niż wynika to z faktycznej długości (polski znak jest kodowany na więcej niż jednym bajcie)

EBS: XDO layout/ data template zablokowane przed edycją

Raport został wykonany na loginie SYSADMIN, a następnie przeniesiony z do innego środowiska. W celu odblokowania edycji wykonaj następujące polecenia SQL:

```
UPDATE XDO_DS_DEFINITIONS_B SET CREATED_BY = :new_user_id, LAST_UPDATED_BY = :new_user_id WHERE DATA_SOURCE_CODE = :reaname
UPDATE XDO_DS_DEFINITIONS_TL SET CREATED_BY = :new_user_id, LAST_UPDATED_BY = :new_user_id WHERE DATA_SOURCE_CODE = :reaname
UPDATE XDO_LOBS SET CREATED_BY = :new_user_id, LAST_UPDATED_BY = :new_user_id WHERE LOB_CODE = :reaname
UPDATE XDO_TEMPLATES_TL SET CREATED_BY = :new_user_id, LAST_UPDATED_BY = :new_user_id WHERE TEMPLATE_CODE = :reaname
UPDATE XDO_TEMPLATES_B SET CREATED_BY = :new_user_id, LAST_UPDATED_BY = :new_user_id WHERE TEMPLATE_CODE = :reaname
commit
```

Format liczb i kwot w pliku XML

Liczby muszą być w formacie z kropką, bez separatorów tysięcy np. 12345.67 tj. TO_CHAR(liczba, 'TM9')

Daty muszą być w formacie YYYY-MM-DDThh:mm:ss+HH:MM

EBS: translations (xlf)

create

ORACLE XML Publisher Zamykanie okna Preferencje Pomoc Str

Szablony | Definicje danych | Administracja

Szablony >

Tworzenie szablonu
Data początkowa i końcowa definiują okres aktywności szablonu. Nazwę można przetłumaczyć, natomiast kod jest taki sam dla wszystkich języków.

Ogólne

Personalizacja "Informacje ogólne"

* Nazwa * Kod
 * Aplikacja * Definicja danych
 * Typ RTF * Data początkowa 11-05-2011
 Data końcowa
 Szablon podrzędny

Domyślny typ wyjścia PDF

Opis

Plik szablonu

Personalizacja Stack Layout: (File)

* Plik Przeglądaj...
 * Język Angielski
 Obszar
 Do przetłumaczenia

update

Pliki szablonu

Personalizacja "Pliki szablonu"

Personalizacja Stack Layout: (Files.PreviewFormatRn)

Format podglądu PDF

Szablon do przetłumaczenia

Personalizacja "Szablon do przetłumaczenia"

Eksportowanie tłumaczenia Ładowanie tłumaczenia

Nazwa pliku	Język	Obszar	Podgląd	Pobieranie	Aktualizuj	Kasowanie
XXEAMW004.rtf	Angielski					

Dostępne tłumaczenia

Personalizacja "Dostępne tłumaczenia"

Wybór tłumaczeń i... Włączanie Wylączenie

Wybierz wszystko | Nic nie wybiera

Wybierz	Status	Język	Obszar	Podgląd	Eksportowanie tłumaczenia	Zaawansowanie
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Polski	Polska			<input checked="" type="checkbox"/> Zakończona

Napotkane problemy

Opis problemów 1,2,3,4 dotyczy raportu xml z template z plikami xlf i .rtf
 Opis problemów 4 dotyczy raportu xml z template z plikiem .rtf i (xlf i .rtf)

problem 1. Aktualizacja pliku .rtf (wykonana za pomocą formularza lub javy) nie przynosi efektu. Chodzi o to, że zmieniając cokolwiek w pliku rtf raport wynikowy wygląda cały czas tak samo. Dopiero ponowne wczytanie pliku xlf za pomocą formularza rozwiązuje problem. Nie testowałem, czy wczytanie pliku xlf za pomocą javy rozwiązuje problem.

problem 2. Załóżmy, że mamy prawidłowo działający raport z plikami .rtf i xlf. jeżeli następnie załadujemy plik .rtf za pomocą javy (nie zmieniając zawartości pliku xlf), to przestanie działać tłumaczenie. jeżeli wczytamy za pomocą formularza tłumaczenie ponownie, nadal tłumaczenie nie będzie działać ! problem rozwiązuje wczytanie pliku .rtf za pomocą formularza (a nie javy).

problem 3. Stosowanie plików .xlf wiąże się z niedogodnością polegającą na tym, że jeżeli coś zmienimy w pliku .rtf

to powinniśmy wygenerować i przetłumaczyć ponownie plik .xlf odpowiadający zmienionemu plikowi rtf.

