ORACLE®
Financial Services

Bulk Data Export using API

Oracle Financial Service Lending and Leasing

ORACLE®
PARTNERNETWORK

Disclaimer

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Table of Contents

Introduction

This document intends to explain the process of performing bulk export of structured data in machine readable format using API (Oracle BI Publisher).

Pre-requisites

Following are the pre-requisites to be noted:

» Download the required version of Oracle Financial Service Lending and Leasing release package from e-delivery.
» Ensure that BI Publisher is installed and JDBC data source is configured. The process for the same is detailed in https://docs.oracle.com/cd/E89525_01/html/BIP_Installation_Guide/BIP_Installation_GuideTOC.htm
» Ensure that BI Publisher plugin for MS word is installed. If not, follow the procedure detailed in section 'Installing BI Publisher Plugin'.

Assumptions

» It is assumed that you are familiar with basics of BI Publisher, SQL and Microsoft Word usage.

Bulk Data Export

Following are the steps to be followed while doing a bulk data export using BI Publisher:

» Identify required data for export
  » Look for column details where data exist
  » Identify relation of column data from main table
» Create Data Model
  » Create Data Set using SQL query
  » Define Parameters with variable elements
  » Export variable parameter data in XML format
  » Create reporting template using BI Publisher plugin
  » Import and upload template in BI Publisher
  » Map reporting template to Data Model in BI Publisher
  » Customize Data Output format
» Generate Data in BI Publisher
» Export BI Publisher data to required format

The above steps are explained in detail in the subsequent section using sample data to indicate data export functionality. However, since exporting data has to comply with confidentiality norms, it is recommended to contact your IT staff for assistance.

Sample Data Export

As a sample exercise, assume you need the data of portfolio companies associated with matching accounts. To know how to extract the details of the same from database, refer to the below sections.

Identify Required Data for Export

The first step in data extraction is to identify the required data and its location in the database.

Look for column details where data exist

» In the Oracle Financial Services Lending and Leasing release directory access the tech doc which is available in the path (<user directory>\LL\release\<OFSLL version>\docs\techdoc). Unzip the files and open 'Index.html'.

» Search and select the appropriate table which matches the data you need. As indicated below, searching for 'companies' displays the table PORTFOLIO_COMPANIES which has the required field information. On clicking the same, you can identify the column to which it is mapped in the database.

Identify relation of column data from main table

» In Oracle Financial Services Lending and Leasing release - tech doc directory, open the Entity Relationship Diagram available in pdf format with the name ofsll_<version>_erd.pdf. This helps you to identify the relation of the required data from the main table(s).
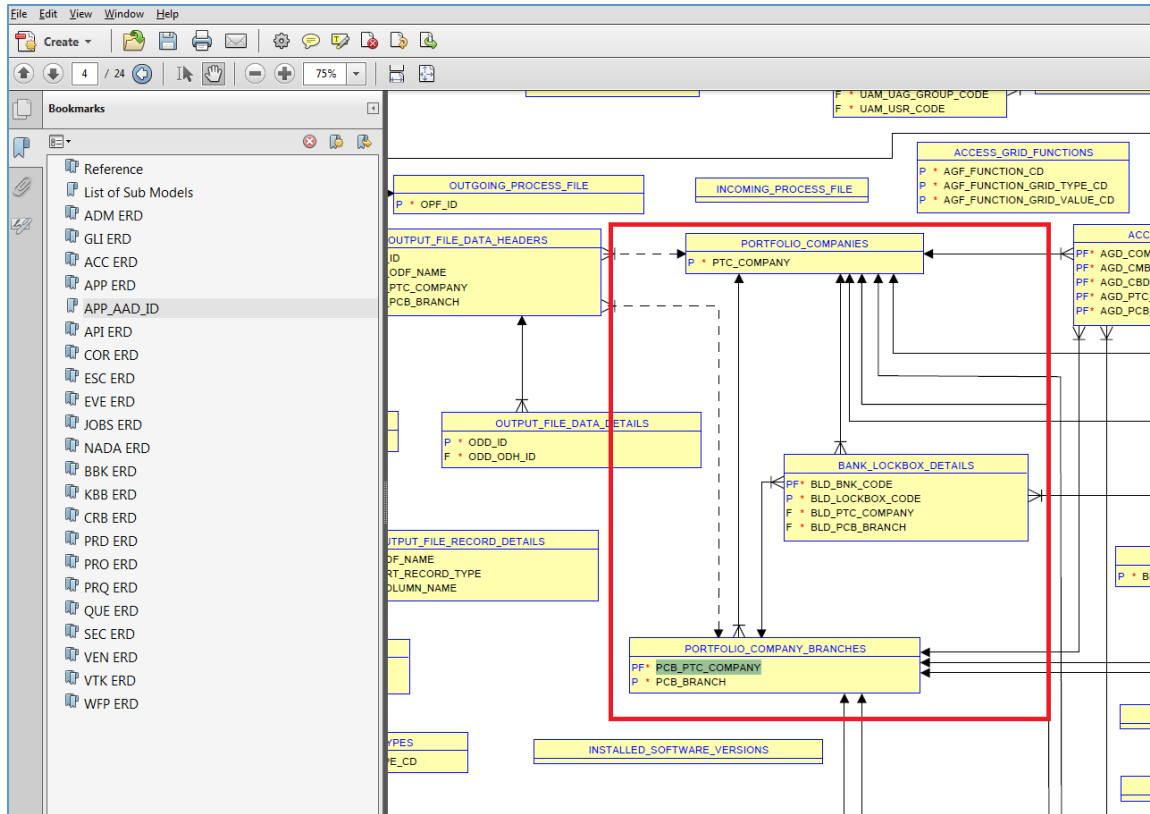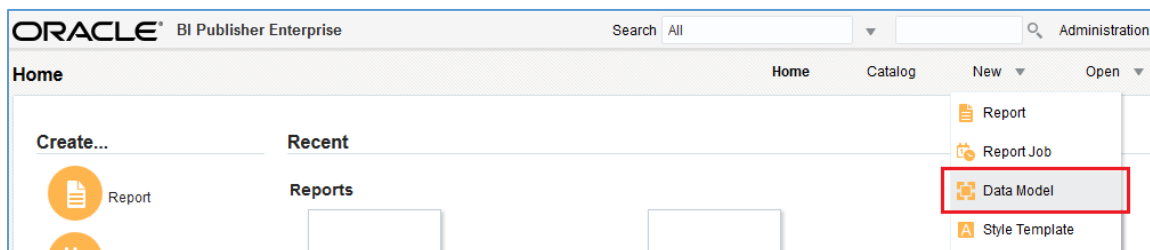

