

DEFECTS ARE ONE OF THE MAIN REASON THAT CAUSE SIGNIFICANT PROJECT REWORK, DELAYS AND COST OVER RUNS.

You are here: [BI Publisher](#) ► [BI Publisher](#) ► Exploring BI Publisher Bursting functionality

Exploring BI Publisher Bursting functionality

Written by Raj

font size

[Email](#)

[4 comments](#)

Rate this item

(22 votes)

Bursting is one of the most widely implemented features of BI Publisher. Over the period of our BI Consultancy, Adiva consultants implemented around 10 implementation of Bursting Reports, that's include the EBS bursting which is control file based bursting and the Enterprise Bursting Reports. Our implementation includes the Invoice, Purchase Order, Telecommunication billing, Bank Payments through EFT/Etext, Tax documents, Employee salary slips, Student admission notices and Business promotion emails. We implemented a complete Business Promotion framework for one of our client using this functionality to send email and fax for the promotional events.



Click to enable Adobe Flash Player

Although the feature is documents in BI Publisher user guide and there are many blogs available on the same subject line, still every time we work on new implementation, we struggle with one or other issue as each bursting requirement is different and User guide does not cover all the functionality.

In this article, we tried to cover most of the common implementation. I would like to thanks existing blogs and Oracle support, who helped us to provide all valuable information during our bursting implementations and that's motivate us to share those valuable information with Community.

Bursting Report is the Single process, which accepts a data stream and splits it based on multiple criteria, generates output based on a template, then delivers the individual documents through the delivery channel of choice.

Bursting Processor Engine accepts the three inputs

1) Bursting Control File : Bursting Control file is the XML file holds the metadata to control the bursting process. It has following structure.

```
=>RequestSet
=>Request
  =>Delivery Section
    =>Delivery Channel
  =>Document Section
=>Template
  =>Request
    => Delivery Section
      =>Delivery Channel
    =>Document Section
=>Template
```



Recent Videos

[Oracle Application Framework Demo Session 1](#)

[Oracle Application Framework Demo Session 2](#)

Categories

OBIEE

[BI Publisher](#)

Oracle Application Framework

Hyperion

BI APPS

- 2) XML Data file
- 3) Temporary directory location

As we go through sample examples, you will get better idea. We would talk about structure and available features through these examples.

We will use following sample xml data for the bursting report.

```
<?xml version="1.0" encoding="UTF-8" ?>
<ROWSET>
  <ROW>
    <EMPLOYEE_ID>200</EMPLOYEE_ID>
    <NAME>Jennifer Whalen</NAME>
    <FIRST_NAME>Jennifer</FIRST_NAME>
    <LAST_NAME>Whalen</LAST_NAME>
    <SALARY>4400</SALARY>
    <ANNUAL_SALARY>52800</ANNUAL_SALARY>
    <TAX>14784</TAX>
    <JOB_TITLE>Administration Assistant</JOB_TITLE>
    <DEPARTMENT_NAME>Administration</DEPARTMENT_NAME>
    <MANAGER>Neena Kochhar</MANAGER>
    <EMAIL_ADDRESS>Jennifer.Whalen@testhost.com</EMAIL_ADDRESS>
    <MGR_EMAIL_ADDRESS>Neena.Kochhar@testhost.com</MGR_EMAIL_ADDRESS>
    <DELIVERY_CHANNEL>EMAIL</DELIVERY_CHANNEL>
  </ROW>
  <ROW>
    <EMPLOYEE_ID>201</EMPLOYEE_ID>
    <NAME>Michael Hartstein</NAME>
    <FIRST_NAME>Michael</FIRST_NAME>
    <LAST_NAME>Hartstein</LAST_NAME>
    <SALARY>13000</SALARY>
    <ANNUAL_SALARY>156000</ANNUAL_SALARY>
    <TAX>43680</TAX>
    <JOB_TITLE>Marketing Manager</JOB_TITLE>
    <DEPARTMENT_NAME>Marketing</DEPARTMENT_NAME>
    <MANAGER>Steven King</MANAGER>
    <EMAIL_ADDRESS>Michael.Hartstein@testhost.com</EMAIL_ADDRESS>
    <MGR_EMAIL_ADDRESS>Steven.King@testhost.com</MGR_EMAIL_ADDRESS>
    <DELIVERY_CHANNEL>EMAIL</DELIVERY_CHANNEL>
  </ROW>
  <ROW>
  <ROW>
  <ROW>
  <ROW>
</ROWSET>
```

XML Data file, Sample control files, RTF templates and a sample java program can be downloaded from [here](#)

1) Lets design the very simple bursting report control file. Employee Bursting Report with File System delivery. It burst the individual employee report and delivers to specific file system location. The file name is dynamically generated based on Employee ID.