Nowoutworzony plik .xlf znów będzie miał całą zawartość w języku angielskim- do przetłumaczenia. Mergowanie takie przeprowadziłem dzisiaj ręcznie, choć pewnie jest na świecie gdzieś narzędzie do mergowania plików (przepisz tłumaczenia ze starego do nowego).

problem 4. Wyłączenie template end-datą powoduje, że template znika z listy i nie można się już do niego w żaden sposób dostać przez formularza. Mimo tego, że jest on niewidoczny, to nie można zarejestrować ponownie template o tej samej nazwie to zniknięty template.

EBS: lexicals

This functionality is similar to "user exit" in reports.

Example 1.

How to obtain WHERE clause for gl account (segment1 between .. and .. and segment 2 between .. and .. and segment3 ..)

Search FLEX_WHERE



source: Sławek Joński.

Example 2

How to obtain item name (typically it is mtl_system_items.segment 1)

```
<lexicals>
  <lexical type="oracle.apps.fnd.flex.kff.select" name="C_FLEX_ITEM_DISP" application_short_name="INV" id_flex_code="MSTK" id_flex_num="101"
multiple_id_flex_num="N" code_combination_table_alias="MSI" segments="ALL" show_parent_segments="Y" output_type="VALUE" />
  <lexical type="oracle.apps.fnd.flex.kff.select" name="C_FLEX_CAT_DISP" application_short_name="INV" id_flex_code="MCA"
id_flex_num="P_STRUCT_NUM" multiple_id_flex_num="Y" code_combination_table_alias="MCA" segments="ALL" show_parent_segments="Y" output_type="VALUE" />
</lexicals>

(...)

SELECT
, &C_FLEX_CAT_DISP C_FLEX_CAT_DISP
, decode(prl_item_id,null,null,&C_FLEX_ITEM_DISP) C_FLEX_ITEM_DISP
FROM po_requisition_lines      prl
, mtl_system_items            msi
(...)
```

How inventory_item_id is passed ?

code_combination_table_alias="MSI" refers to mtl_system_items msi.

It is assumed that key column is inventory_item_id.

Source:

F:\MyDocuments\Work\PPL\tasks\eam\EAM_W_004 Rejestr_wnio_zak_wyst_na_us\install\fnloader-files

Summaries (podsumowania)

You can do summaries on either sql level or data template level or layout level.

See .. \XMLPublisher\examples\summaries

EBS: Zmiana ID parametrów na nazwy

```
<sqlStatement name="QPARAMS"><![CDATA[
  select
    to_char( fnd_date.canonical_to_date( :P_DATE_FROM ), 'yyyy-mm-dd' ) P_DATE_FROM_DSP
    ,xxmsz_ebs_tools.decode_valset( 'XXPERCHANGES_HIERARCHY' , :P_HIERARCHY_ID ) P_HIERARCHY_DSP
    ,xxmsz_ebs_tools.decode_valset( 'XXPERCHANGES_ORG' , :P_ORG_ID , 'MEANING' , '$FLEX$.P_STRUCTURE_VERSION_ID' ,
:P_STRUCTURE_VERSION_ID ) P_ORG_DSP
    ,xxmsz_ebs_tools.decode_valset( 'Yes_No' , :P_INCLUDE_CHILDS ) P_INCLUDE_CHILDS_DSP
  from dual
]]></sqlStatement>
```

Displaying Images in BI Publisher Reports (Repost)

I usually do not re-post but I thought this exception is worth it. With the current push of BI Publisher in the Siebel CRM community, BI Publisher knowledge is now more important than ever before. The below post (slightly redesigned) was published more than a year ago.