Create Data Model

Once you have the required information that needs to be extracted from database tables, open Oracle BI Publisher using which you need to create a data model with SQL query.

Create Data Set using SQL query

» In BI Publisher home page, select Data Model from 'New' drop-down list.

» Click '+' button and create a data set as indicated below. In 'Data Source' field, ensure that the same data source is used as configured during BI Publisher installation. For information, refer to the following link - https://docs.oracle.com/cd/E89525_01/html/BIP_Installation_Guide/Creating_BIPDatasource.htm
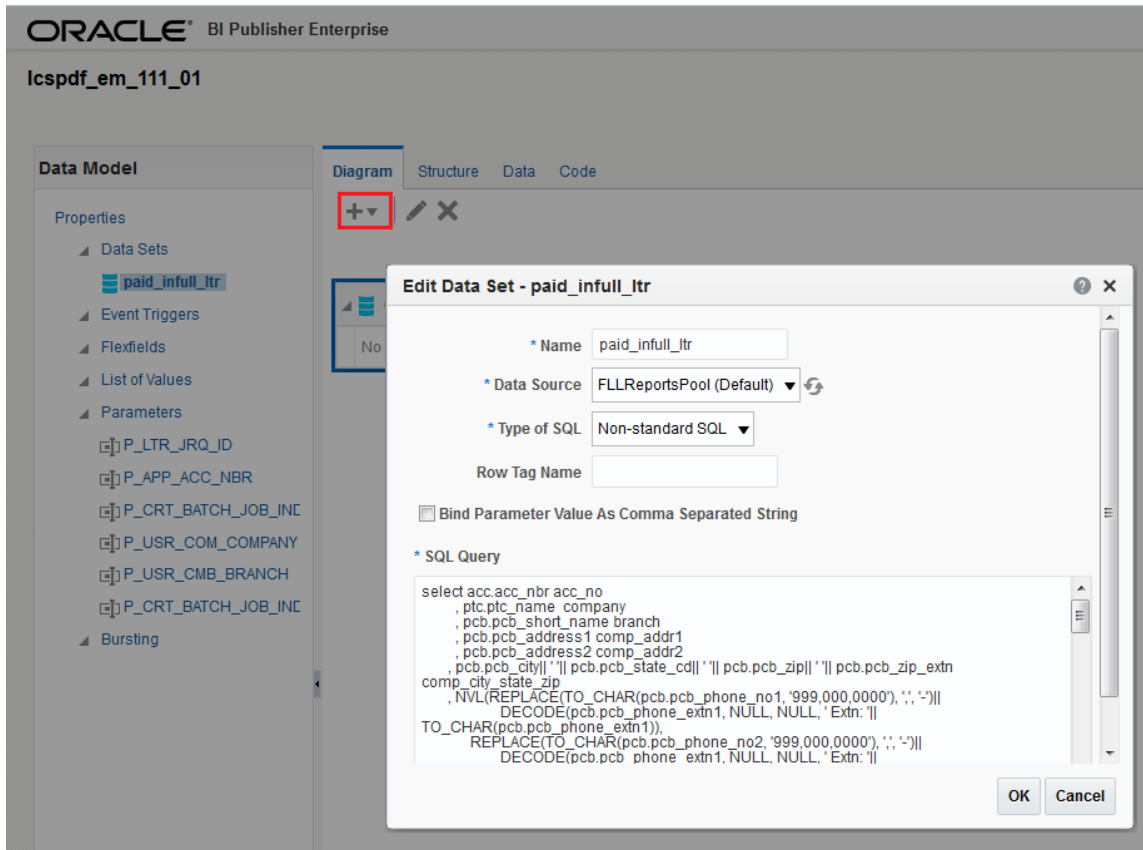
» While building the query, ensure that the three components; select, from, and where clause are defined accurately based on requirement. In the image below, the 'Select' statement indicates the elements being presented in the output, the 'From' statement indicates from where the elements are being called, and the 'Where' statement indicates the criteria on which the elements are being represented.

```
select acc.acc_nbr acc_no
       , ptc.ptc_name  company
       , pcb.pcb_short_name branch
       , pcb.pcb_address1 comp_addr1
       , pcb.pcb_address2 comp_addr2
     , pcb.pcb_city|| ' '|| pcb.pcb_state_cd|| ' '|| pcb.pcb_zip|| ' '|| pcb.pcb_zip_extn  comp_city_state_zip
     , NVL(REPLACE(TO_CHAR(pcb.pcb_phone_no1, '999,000,0000'), ',', '-')||
                 DECODE(pcb.pcb_phone_extn1, NULL, NULL, ' Extn: '|| TO_CHAR(pcb.pcb_phone_extn1)),
         REPLACE(TO_CHAR(pcb.pcb_phone_no2, '999,000,0000'), ',', '-')||
                 DECODE(pcb.pcb_phone_extn1, NULL, NULL, ' Extn: '|| TO_CHAR(pcb.pcb_phone_extn2)))    comp_phone
     , NVL(REPLACE(TO_CHAR(pcb.pcb_fax_no1, '999,000,0000'), ',', '-'),
           REPLACE(TO_CHAR(pcb.pcb_fax_no2, '999,000,0000'), ',', '-')) comp_fax
       , acc.acc_title
       , decode(nvl(acc.acc_business_ind,'N'),'Y',bus.bus_business_name,cus.cus_first_name|| ' '|| cus.cus_last_name) cus_name
       , decode(nvl(acc.acc_business_ind,'N'),'Y',bus.bus_contact_person,cus.cus_first_name) cus_first_name
       , decode(nvl(acc.acc_business_ind,'N'),'Y',bus.bua_address1,adr.adr_address1) adr_address1
       , decode(nvl(acc.acc_business_ind,'N'),'Y',bus.bua_address2,adr.adr_address2) adr_address2
       , decode(nvl(acc.acc_business_ind,'N'),'Y', bus.bua_city|| ' '|| bus.bua_state_cd|| ' '|| bus.bua_zip|| ' '||
       bus.bua_zip_extn,
             adr.adr_city|| ' '|| adr.adr_state_cd|| ' '|| adr.adr_zip|| ' '|| adr.adr_zip_extn) city_state
       , ase.ase_desc || '-' || ase.ase_identification_nbr collateral
       , acc.acc_loan_cur_currency_code
from accounts acc
       , portfolio_companies ptc
       , portfolio_company_branches pcb
       , cus_acc_relations car
       , customers cus
     , address adr
     , assets ase
     ,(select  bar_aad_id, bus_id, bus_business_name , bus_contact_person ,
         bua_address1,bua_address2,bua_city,bua_state_cd,bua_zip,bua_zip_extn
          from business_details ,business_address, bus_acc_relations
          where bus_id = bar_bus_id
          and bua_bus_id = bus_id
          and bua_current_ind = 'Y'
          and bua_mailing_ind = 'Y') bus
where ptc.ptc_company = acc.acc_ptc_company
     and pcb.pcb_ptc_company = ptc.ptc_company
     and pcb.pcb_branch = acc.acc_pcb_branch
     and ase.ase_aad_id(+) = acc.acc_aad_id
     and ase.ase_primary_ind(+) = 'Y'
     and car.car_aad_id = acc.acc_aad_id
     and car.car_relation_type_cd in ('PRIM')
     and cus.cus_id = car.car_cus_id
     and adr.adr_cus_id = cus.cus_id
     and adr.adr_current_ind = 'Y'
     and adr.adr_mailing_ind = 'Y'
     and acc.acc_aad_id = bus.bar_aad_id(+)
```