Control file with File delivery

```
<?xml version="1.0" encoding="UTF-8" ?>
<xapi:requestset xmlns:xapi="http://xmlns.oracle.com/oxp/xapi">
  <xapi:request select="/ROWSET/ROW">
    <xapi:delivery>
      <xapi:filesystem id="FILE_DELIVERY" output="c:\emp\out1\${EMPLOYEE_ID}.pdf" />
    </xapi:delivery>
    <xapi:document output-type="pdf" delivery="FILE_DELIVERY">
      <xapi:template type="rtf" location="c:\emp\employee.rtf" />
    </xapi:document>
  </xapi:request>
</xapi:requestset>
```

Instead of defining the File as one of the delivery channel, it could be specified at Document Section as output attribute. Both definition has same effect.

```

<?xml version="1.0" encoding="UTF-8" ?>
- <xapi:requestset xmlns:xapi="http://xmlns.oracle.com/oxp/xapi">
  - <xapi:request select="/ROWSET/ROW">
    <xapi:delivery />
    - <xapi:document output="c:\emp\out1\${EMPLOYEE_ID}.pdf" output-type="pdf">
      <xapi:template type="rtf" location="c:\emp\employee.rtf" />
    </xapi:document>
  </xapi:request>
</xapi:requestset>

```

Please use the BurstingSample.java program to test this template. Make sure you have all required libraries in class path.

Defining delivery channels

A single burst report can be delivered to one or multiple channels. Beside File delivery, EMAIL, FAX, FTP and SFTP delivery also supported. Each deliver definition in Delivery section identified by ID and that referenced in Document section to associate the document to require deliver channels.

EMAIL Delivery:

```

- <xapi:delivery>
  - <xapi:email server="${EMAIL_SERVER}" port="${EMAIL_PORT}"
    from="${FROM_EMAIL_ADDRESS}" reply-to="${REPL_TO_ADDRESS}">
    <xapi:message id="EMAIL_DELIVERY1" to="${EMAIL}" cc="${CC_EMAIL_ADDRESS}"
      attachment="true" content-type="html/text" subject="Employee Details for
        ${EMPLOYEE_ID}">Dear ${NAME}, Please review your attached HR data. Thanks, HR
        Department</xapi:message>
    </xapi:email>
  </xapi:delivery>

```

Attribute value can be hard-coded or defined as substitute variable. These substitutes values can be passed as User properties to BurstingProcessorEngine.class API or can be referenced from XML data element.

Passing substitute variable as user properties while calling BurstingProcessorEngine:

```

public void run(){
try {
  BurstingProcessorEngine dp = new BurstingProcessorEngine();
  dp.setTempDirectory("c:\\emp\\tmp"); //Set the temp file
  dp.setXMLAPI("c:\\emp\\BurstingControlFile_DEPT_SUMMARY.xml"); //Set Bursting Control file.
  dp.setData("c:\\emp\\Employee_data.xml"); //Data File
  //User Properties//
  Properties prop = new Properties(); //add properties and variables//
  prop.put("user-variable:EMAIL_SERVER", "MySMTPSERVER");
  prop.put("user-variable:EMAIL_PORT", "25");
  prop.put("user-variable:FROM_EMAIL_ADDRESS", "test@testhost.com");
  prop.put("user-variable:CC_EMAIL_ADDRESS", "cc@testhost.com");
  dp.setConfig(prop);
  dp.process();
} catch (Exception e) {
  e.printStackTrace(); }
}

```

For single delivery definition, there could be multiple Messages. The same report can be sent to different recipient with different subject and message content.

