

Skip Headers

Oracle Fusion Middleware Data Modeling Guide for Oracle Business Intelligence Publisher
Release 11g (11.1.1)
Part Number E18862-01

  
Contents Previous Next

Adding Bursting Definitions

This chapter covers the following topics:

- About Bursting
- What is the Bursting Definition?
- Adding a Bursting Definition to Your Data Model
- Defining the Query for the Delivery XML
- Defining the Split By and Deliver By Elements for a CLOB/XML Data Set
- Configuring a Report to Use a Bursting Definition
- Sample Bursting Query
- Creating a Table to Use as a Delivery Data Source

About Bursting

Bursting is a process of splitting data into blocks, generating documents for each block, and delivering the documents to one or more destinations. The data for the report is generated by executing a query once and then splitting the data based on a "Key" value. For each block of the data, a separate document is generated and delivered.

Using BI Publisher's bursting feature you can split a single report based on an element in the data model and deliver the report based on a second element in the data model. Driven by the delivery element, you can apply a different template, output format, delivery method, and locale to each split segment of your report. Example implementations include:

- Invoice generation and delivery based on customer-specific layouts and delivery preference
- Financial reporting to generate a master report of all cost centers, splitting out individual cost center reports to the appropriate manager
- Generation of pay slips to all employees based on one extract and delivered via e-mail

What is the Bursting Definition?

A bursting definition is a component of the data model. After you have defined the data sets for the data model, you can set up one or more bursting definitions. When you set up a bursting definition, you define the following:

- The **Split By** element is an element from the data that will govern how the data is split. For example, to split a batch of invoices by each invoice, you may use an element called CUSTOMER_ID. The data set must be sorted or grouped by this element.
- The **Deliver By** element is the element from the data that will govern how formatting and delivery options are applied. In the invoice example, it is likely that each invoice will have delivery criteria determined by customer, therefore the Deliver By element may also be CUSTOMER_ID.
- The **Delivery Query** is a SQL query that you define for BI Publisher to construct the delivery XML data file. The query must return the formatting and delivery details.

Adding a Bursting Definition to Your Data Model

Prerequisites:

- You have defined the data set for this data model
- The data set is sorted or grouped by the element by which you want to split the data in your bursting definition
- The delivery and formatting information is available to BI Publisher. The information can be provided at runtime to BI

Publisher in one of the following ways:

- The information is stored in a database table available to BI Publisher (for a dynamic delivery definition)
- The information is hard coded in the delivery SQL (for a static delivery definition)
- The report definition for this data model has been created and includes the layouts that will be applied to the report data.

To add a bursting definition

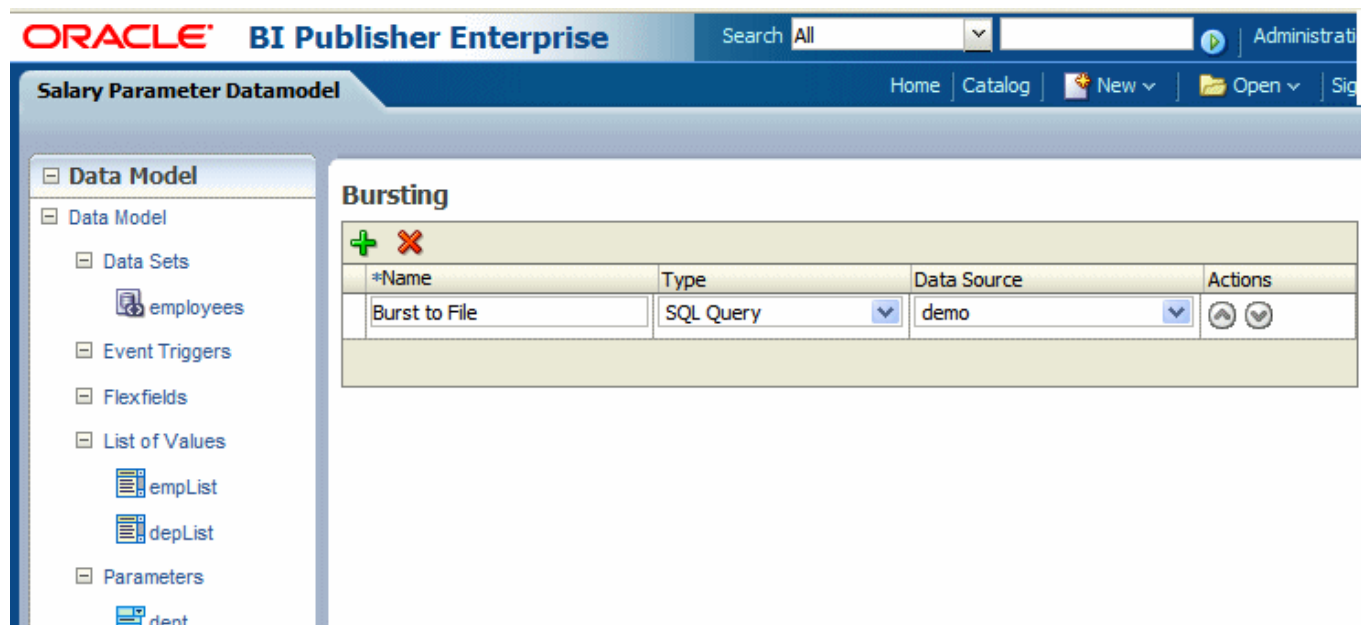
1. On the component pane of the data model editor, click **Bursting**.
2. On the Bursting definition table, click the **Create new Bursting** button.
3. Enter the following for this bursting definition:

Name - for example, "Burst to File"

Type- SQL Query is currently the only supported type

Data Source - select the data source that contains your delivery information

The following figure shows a Bursting definition:



4. In the lower region, enter the following for this bursting definition:

Split By - select the element from the data set by which to split the data

Deliver By - select the element from the data set by which to format and deliver the data

Note: If the Split By and Deliver By elements reside in an XML document stored as a CLOB in your database, you must enter the full XPATH in the Split By and Delivery By fields. For more information, see Defining the Split By and Deliver By Elements for a CLOB/XML Data Set.

SQL Query - enter the query to construct the delivery XML. For information on how to construct the bursting query, see Defining the Query for the Delivery XML.

The screenshot shows the Oracle BI Publisher Enterprise web interface. The top navigation bar includes 'Administration', 'Help', and 'Sign Out'. The main header displays 'Salary Parameter Datamodel' and 'Signed In As administrator'. The left sidebar shows a tree view of the 'Data Model' with categories like 'Data Sets', 'Event Triggers', 'Flexfields', 'List of Values', 'Parameters', and 'Bursting'. Under 'Bursting', 'Burst to Email' is highlighted.

The main content area is titled 'Bursting' and contains a table with the following data:

Name	Type	Data Source	Actions
Burst to File	SQL Query	demo	[Up] [Down]

Below the table, the 'Burst to File' configuration is shown. It includes dropdowns for 'Split By' and 'Deliver By', both set to '/ROWSET/ROW/DEPARTMENT_NAME'. There is a 'SQL Query' section with a 'Query Builder' button. The SQL query is displayed in a text area:

```
select "CUSTOMERS"."CUSTOMER_ID" as "KEY",
"CUSTOMERS"."CST_TEMPLATE" as TEMPLATE,
"CUSTOMERS"."CST_LOCALE" as LOCALE,
"CUSTOMERS"."CST_FORMAT" as OUTPUT_FORMAT,
"EMAIL" as DEL_CHANNEL,
"CUSTOMERS"."CUST_EMAIL" as PARAMETER1,
'accounts.receivable@orade.com' as PARAMETER2,
'bip-collections@orade.com' as PARAMETER3,
'Your Invoices' as PARAMETER4,
'Hi'||CUST_FIRST_NAME||chr(13)|| 'Please find attached your invoices.' as PARAMETER5,
'true' as PARAMETER6,
'donotreply@mycompany.com' as PARAMETER7
from "OE"."CUSTOMERS" "CUSTOMERS"
```

Defining the Query for the Delivery XML

The bursting query is a SQL query that you define to provide BI Publisher with the required information to format and deliver the report. BI Publisher uses the results from the bursting query to create the delivery XML.