Define Parameters with Variable elements

» In BI Publisher, click 'Parameters' and define the values for each parameter that is identified as the variable element. In the above query, information for the following variables are to be generated:

   » ACC_NO – Account Number

   » COMPANY – Portfolio Company Name

   » COMP_ADDR1 - Portfolio Company Address

   » COMP_ADDR2 - Portfolio Company Address

   » COMP_PHONE - Portfolio Company Phone

   » CUS_NAME – Customer Name associated with Account

» Also, the display label specified here will be used as field name as displayed in Data tab below.
  For example: P_LTR_JRQ_ID

» Select 'Data Set' link from LHS and navigate to Data tab. Here you can specify the value for each variable parameter which is queried from database to fetch the matching records.

As indicated below, on specifying the account number in P_APP_ACC_NBR parameter and clicking on 'View', the following details are fetched from database.

ORACLE BI Publisher Enterprise

lcspdf_em_111_01

Data Model

Diagram   Structure   **Data**   Code

Properties

▲ **Data Sets**
  📄 paid_infull_ltr
▲ Event Triggers
▲ Flexfields
▲ List of Values
▲ Parameters
  P_LTR_JRQ_ID
  P_APP_ACC_NBR
  P_CRT_BATCH_JOB_IND
  P_USR_COM_COMPANY
  P_USR_CMB_BRANCH
  P_CRT_BATCH_JOB_IND
▲ Bursting

P_LTR_JRQ_ID
P_CRT_BATCH_JOB_IND N
P_USR_CMB_BRANCH OD-001

P_APP_ACC_NBR 20000100011812
P_USR_COM_COMPANY O-0001

Rows 5 ▼   **View**   **Export** | **Save As Sample Data** | **View Engine Log**

ROWSET
  ▲ ROW
    ACC_NO (20000100011812)
    COMPANY (DEMO BANK USA)
    BRANCH (USHQ)
    COMP_ADDR1 (LINE1)
    COMP_ADDR2 (LINE2)
    COMP_CITY_STATE_ZIP (MINNEAPOLIS MN 55344 7255)
    COMP_PHONE (############)
    COMP_FAX (############)
    ACC_TITLE (DE AMIT / SMITH EDWIN)
    CUS_NAME (AMIT DE)
    CUS_FIRST_NAME (AMIT)
    ADR_ADDRESS1 (745)
    ADR_ADDRESS2
    CITY_STATE (newyork MA 34038)
    COLLATERAL (2006 CHEVROLET MONTECARLO 2DR-2B4GP45G1XR378252)
    ACC_LOAN_CUR_CURRENCY_CODE (USD)

Export variable parameter data in XML format

» Save the query details clicking on 'Save As Sample Data'. Click 'Export' and export the sample data to a local directory.

Create reporting template using BI Publisher plugin

» Using the exported sample data, create template in Microsoft word using the BI Publisher plugin. For information on plugin configuration, refer to section 'Installing BI Publisher Plugin'.

» Open Microsoft Word and in the BIP tab, click 'Sample XML' and import the sample data.