The following is a description of how to display a product image in a BI Publisher Report for Siebel 8.1.1.

Before Siebel 8.1 was released, BI Publisher was not the official reporting tool, but in a Siebel 8.0 project we used it anyway (at least for a POC). One of the report requirements was to display attached images (the signature of a customer that is) in the report and we were able to do that in stunningly short time.

Before we start, let me add that this post is not intended to show you how to create a new BI Publisher report in Siebel 8.1.1 as this procedure is part of the documentation and there are [courses available](#) on this topic. I also assume that you are familiar with Siebel Integration Objects and the BI Publisher Desktop.

Let's start with the Integration Object. When generating the IO you have to include the business component that carries the image information. In our case we include the BC that implements the product image.



To use the IO with BI Publisher Reports in Siebel you must use a name that starts with "BIP" (no kiddin').

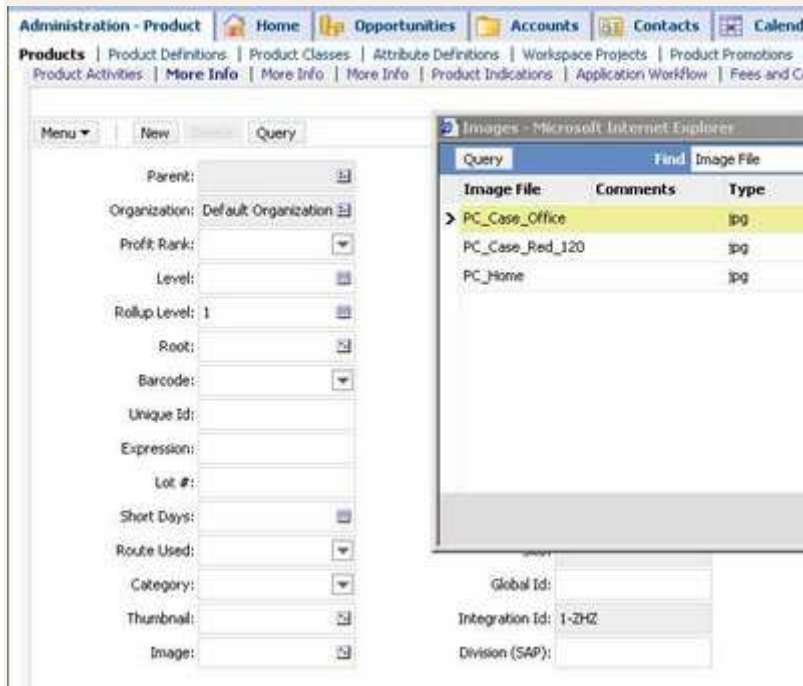
Create and compile the IO as usual. In Siebel 8.1.1 you can also deploy the IO to the database without compiling. A quick test with EAI Siebel Adapter and the Business Service Simulator shows that the IO works properly even when not compiled. Nicely done.

Question: How does EAI Siebel Adapter handle image and attachment data during query operations?

Answer: They are included in the output property set in [base64](#) encoding. The resulting XML will carry the encoded string "inline".

Let's see how we can later decode that string.

Next you register the IO in the BIP Administration Screen and generate the sample data. Make sure that you have some data available, as the sample data generation pulls the first 10 records of the respective primary BC (based on my observations, maybe ordered by ROW_ID, there is no documentation on that). In our example, I created a simple product, uploaded a jpg image file via the Administration - Document screen and associated it with the product in the More Info view.



Create the report template using BI Publisher Desktop as usual. In our example we use the popular plugin for MS Word. We create a simple report with a grouped table.

But how to decode the base64 string for the product image?

Here's the trick: Include the field that carries the base64 encoded string. In our case it's named imageFileBuffer. When you double-click a field, the BI Publisher property window is displayed. Change the code to the following:

```
<fo:instream-foreign-object content-type="image/jpg" xdofo:alt="Product Image" >
<xsl:value-of select="imageFileBuffer" /></fo:instream-foreign-object>
<fo:instream-foreign-object content-type="image/jpg" xdofo:alt="Product Image" >
<xsl:value-of select="imageFileBuffer" /></fo:instream-foreign-object>
```

This code which uses the [fo:instream-foreign-object](#) allows us to decode a base64 string back to the jpg image it previously was.