The BI Publisher bursting engine uses the delivery XML as a mapping table for each Deliver By element. The structure of the delivery XML required by BI Publisher is as follows:

```
<ROWSET>
<ROW>
  <KEY></KEY>
  <TEMPLATE></TEMPLATE>
  <LOCALE></LOCALE>
  <OUTPUT_FORMAT></OUTPUT_FORMAT>
  <DEL_CHANNEL></DEL_CHANNEL>
  <TIMEZONE></TIMEZONE>
  <CALENDAR></CALENDAR>
  <OUTPUT_NAME></OUTPUT_NAME>
  <SAVE_OUTPUT></SAVE_OUTPUT>
  <PARAMETER1></PARAMETER1>
  <PARAMETER2></PARAMETER2>
  <PARAMETER3></PARAMETER3>
  <PARAMETER4></PARAMETER4>
  <PARAMETER5></PARAMETER5>
  <PARAMETER6></PARAMETER6>
  <PARAMETER7></PARAMETER7>
```

```

<PARAMETER8></PARAMETER8>
<PARAMETER9></PARAMETER9>
<PARAMETER10></PARAMETER10>
</ROW>
</ROWSET>

```

where

- KEY is the Delivery key and must match the **Deliver By** element. The bursting engine uses the key to link delivery criteria to a specific section of the burst data.
- TEMPLATE - is the name of the Layout to apply. Note that the value is the Layout name (for example, 'Customer Invoice'), not the template file name (for example, invoice.rtf).
- LOCALE - is the template locale, for example, 'en-US'.
- OUTPUT_FORMAT - is the output format. For a description of each type, see Setting the Output Types, *Oracle Fusion Middleware Report Designer's Guide for Oracle Business Intelligence Publisher*. Valid values are:
 - HTML
 - PDFZ - zipped PDF
 - PDF
 - RTF
 - EXCEL
 - EXCEL2000
 - MHTML
 - PPT
 - PPTX - PowerPoint 2007
 - XSLFO
 - XML
 - CSV
 - ETEXT
- SAVE_OUTPUT - indicates whether to save the output documents to BI Publisher history tables that the output can be viewed and downloaded from the **Report Job History** page.

Valid values are 'true' (default) and 'false'. If this property is not set, the output will be saved.

- DEL_CHANNEL - is the delivery method. Valid values are:
 - EMAIL
 - FAX
 - FILE
 - FTP
 - PRINT
 - WEBDAV

- **TIMEZONE** - is the time zone to use for the report. Values must be in the Java format, for example: 'America/Los_Angeles'. If time zone is not provided, the system default time zone will be used to generate the report.
- **CALENDAR** - is the calendar to use for the report. Valid values are:
 - GREGORIAN
 - ARABIC_HIJRAH
 - ENGLISH_HIJRAH
 - JAPANESE_IMPERIAL
 - THAI_BUDDHA
 - ROC_OFFICIAL (Taiwan)

If not provided, the value 'GREGORIAN' will be used.

- **OUTPUT_NAME** - is the name that will be assigned to the output file in the report job history.
- Delivery parameters by channel. The values required for the parameters depend on the delivery method chosen. The parameter values mappings for each method are shown in the following table. Not all delivery channels use all the parameters.

Delivery Channel	PARAMETER Values
Email	PARAMETER1: Email address PARAMETER2: cc PARAMETER3: From PARAMETER4: Subject PARAMETER5: Message body PARAMETER6: Attachment value ('true' or 'false'). If your output format is PDF, you must set this parameter to "true" to attach the PDF to the e-mail. PARAMETER7: Reply-To PARAMETER8: Bcc (PARAMETER 9-10 are not used)
Printer	PARAMETER1: Printer group PARAMETER2: Printer name or for a printer on CUPS, the printer URI, for example: ipp://myserver.com:631/printers/printer1 PARAMETER3: Number of Copies PARAMETER4: Sides. Valid values are: <ul style="list-style-type: none"> • "d_single_sided" for single-sided • "d_double_sided_l" for duplex/long edge • "d_double_sided_s" for tumble/short edge If the parameter is not specified, single-sided will be used. PARAMETER5: Tray. Valid values are: <ul style="list-style-type: none"> • "t1" for "Tray 1" • "t2" for "Tray 2" • "t3" for "Tray 3" If not specified, the printer default will be used. PARAMETER6: Print range. For example "3" will print page 3 only, "2-5" will print pages 2-5, "1,3-5" will print pages 1 and 3-5 (PARAMETER 7-10 are not used)
Fax	PARAMETER1: Fax server name PARAMETER2: Fax number (PARAMETER 3-10 are not used)
WebDAV	PARAMETER1: Server Name