» You can either use an existing template to represent the data from tables or create a temple of your choice. For instance, the below image indicates data being presented in list form using fields and subsequent image indicated data being presented in table format.
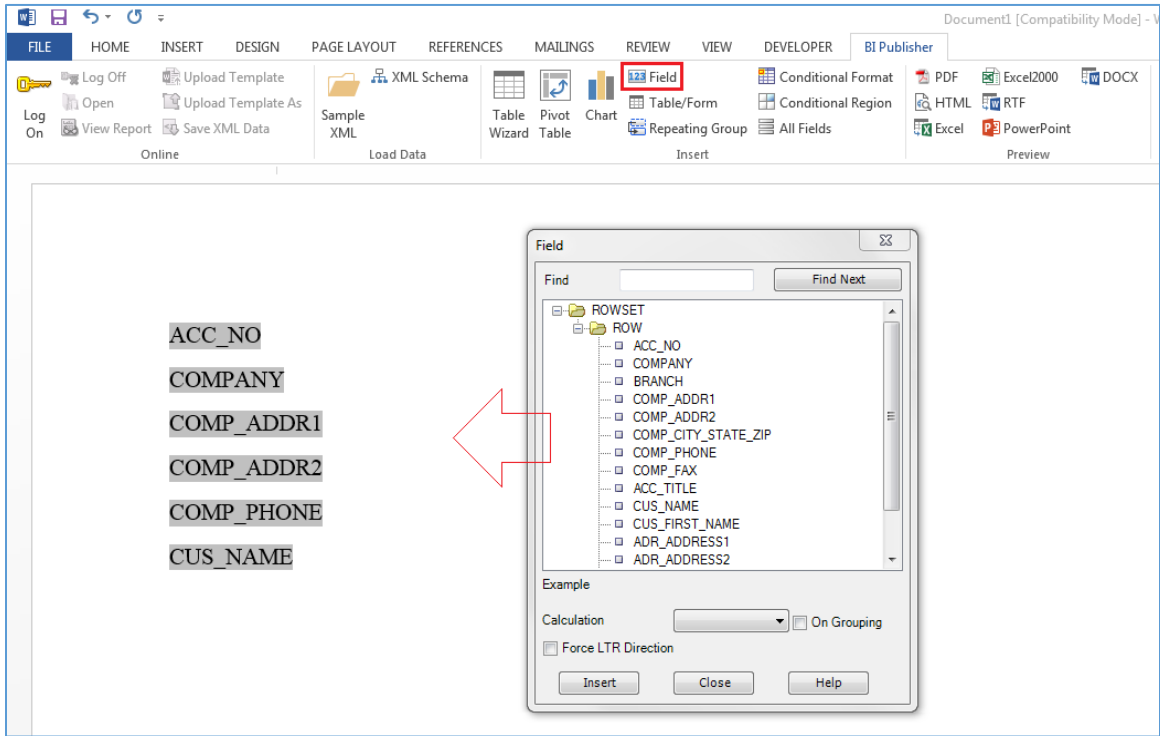
Tables also can be used when multiple records are to be extracted for same set of fields.

The BI Publisher helps to visualize on how the data from tables would be represented in the template layout. Also the BI Publisher plugin supports multiple options to create template. For information on other supported options, refer 'Template Builder for Word Tutorial' document.

» Once done, save the template in RTF (Rich text format) format. You can also choose other required format which are in common for export and import.
The common supported export formats of Microsoft Word / Excel are doc, xls, rtf, xml and html and BI Publisher supports importing files in rtf, pdf, xls, xsl formats.

Import and upload template in BI Publisher

» The next step is to import the template in BI Publisher. Select Catalog link in the LHS menu, click '+' and select 'Report'.
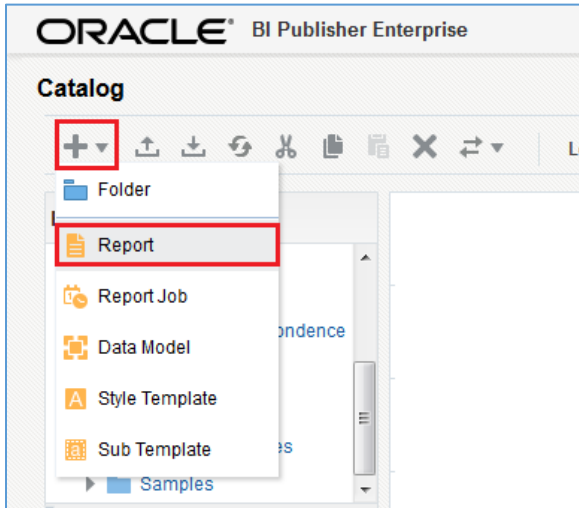
» By default, the BI Publisher interface provides you with Create Report option. Since you need to upload a pre-defined report template, click 'Cancel' and close the Create Report window.