You might have to tweak the path to the base64 encoded string relative to the current object. Example:

```
<xsl:value-of select=" ../AttachmentId" />
```

This would take the AttachmentId element from two levels below the current.

Did I mention that this example works with jpg images? It's supposed to work with other image formats (gif, png) as well but jpg seems the most reliable.


A preview in BI Publisher Desktop should already show the desired outcome.

Now we can upload the rtf template in the BIP Administration screen, generate the XLIFF file and associate our report with a Siebel view.


In the view, we can do a query for the desired products, click the Report button, select the newly created report and marvel at the magic of base64.

PCS Product Catalog

PCS Advantage Desktop

Product Image	Description	Part Number
	Our flagship product. One of the fastest PCs in the world.	PCS-ADV

PCS Delta Mainboard

Product Image	Description	Part Number
	The mainboard of the 3rd millennium (for starters...)	PCS-DEL

PCS Office Desktop

Product Image	Description	Part Number
	The office workhorse. Suitable for all office work including web based apps.	PCS-OFF

[Multilingual BI Publisher Reports for Siebel CRM](#)



As indicated in the [previous post](#), I would like to share some insight in how to create BI Publisher reports for Siebel CRM in multiple languages.

As the [official documentation](#) is a bit sparse, I'll provide a simple example.

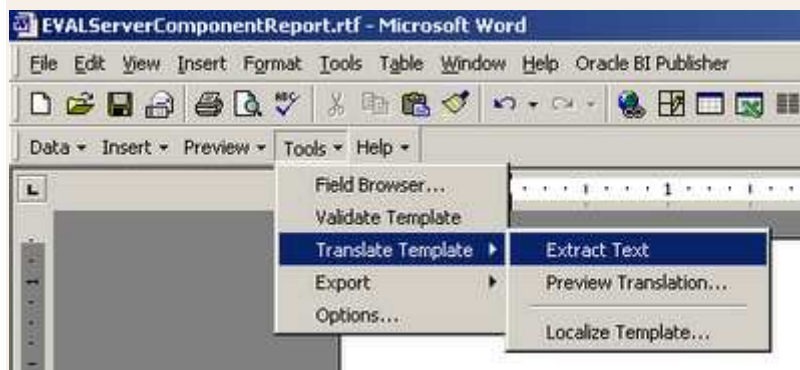
1. Create the report template

Follow the usual steps to create a BI Publisher template. It is recommendable to use English as the development language.

2. Create the XLIFF file

BI Publisher uses the [XML Localization Interchange File Format \(XLIFF\)](#) to allow translation of translatable strings in the report template to multiple languages. For each target language, we have to create one XLIFF (or .xlf) file.

In the BI Publisher Desktop (Word plugin), we can navigate to Tools | Translate Template | Extract Text to create the .xlf file for the current report template.



It is beneficial to save the file as .xlf in the xliiff/[lang] folder of the Siebel Developer Web Client's XMLP directory because this is where they have to stay for testing.

3. Translate the .xlf files.

The file created by the BI Publisher Desktop plugin must now be copied and translated once for each deployed language.

Place each copy in the language specific subfolder of the xliiff directory of the Siebel client installation. In each language specific copy we must change the target language and the translatable strings.