The same report can be delivered to multiple delivery channels. We need to associate delivery channels with document in document section.

```

<?xml version="1.0" encoding="UTF-8" ?>
- <xapi:requestset xmlns:xapi="http://xmlns.oracle.com/oxp/xapi">
- <xapi:request select="/ROWSET/ROW">
- <xapi:delivery>
- <xapi:email server="{EMAIL_SERVER}" port="{EMAIL_PORT}"
  from="{FROM_EMAIL_ADDRESS}" reply-to="{REPL_TO_ADDRESS}">
  <xapi:message id="EMAIL_DELIVERY1" to="{EMAIL_ADDRESS}"
    cc="{CC_EMAIL_ADDRESS}" attachment="true" content-type="html/text"
    subject="Employee Details for {EMPLOYEE_ID}">Dear {NAME}, Please review your
    attached HR data. Thanks, HR Department</xapi:message>
  <!-- Second message for Employee's Manager -->
  <xapi:message id="EMAIL_DELIVERY2" to="{MGR_EMAIL_ADDRESS}"
    cc="{CC_EMAIL_ADDRESS}" attachment="true" content-type="html/text"
    subject="Employee Details for {EMPLOYEE_ID}">Dear {MANAGER}, Please review
    your employee {NAME}'s attached HR data. Thanks, HR Department</xapi:message>
  </xapi:email>
</xapi:delivery>
- <xapi:document output="c:\emp\out2\{EMPLOYEE_ID}.pdf" output-type="pdf"
  delivery="EMAIL_DELIVERY1,EMAIL_DELIVERY2">
  <xapi:template type="rtf" location="c:\emp\employee.rtf" />
</xapi:document>
</xapi:request>
</xapi:requestset>

```

FAX Delivery:

BI Publisher supports the Fax delivers through fax modems configured on CUPS. You can configure fax modems on CUPS with efax (<http://www.cce.com/efax/>) and FAX4CUPS (<http://www.gnu.org/directory/productivity/special/fax4CUPS.html>). Please refer the BI Publisher User Guide on how to setup the Fax Server. Once you setup the Fax server, it can be accessed through URL. Get the Fax Server URL and use it in control file as follows.

```

<xapi:delivery>
<xapi:fax server="ipp://myhost:631/printers/fax">
  <xapi:number id="FAX1">916505069560</xapi:number>
</xapi:fax>
</xapi:delivery>

```

I used the “ipp://myhost:631/printers/fax” as sample URL and 12345567789 as sample Fax No.

PRINT Delivery:

The BIP Delivery Server supports Internet Printing Protocol (IPP) as defined in RFC 2910 and 2911 for the delivery of documents to IPP-supported printers or servers, such as CUPS. Please refer the BI Publisher user guide on how to setup the IPP Print server through CUPS. Once you setup the server, Printer can access through Print URL. Use the same URL to define the Print delivery channel for Bursting.

```

<xapi:delivery>
<xapi:print id="PRINT_DELIVERY" printer="ipp://myhost:631/printers/P2165" copy-no="2"
paper- size="letter" orientation-requested="3" />
</xapi:delivery>

```

Here “ipp://myhost:631/printers/P2165” is the sample URL. Print attributes can be hard-coded or derived from Substitute variables.

FTP Delivery

```

<xapi:ftp id="ftp_delivery" server="myCompany.com" user="{user}" password="{password}" remote-
directory="{RemoteDir}" remote-file="{EMPLOYEE_ID}.pdf" />

```

SFTP Delivery:

For SFTP delivery, there is one additional attribute “secure-ftp”.

```
<xapi:ftp id="sftp_delivery" server="myCompany.com" user="{user}" password="{password}" remote-
directory="{RemoteDir}" remote-file="{EMPLOYEE_ID}.pdf" secure-ftp="TRUE"/>
```

Template and Output Type:

Following table describe the type and output-type attributes values for the supported templates by Bursting Report:

Template Type	type attribute	Valid output-type attribute value
RTF	ptf	pdf , rtf, html, excel
PDF	pdf	pdf
ETEXT	etext	text
EXCEL	excel	excel
XSLFO	Xslfo	pdf , rtf, html, excel

Make sure to set the valid type (template type) and output-type value while defining the Document Section. type, output-type and template location can hard-coded or derived through substitute variables. Defining Key is optional but required when we want to generate status report.

```
<xapi:document key="{EMPLOYEE_ID}" output="c:\emp\out1\{EMPLOYEE_ID}.pdf" output-
type="{outputType}">
```

```
<xapi:template type="rtf" location="{Template_location}"/>
```

```
</xapi:document>
```

Conditional Template Layout..

Using the filter attribute at template section allow to burst the report conditionally. The value of filter should be a valid XPATH expression against the xml data.