» Select 'Upload or Generate Layout' option in the second section. In the 'Upload Template File' dialog specify the required details (Layout Name, Type, Locale), browse to the existing RTF file and click 'Upload'.
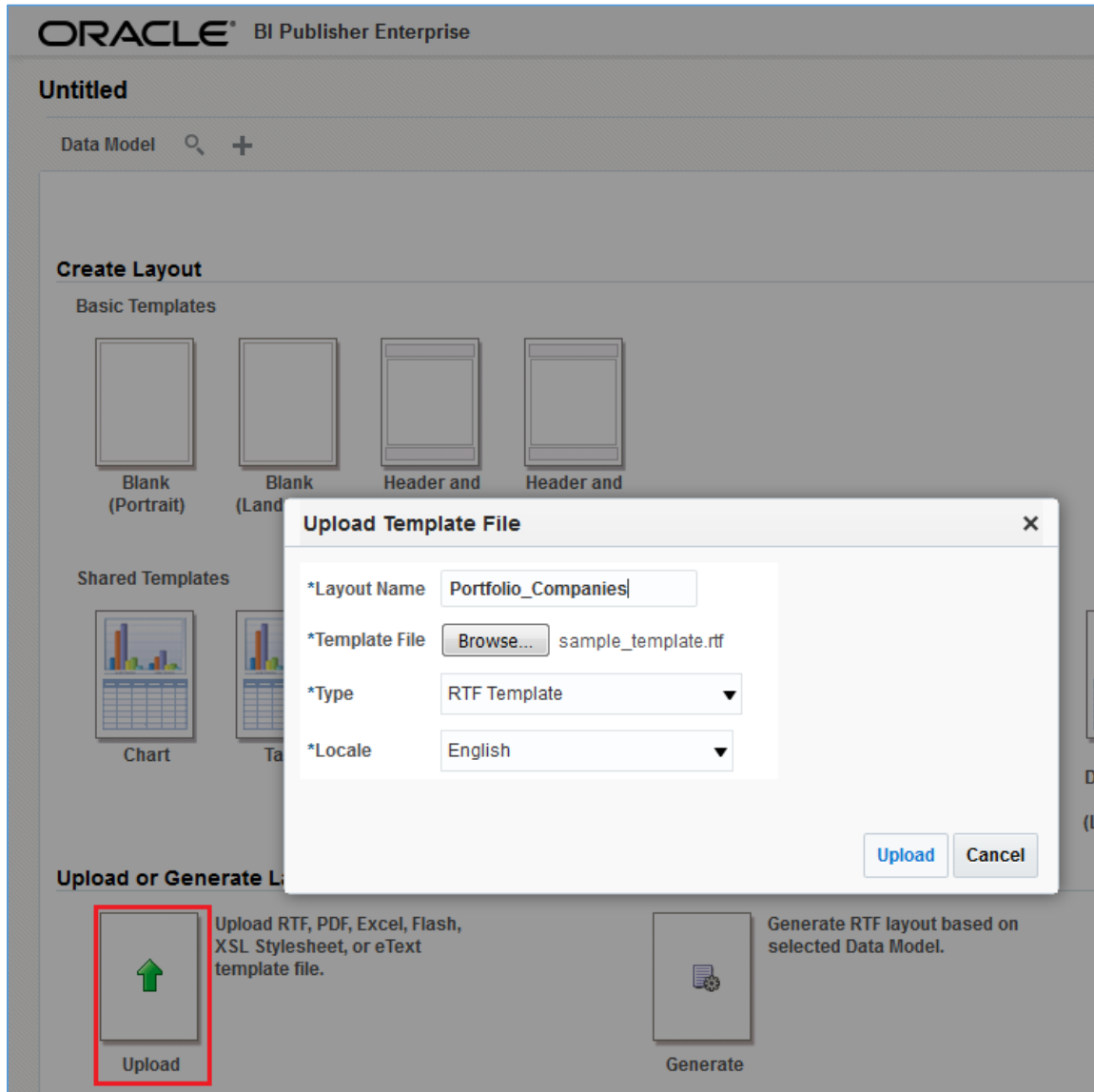
Map reporting template to Data Model in BI Publisher