	PARAMETER2: Username PARAMETER3: Password PARAMETER4: Remote Directory PARAMETER5: Remote File Name PARAMETER6: Authorization type, values are 'basic' or 'digest' (PARAMETER 7-10 are not used)
File	PARAMETER1: Directory PARAMETER2: File Name (PARAMETER 3-10 are not used)
FTP and SFTP	PARAMETER1: Server name PARAMETER2: Username PARAMETER3: Password PARAMETER4: Remote Directory PARAMETER5: Remote File Name PARAMETER6: Secure (set this value to 'true' to enable Secure FTP) (PARAMETER 7-10 are not used)

Passing a Parameter to the Bursting Query

You can pass the value for an element of your bursting XML using a parameter defined in the data model. For example, if you want to be able to select the template at the time of submission, you can define a parameter in the data model and use the :parameter_name syntax in your query. The following example demonstrates this use case of a parameter in a bursting query.

Your report definition includes three layouts: layout1, layout2, and layout3. At submission time you want to select the layout (TEMPLATE) to use. In your data model, define a list of values with the layout names. The following figure shows a data model with the layout list of values:

The screenshot shows the 'my data model' interface. On the left is a navigation pane with the following items: Data Model, Data Model (expanded), Data Sets (containing 'employees'), Event Triggers, Flexfields, List of Values (containing 'template'), Parameters, and Bursting. The 'template' item is selected. The main area displays the 'List of Values' for 'template'. It shows a table with columns: Name, Type, Data Source, and Reorder. The 'Name' column contains 'template', 'Type' is 'Fixed Data', and 'Data Source' is empty. Below this, there is a section titled 'template: Type: Fixed Data' which contains a table with two columns: Label and Value. The 'Label' column contains 'layout1', 'layout2', and 'layout3'. The 'Value' column contains 'layout1', 'layout2', and 'layout3'.

Name	Type	Data Source	Reorder
template	Fixed Data		

Label	Value
layout1	layout1
layout2	layout2
layout3	layout3

Next create a menu type parameter, here named P1:

my data model

Data Model

- Data Model
 - Data Sets
 - employees
 - Event Triggers
 - Flexfields
 - List of Values
 - template
 - Parameters
 - P1**
 - Bursting

Parameters

*Name	Data Type	Default Value	Parameter Type	Reorder
P1	String	layout1	Menu	

P1: Type: Menu

Display Label:

List of Values:

Options:
 ☐ Multiple Selection
☐ Can select all
☐ Refresh other parameters on change

In the bursting query, pass the parameter value to the TEMPLATE field using :P1 as shown in the following figure:

my data model

Data Model

- Data Model
 - Data Sets
 - employees
 - Event Triggers
 - Flexfields
 - List of Values
 - template
 - Parameters
 - P1
 - Bursting**
 - Burst to Email**

Bursting

*Name	Type	Data Source	Actions
Burst to Email	SQL Query	demo	

Burst to Email

Split By:

Deliver By:

SQL Query Query Builder


```

select
  MANAGER_ID KEY,
  :P1 TEMPLATE,
  'en-US' LOCALE,
  'PDF' OUTPUT_FORMAT,
  'EMAIL' DEL_CHANNEL,
  email PARAMETER1,
  'nobody@orade.com' PARAMETER2,
  'from@nowhere.com' PARAMETER3,
  'SUBJECT: Employee Salary Report' PARAMETER4,
  'BODY: Attached is a list of your employees.' PARAMETER5,
  'true' PARAMETER6,
  'replyto@nowhere.com' PARAMETER7
from
  employees
  
```

Defining the Split By and Deliver By Elements for a CLOB/XML Data Set

If the split-by and deliver-by elements required for your bursting definition reside in a data set retrieved from a CLOB column in a database, BI Publisher cannot parse the XML to present the elements in the **Split By** and **Deliver By** lists. You therefore must manually enter the XPath to locate each element in the retrieved XML data set. To ensure that you enter the path correctly, use the data model editor's **Get XML Output** feature to view the XML that will be generated by the data engine.