In the following example, we are using different template for different department.

```
<?xml version="1.0" encoding="UTF-8" ?>
- <xapi:requestset xmlns:xapi="http://xmlns.oracle.com/oxp/xapi">
- <xapi:request select="/ROWSET/ROW">
- <xapi:delivery>
  <xapi:filesystem id="FILE_DELIVERY" output="c:\emp\out3
    \{EMPLOYEE_ID}.pdf" />
</xapi:delivery>
- <xapi:document output-type="pdf" delivery="FILE_DELIVERY">
  <xapi:template type="rtf" location="c:\emp\employee_admin.rtf"
    filter="//ROW[DEPARTMENT_NAME='Administration']" />
  <xapi:template type="rtf" location="c:\emp\employee_mkt.rtf"
    filter="//ROW[DEPARTMENT_NAME='Marketing']" />
</xapi:document>
</xapi:request>
</xapi:requestset>
```

There could be multiple templates within one document section and there could be multiple document section within one Request. Using the Filter we can apply the Template conditionally.

Conditional Delivery

The same Filter functionality can be used to implement the conditional delivery based on the recipient delivery preference.


```

<xapi:document key="{EMPLOYEE_ID}" output="c:\emp\out2\{EMPLOYEE_ID}.pdf"
  output-type="pdf" delivery="EMAIL_DELIVERY">
  <xapi:template type="rtf" location="c:\emp\employee.rtf" filter="//ROW
    [DELIVERY_CHANNEL='EMAIL']" />
</xapi:document>
<xapi:document key="{EMPLOYEE_ID}" output="c:\emp\out2\{EMPLOYEE_ID}.pdf"
  output-type="pdf" delivery="FAX_DELIVERY">
  <xapi:template type="rtf" location="c:\emp\employee.rtf" filter="//ROW
    [DELIVERY_CHANNEL='PRINT']" />
</xapi:document>
<xapi:document key="{EMPLOYEE_ID}" output="c:\emp\out2\{EMPLOYEE_ID}.pdf"
  output-type="pdf" delivery="PRINT_DELIVERY">
  <xapi:template type="rtf" location="c:\emp\employee.rtf" filter="//ROW
    [DELIVERY_CHANNEL='PRINT']" />

```

Single Template multiple output

```

<?xml version="1.0" encoding="UTF-8" ?>
- <xapi:requestset xmlns:xapi="http://xmlns.oracle.com/oxp/xapi">
-   <xapi:request select="/ROWSET/ROW">
-     <xapi:delivery>
-       <xapi:filesystem id="FILE_DELIVERY" output="c:\emp\out3
        \{EMPLOYEE_ID}.pdf" />
-     </xapi:delivery>
-     <xapi:document output-type="pdf" delivery="FILE_DELIVERY">
-       <xapi:template type="rtf" location="c:\emp\employee_admin.rtf"
        filter="//ROW[DEPARTMENT_NAME='Administration']" />
-       <xapi:template type="rtf" location="c:\emp\employee_mkt.rtf"
        filter="//ROW[DEPARTMENT_NAME='Marketing']" />
-     </xapi:document>
-   </xapi:request>
- </xapi:requestset>

```

Multiple Requests

Here is an example of multiple bursting reports within a single process. These are Employee, Department and Organization level report. Employee report will be delivered to employee and Department Level Report will be delivered to respective Department Manager. The organization level report will be for the record purpose

```

<?xml version="1.0" encoding="UTF-8" ?>
- <xapi:requestset xmlns:xapi="http://xmlns.oracle.com/oxp/xapi">
  <xapi:request select="/ROWSET/ROW">
    - <xapi:delivery>
      - <xapi:email server="{EMAIL_SERVER}" port="{EMAIL_PORT}"
        from="{FROM_EMAIL_ADDRESS}" reply-to="{REPL_TO_ADDRESS}">
        <xapi:message id="EMAIL_DELIVERY1" to="{EMAIL}" cc="{CC_EMAIL_ADDRESS}"
          attachment="true" content-type="html/text" subject="Employee Details for
            ${EMPLOYEE_ID}>Dear ${NAME}, Please review your attached HR data. Thanks, HR
            Department</xapi:message>
        </xapi:email>
      </xapi:delivery>
      - <xapi:document output="c:\emp\out1\${EMPLOYEE_ID}.pdf" output-type="pdf"
        delivery="EMAIL_DELIVERY1">
        <xapi:template type="rtf" location="c:\emp\employee.rtf" />
      </xapi:document>
    </xapi:request>
    <xapi:request select="/ROWSET/ROW/DEPARTMENT_NAME">
      - <xapi:delivery>
        - <xapi:email server="{EMAIL_SERVER}" port="{EMAIL_PORT}"
          from="{FROM_EMAIL_ADDRESS}" reply-to="{REPL_TO_ADDRESS}">
          <!-- Second message for Employee's Manager -->
          <xapi:message id="EMAIL_DELIVERY2" to="{MGR_EMAIL_ADDRESS}"
            cc="{CC_EMAIL_ADDRESS}" attachment="true" content-type="html/text"
            subject="Employee Details for Dept ${DEPARTMENT_NAME}>Dear ${MANAGER}, Please
            review attached HR data for your department. Thanks, HR Department</xapi:message>
          </xapi:email>
        </xapi:delivery>
        - <xapi:document output="c:\emp\dept\${DEPARTMENT_NAME}.pdf" output-type="pdf">
          <xapi:template type="rtf" location="c:\emp\Department.rtf" />
        </xapi:document>
      </xapi:request>
      <xapi:request>
        <xapi:delivery />
        - <xapi:document output="c:\emp\org\org_report.pdf" output-type="pdf">
          <xapi:template type="rtf" location="c:\emp\organization.rtf" />
        </xapi:document>
      </xapi:request>
    </xapi:requestset>