» Once the template is uploaded, map the template to Data Model in BI Publisher to fetch and present the data. Click the Search icon and select the data model created.
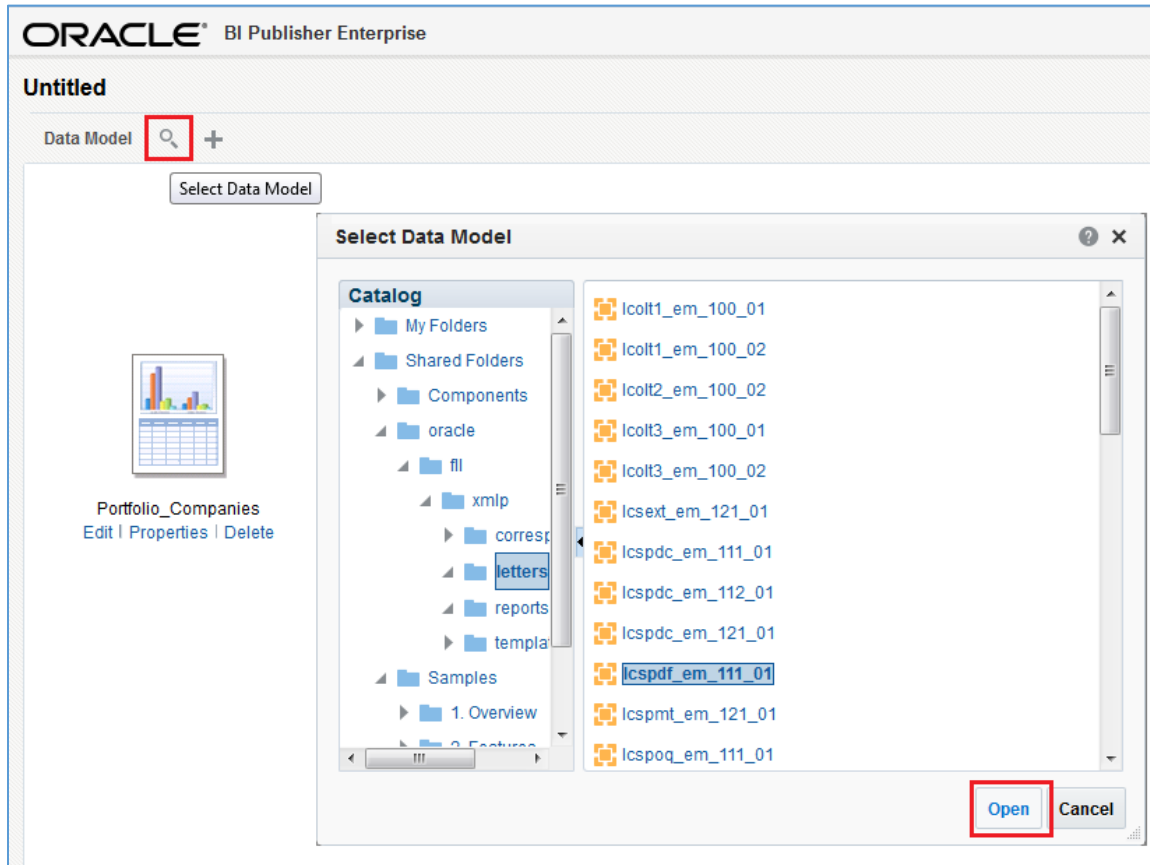
» On mapping the data model to the template, click 'Save'.
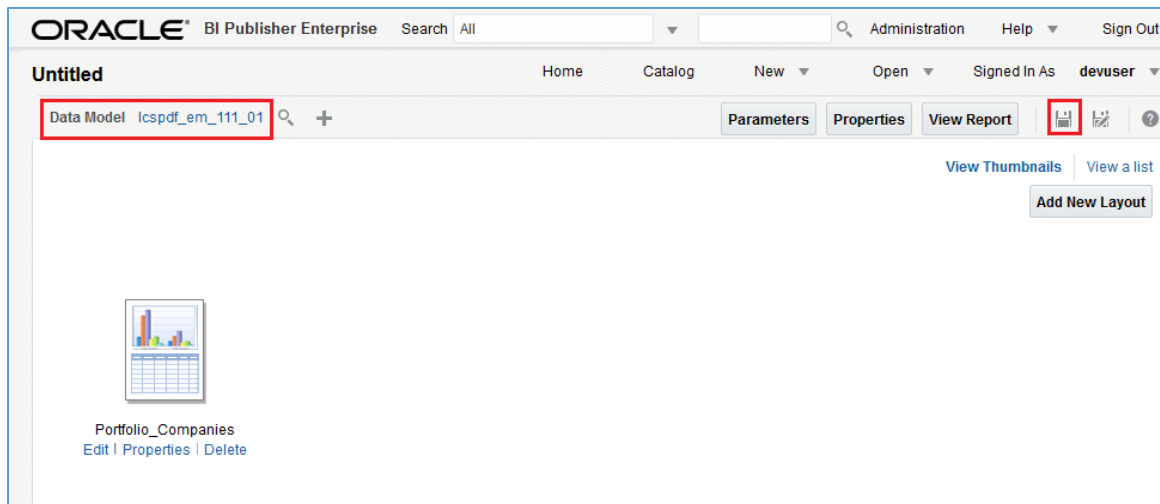
» In the Save As window, specify a name and save the mapping.
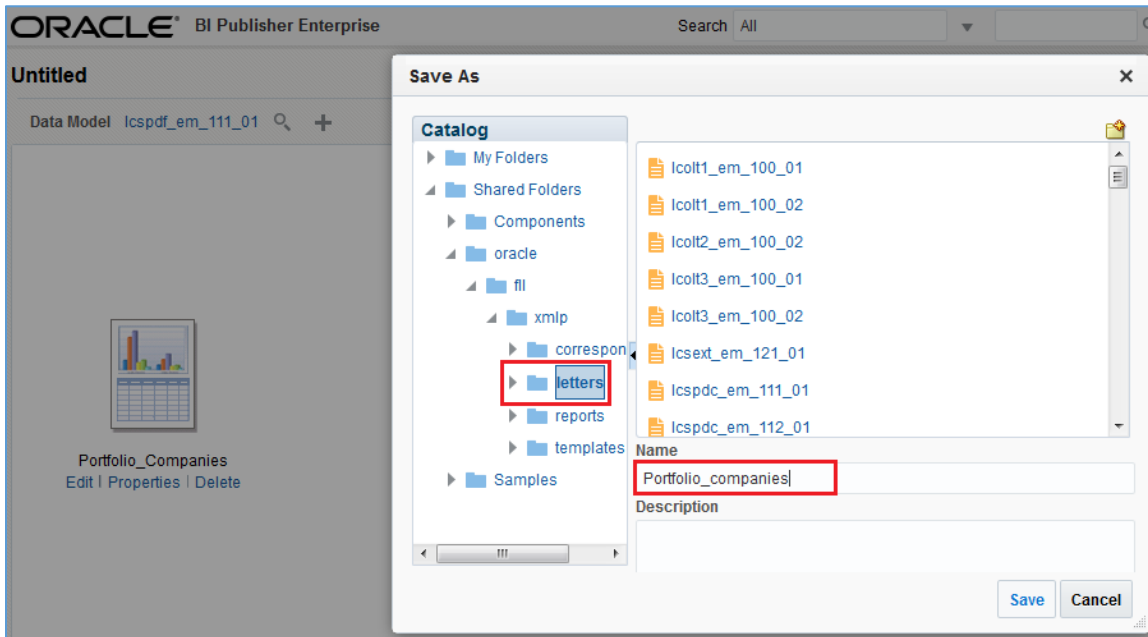
Customize Data Output format

BI Publisher by default displays the output data in pre-defined format (html or so on) and the same can be customized as per requirement. In this case a machine readable output such as xml or csv is required, and hence the default format is to be set.