For example, the following sample XML was stored in a CLOB column in the database called "XMLTEXT", and extracted as an XML data set:



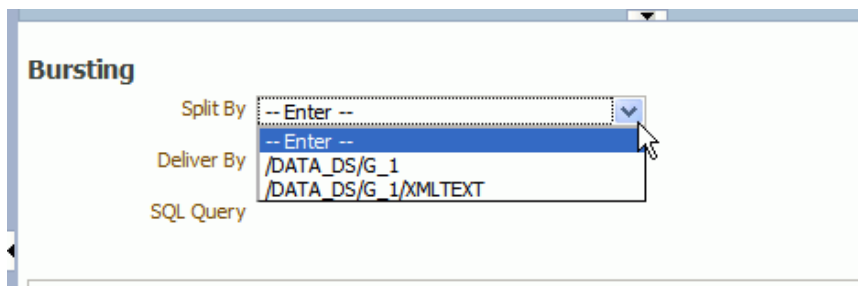
```

<!--Generated by Oracle BI Publisher -->
-<DATA_DS>
  -<G_1>
    -<XMLTEXT>
      -<DATA_DS>
        -<G_1>
          <DEPARTMENT_ID>10</DEPARTMENT_ID>
          <DEPARTMENT_NAME>Administration</DEPARTMENT_NAME>
          <MANAGER_ID>200</MANAGER_ID>
          <LOCATION_ID>1700</LOCATION_ID>
        -<G_2>
          <EMPLOYEE_ID>200</EMPLOYEE_ID>
          <FIRST_NAME>Jennifer</FIRST_NAME>
          <LAST_NAME>Whalen</LAST_NAME>
          <EMAIL>JWHALEN</EMAIL>
          <PHONE_NUMBER>515.123.4444</PHONE_NUMBER>
          <HIRE_DATE>1987-09-17T00:00:00.000-07:00</HIRE_DATE>
          <JOB_ID>AD_ASST</JOB_ID>
          <SALARY>4400</SALARY>
          <MANAGER_ID_1>101</MANAGER_ID_1>
          <DEPARTMENT_ID_1>10</DEPARTMENT_ID_1>

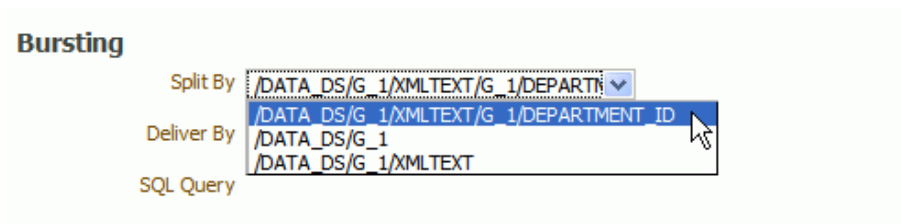
```

For this example, you want to add a bursting definition with split by and deliver by element based on the DEPARTMENT_ID, which is an element within the CLOB/XML data set.

When you add the bursting definition, the Split By and Deliver By lists cannot parse the structure beneath the XMLTEXT element:



To use the DEPARTMENT_ID element as the Split By element, type the XPath into the field as shown:



Configuring a Report to Use a Bursting Definition

Although you can define multiple bursting definitions for a single data model, you can enable only one for a report.

Enable a report to use a bursting definition on the **Report Properties** dialog of the report editor. For more information see *Configuring Report Properties, Oracle Fusion Middleware Report Designer's Guide for Oracle Business Intelligence Publisher*.