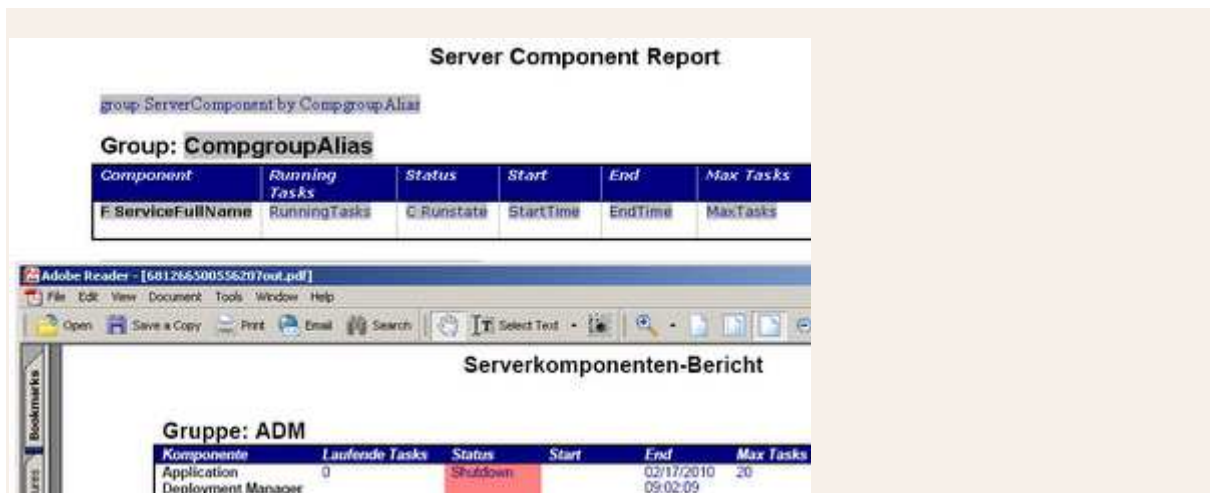


Comparison of the original .xlf file (English-US, left) and a translated copy (German, right)
Changes are highlighted.

For a list of valid language identifiers, [click here](#).

4. Test

We can use the BI Publisher Desktop to apply an .xlf file to a report template. To do so, we go to Tools | Translate Template | Preview Translation. We can then select an .xlf file and inspect the preview.



5. Deploy

The deployment of multilingual BI Publisher reports includes the following steps.

Copy the report template and the .xlf files to the respective directories on all Siebel Servers. You can also use ADM to accomplish this.

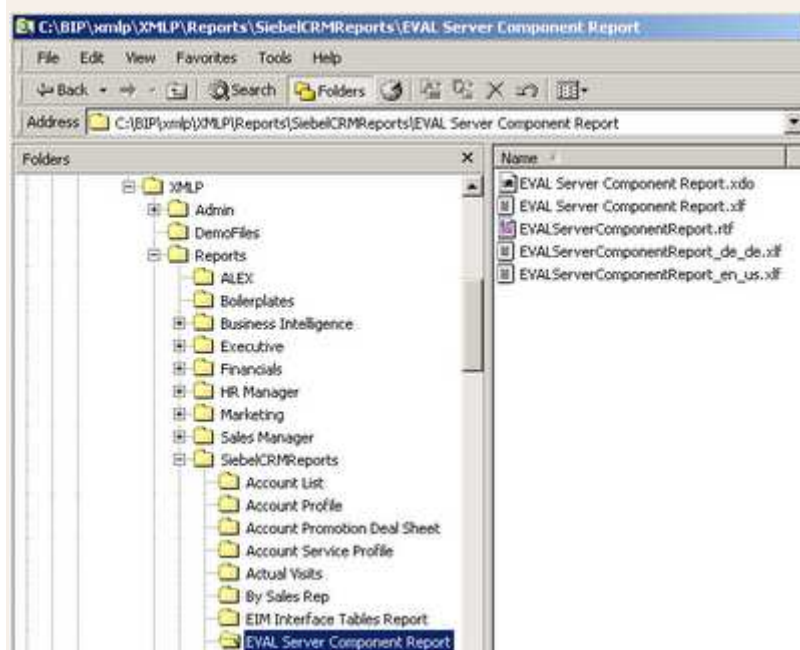
In the Siebel Web Client (connected mode), use the views in the Administration - BIP Reports screen to register the template and associate it with one or more views as usual.

You can use the Report Template Translation view to specify a display name (for the dropdown menu of the Report button) for each deployed language.

Upload the report to the BI Publisher server.

Copy the .xlf files to the BI Publisher server. The .xlf files must reside in the same report-specific folder where the uploaded .rtf template is found.

Rename the .xlf files so that the name is equal to the .rtf file, suffixed with the language identifier.