» Click 'View a List' option and select the required formats (XML, CSV and so on) in 'Output Formats' drop-down list. If you are in the reporting format screen, select [gear icon] and click 'Edit Report' to view this screen.
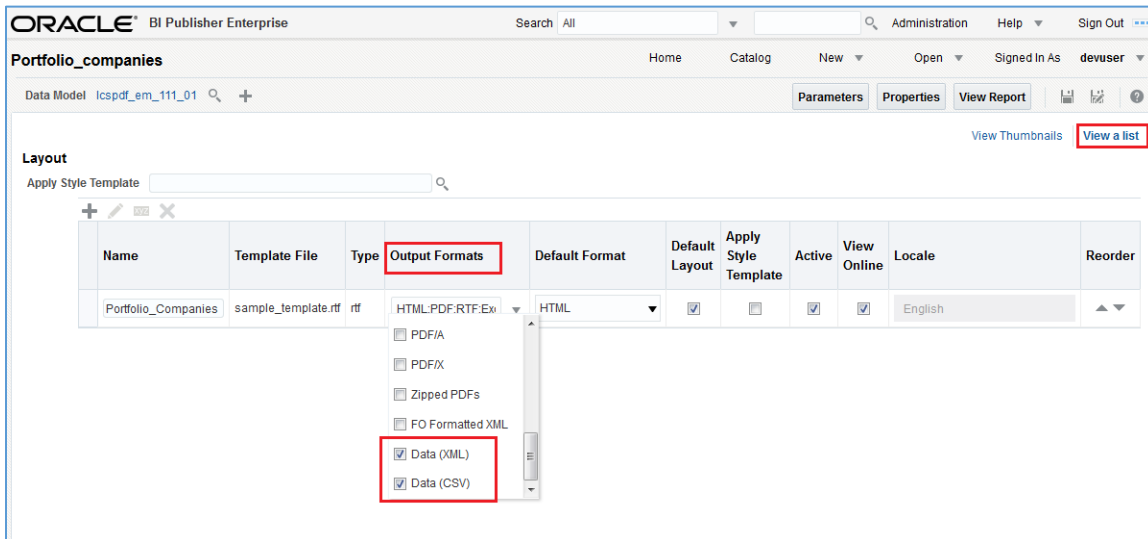
» Also select the 'Default Format' (as XML / CSV) in which the output data is to be represented and 'Save' the details.
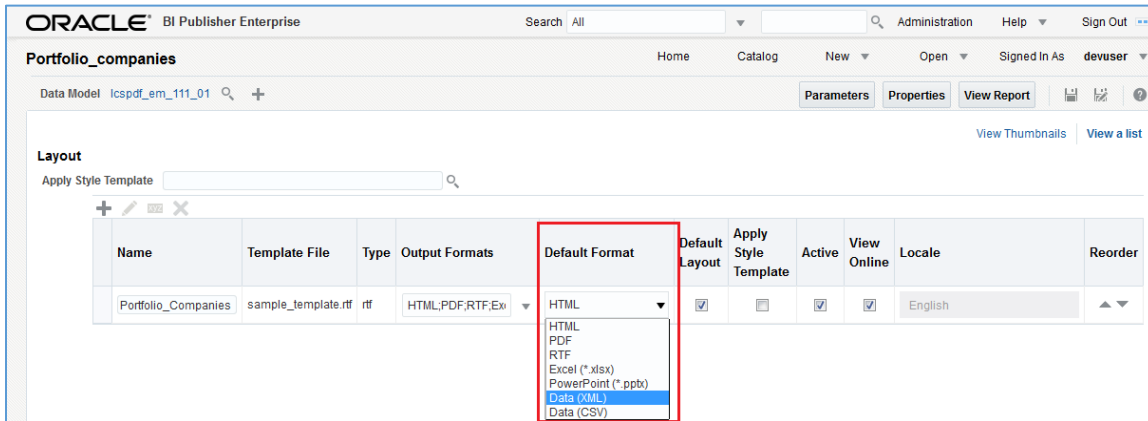
Generate Data in BI Publisher

» Using the search field at the top, enter the name of report and click search.
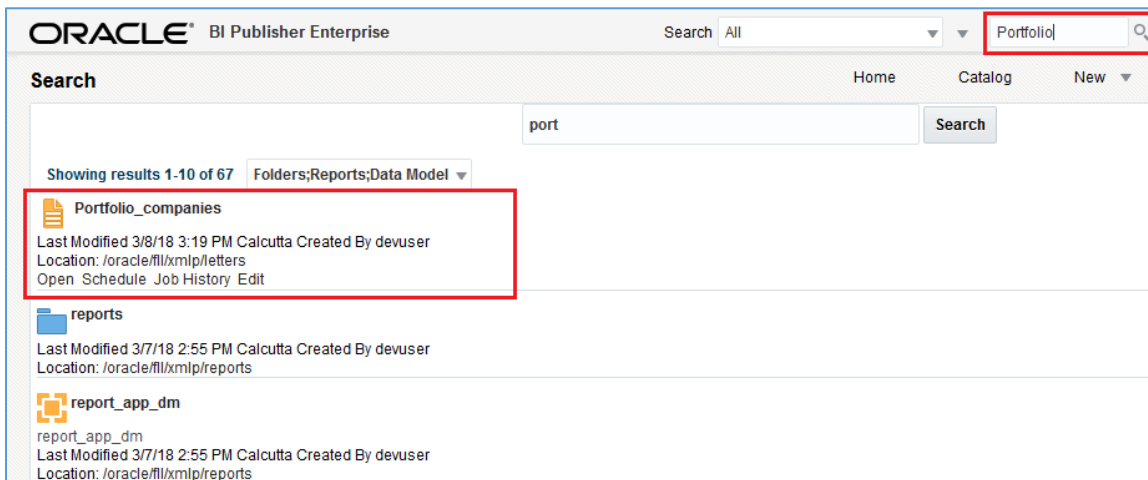
» Click on the matching report in search results. BI Publisher by default displays a blank report since parameter values are not yet provided.

» Specify the values for the defined parameters to fetch the corresponding details from the database into the report. Once done, click Apply. The report is generated in XML format (default) with the matching records from the database as indicated below.

» Also on selecting CSV from the 'View Report' option, the output generated in CSV (comma separated values) format as indicated below:

Export BI Publisher data to required format

» On generating the report with required data, you can export the same to the following BI Publisher supported formats. Click Actions > Export and select the required format.