```

Sample Java program

```

import java.util.Properties;
import oracle.apps.xdo.batch.BurstingProcessorEngine;
import oracle.apps.xdo.common.log.Logger;

public class BurstingSample {
    public BurstingSample() {
    }

    public static void main(String[] args) {
        BurstingSample burstingSample = new BurstingSample();
        burstingSample.run();
    }

    public void run(){
        try {
            Logger.setLevel(Logger.STATEMENT); //trun on the debug mode//
            BurstingProcessorEngine dp = new BurstingProcessorEngine();
            dp.setTempDirectory("c:\\emp\\tmp"); //Set the temp file
            dp.setXMLAPI("c:\\emp\\BurstingControlFile.xml"); //Set Bursting Control file.
            dp.setData("c:\\emp\\Employee_data.xml"); //Data File
            //User Properties//
            Properties prop =
                new Properties(); //add properties and variables//
            prop.put("user-variable:EMAIL_SERVER", "MYSMTPHOST");
            prop.put("user-variable:EMAIL_PORT", "25");
            prop.put("user-variable:FROM_EMAIL_ADDRESS", " test@testhost.com");
            prop.put("user-variable:CC_EMAIL_ADDRESS", " cc@testhost.com");
            dp.setConfig(prop);

            dp.process();

        } catch (Exception e) {

```

```
e.printStackTrace();  
}  
}}
```

Read 49216 times

4 comments

**Wednesday, 09 January 2013 14:02** | posted by Ebin[Comment Link](#)

Hi,

In the above example

If in the control file where to='\${EMAIL}' is present..

if this EMAIL refers to a column of a table at the backend, consisting of various emails..like
EMAILramesh@abc.comsuresh@abc.comthink if, one of the case is where there are multiple emails mentioned in the column like
EMAILramesh@abc.com;suresh@abc.com

will in this case, it will pick up this multiple mail ids and send mails to both the addresses.

In my case, with one mail id its working fine, but wherever there are multiple mail id like 2 mails together separated by a semicolon,
the bursting is failing..

Please help..

How can i achieve this..

[Report](#)**Wednesday, 19 September 2012 12:47** | posted by Guy[Comment Link](#)

Hi,

I see that when you define the print delivery through the xapi:print element, you use attributes such as copy-no="2" and
orientation-requested="3". Now personally I am looking for the attribute that allows for printing on both sides of the
paper. Do you have any idea what this is? Is there a place where I can find documentation on this and other attributes for the print
element?

Thanks in advance.

[Report](#)**Thursday, 06 September 2012 15:27** | posted by Raj[Comment Link](#)Yes, it is possible. Add 2 Document Section, in one section u can define the template section for pdf template and for
other document section u can define the template section for csv template. I have never tested the cvs format so, you
have to check it out.[Report](#)**Wednesday, 05 September 2012 13:18** | posted by Naveen[Comment Link](#)

Hi,

I have a report which has Invoice Details (Invoice ID, Item, Cost, etc...) in PDF format. Same report i need details of only
certain fields (Invoice No and Item) in CSV format.

I can achieve this by having two templates (full details and partial details).

Now can write a bursting logic to send two files (CSV and PDF) to a file server?

[Report](#)

Leave a comment

Make sure you enter the (*) required information where indicated.

Basic HTML code is allowed.

Message *

enter your message here...

Name *

enter your name...

Email *

enter your e-mail address...

Enter the two words you see below

Submit comment

[back to top](#)