Now, you're ready to test the BIP report from each language-specific object manager

Here's a screenshot of the Server component report of the [previous post](#) in the German Siebel application.



[BI Publisher Debugging](#)

When you work a lot with BI Publisher, be it to [migrate Actuate reports](#) for Siebel CRM or for other purposes, there is a certain need to debug those reports which throw errors.

[Tim Dexter](#), who deserves all the kudos has [the answer on this](#).

I'll summarize the settings for those of you who "just need the log".

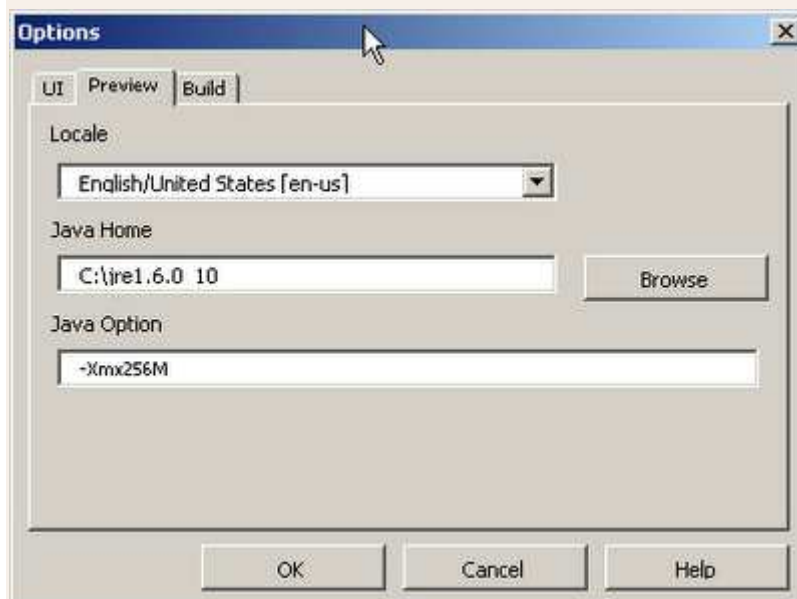
1. Create a file `xdodebug.cfg`

The file content should be similar to the following

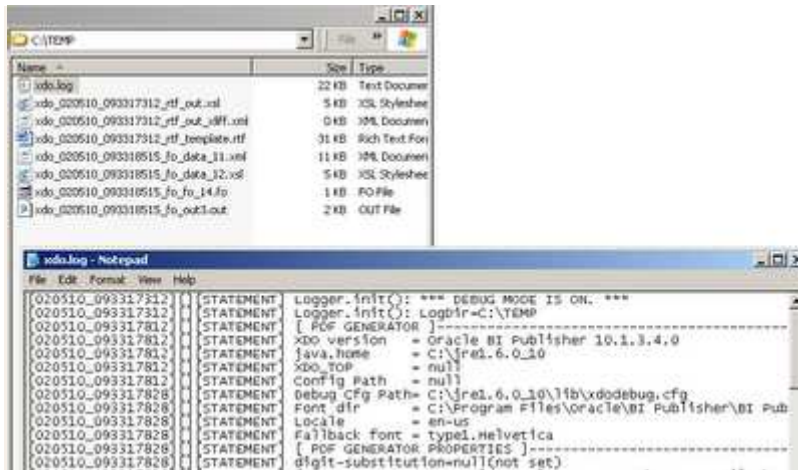
```
LogLevel=STATEMENT  
LogDir=c:\temp
```

2. Save the file in the lib directory of the JRE you are using

To answer the question which JRE you are using, open the Options dialog of the BI Publisher Desktop plugin and click the Preview tab. The JRE used by BIP Desktop is defined in the Java Home field.



Now you can for example preview a report in BI Publisher desktop. You will find an `xdo.log` and many other useful files in the directory specified in the `xdodebug.cfg` file.



http://download.oracle.com/docs/cd/E14004_01/books/Reports/ReportsTroubleshoot.html

<?choose:?>
<?when:expression?>
<?otherwise?>

--- NIE DZIAŁA---

<?copy-to-bookmark:?>

directly above your table of contents and

<?end copy-to-bookmark:?>