  » HTML
  » PDF
  » RTF
  » Excel
  » PowerPoint
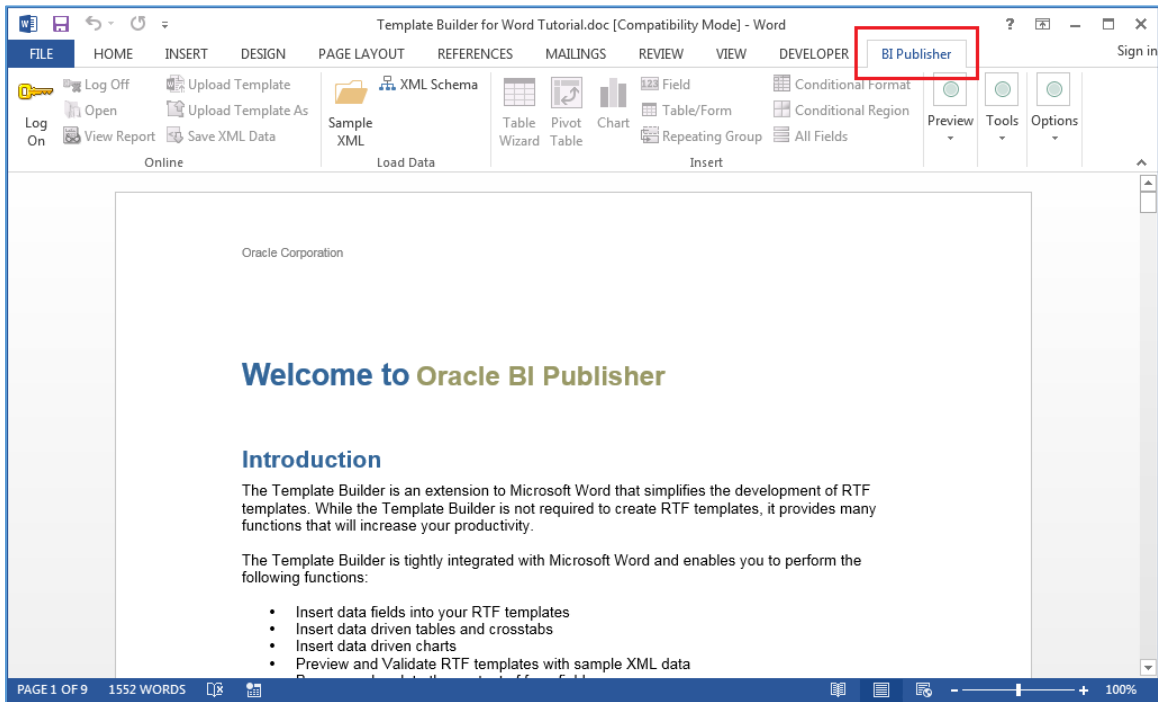 » Data (XML format)
  » CSV (comma separated values format)
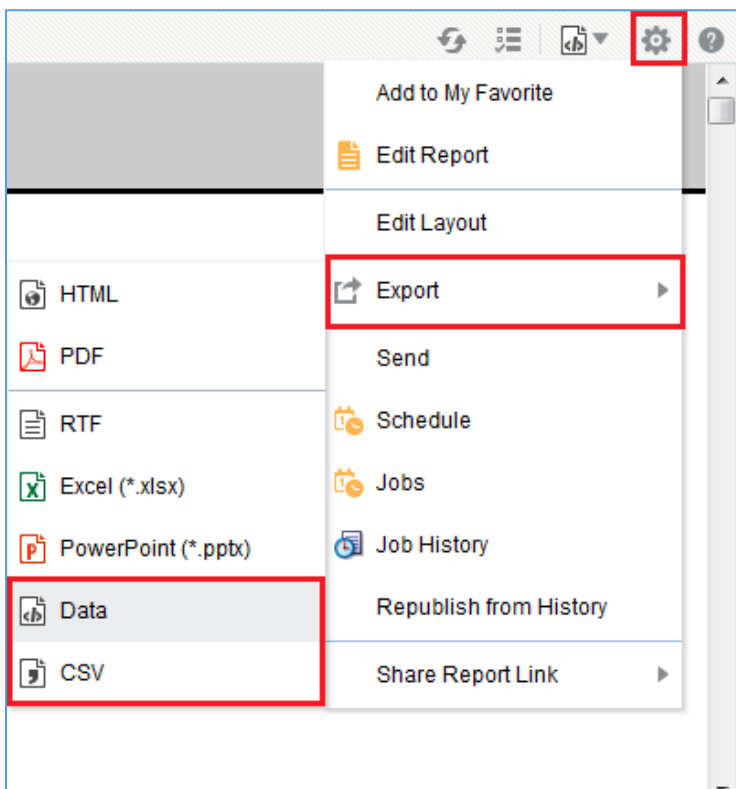
Installing BI Publisher Plugin

1. In the BI Publisher home page, click Help > Download BI Publisher Tools > BI Publisher Desktop (32/64 bit Office) option.
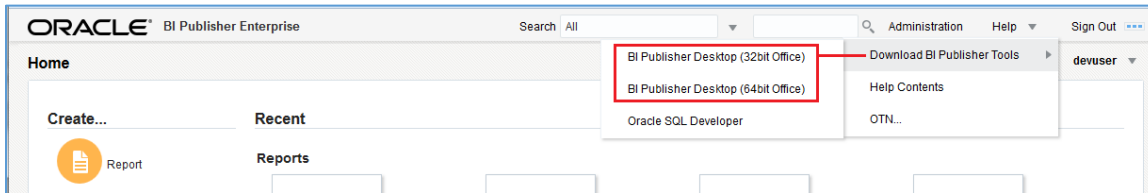
2. Save the binary file on to local directory and install the plugin.

3. Once done, verify the BI Publisher tab in the Menu by opening a new MS word document.

Oracle Corporation, World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries
Phone: +1.650.506.7000
Fax: +1.650.506.7200

Integrated Cloud Applications & Platform Services

Oracle is committed to developing practices and products that help protect the environment