After you configure the report to use the bursting definition, when you schedule a job for this report you can choose to use the bursting definition to format and deliver the report. For more information see *Creating a Bursting Job, Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Publisher*.

Note that you can also opt not to use the bursting definition and choose your own output and destination as a regular scheduled report.

Sample Bursting Query

The following example is based on an invoice report. This report is to be delivered by CUSTOMER_ID to each customer's individual e-mail address.

This example assumes that the delivery and formatting preferences for each customer are contained in a database table named "CUSTOMERS". The CUSTOMERS table includes the following columns that will be retrieved to create the delivery XML dynamically at runtime:

- CST_TEMPLATE, CST_LOCALE, CST_FORMAT, CST_EMAIL_ADDRESS

The SQL to generate the delivery data set for this example is as follows:

```
select distinct
CUSTOMER_ID as "KEY",
CST_TEMPLATE TEMPLATE,
CST_LOCALE LOCALE,
CST_FORMAT OUTPUT_FORMAT,
'EMAIL' DEL_CHANNEL,
CST_EMAIL_ADDRESS PARAMETER1,
'accounts.receivable@oracle.com' PARAMETER2,
'bip-collections@oracle.com' PARAMETER3,
'Your Invoices' PARAMETER4,
'Hi'||CUST_FIRST_NAME||chr(13)|| 'Please find attached your
invoices.' PARAMETER5,
'true' PARAMETER6,
'donotreply@mycompany.com' PARAMETER7
from CUSTOMERS
```

Creating a Table to Use as a Delivery Data Source

If the delivery information is not easily available in your existing data sources, you may consider creating a table to use for your query to create the delivery XML. Following is a sample:

Important: If the JDBC driver you use does not support column alias, when you define the bursting control table, the columns must match exactly the control XML tag name. For example, the KEY column must be named "KEY", upper case is required. PARAMETER1 must be named "PARAMETER1", not "parameter1" nor "param1", and so on.

```
CREATE TABLE "XXX"."DELIVERY_CONTROL"
( "KEY" NUMBER,
  "TEMPLATE" VARCHAR2(20 BYTE),
  "LOCALE" VARCHAR2(20 BYTE),
  "OUTPUT_FORMAT" VARCHAR2(20 BYTE),
  "DEL_CHANNEL" VARCHAR2(20 BYTE),
  "PARAMETER1" VARCHAR2(100 BYTE),
  "PARAMETER2" VARCHAR2(100 BYTE),
  "PARAMETER3" VARCHAR2(100 BYTE),
  "PARAMETER4" VARCHAR2(100 BYTE),
```

```
"PARAMETER5" VARCHAR2(100 BYTE),
"PARAMETER6" VARCHAR2(100 BYTE),
"PARAMETER7" VARCHAR2(100 BYTE),
"PARAMETER8" VARCHAR2(100 BYTE),
"PARAMETER9" VARCHAR2(100 BYTE),
"PARAMETER10" VARCHAR2(100 BYTE),
"OUTPUT_NAME" VARCHAR2(100 BYTE),
"SAVE_OUTPUT" VARCHAR2(4 BYTE),
"TIMEZONE" VARCHAR2(300 BYTE),
"CALENDAR" VARCHAR2(300 BYTE)
) PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT)
TABLESPACE "EXAMPLES";
```

Tips for creating a creating bursting delivery table:

- If your split data set does not contain a DELIVERY_KEY value, then the document will not be generated nor will it be delivered because the system will not know how. For example, using the preceding example, if customer with ID 123 is not defined in the bursting delivery table, this customer will not have a document generated or delivered.
- To enable a split data set to generate more than one document or deliver to more than one destination, duplicate the DELIVERY_KEY value and provide different sets of OUTPUT_FORMAT, DEL_CHANNEL, or other parameters. For example, customer with ID 456 wants his document delivered to two e-mail addresses. To achieve this, insert two rows in the table, both with 456 as the DELIVERY_KEY and each with its own e-mail address.

[Contents](#) | [Previous](#) | [Top of Page](#) | [Next](#)



Copyright © 2010, 2011, Oracle and/or its affiliates. All rights reserved.