

# XML Publisher and FSG for Beginners

Susan Behn, Alyssa Johnson, and Karen Brownfield  
*Solution Beacon, LLC*

While BI Publisher has provided nearly unlimited formatting capabilities to report developers, use of BI Publisher with FSG can present interesting challenges. This whitepaper is for users familiar with FSG but who may be beginners with BI Publisher. Topics will include a brief overview of BI Publisher, the two ways to publish FSG reports; the two seeded FSG templates, advantages and disadvantages of each template, what FSG formatting features are still usable, and show some simple changes to each template.

## BI Publisher

BI Publisher, previously known as XML Publisher, and has been available as part of the E-business Suite of products since 2006. BI Publisher keeps the data extract from the database separate from the presentation of that data in a report.

This provides several advantages

- The same data file can be used for multiple reports
- Output can be formatted in Word, PDF, Excel, XBRL, and XML without changing the program that extracted the data
- Users proficient in Word, Excel can design the output

The program that produces the XML file is called the Data Template. Any tool that can output XML can create the Data Template. This can be Oracle Reports, JDeveloper, PL/SQL etc. Data Templates can even contain embedded SQL to do further extracts.

The portion of BI Publisher that takes the XML file as input and creates the report is the Report Template or Presentation Template. Presentation Templates cannot create XML, cannot contain SQL, and cannot add data to the template. Within limits, data can be reordered, reformatted, and conditions can be added to exclude data in the XML file. Presentation Templates can be created with common desktop tools like Microsoft Word or Adobe Acrobat. BI Publisher works with Office 2000, 2003 and 2007.

BI Publisher has three versions. One version is the Desktop version, more commonly called the Template Builder. This is the add-in to Word or Adobe that allows you to build the templates. The Enterprise version of the Desktop BI Publisher can be downloaded from OTN (current version is 10.1.3.4) and used against the E-Business Suite embedded version as long as pivot tables are not needed. Alternatively, the E-Business Suite client side version can be downloaded from My Oracle Support. This will actually download version 10.1.3.2 of the Enterprise version. The Template Builder supports special formatting such as logos, charts and graphs, tables, and barcodes. It also supports the use of any special formatting offered by Word or Adobe.

The next two versions reside on the database and are also called the Processing Engine. One is embedded in E-Business Suite. This version is referenced as XML Publisher or BI Publisher 5.6.3. This version of BI Publisher is included in the E-Business Suite license as long as no custom tables or no custom columns added to Oracle tables are referenced. The other version residing at the database level is BI Publisher Enterprise Edition. This version allows you to use any field in any table, but requires additional licensing. Whether you use the E-Business Suite embedded version or the Enterprise version, it is this Processing Engine that merges the XML data with the Presentation or Report Template. The Processing Engine supports multiple methods of report delivery such as print, fax, email, and web. Reports can be burst according to your defined criteria and each section delivered to a different person.

BI Publisher is a tremendous improvement over the traditional method of writing and delivering reports. Because of this the proliferation of BI Publisher reporting has grown exponentially. In Release 11i, 500 templates were available. Now in Release 12.1, all Oracle standards reports have seeded BI Publisher templates available. Currently, there are 2724 templates and 2092 data sources across 95 products.

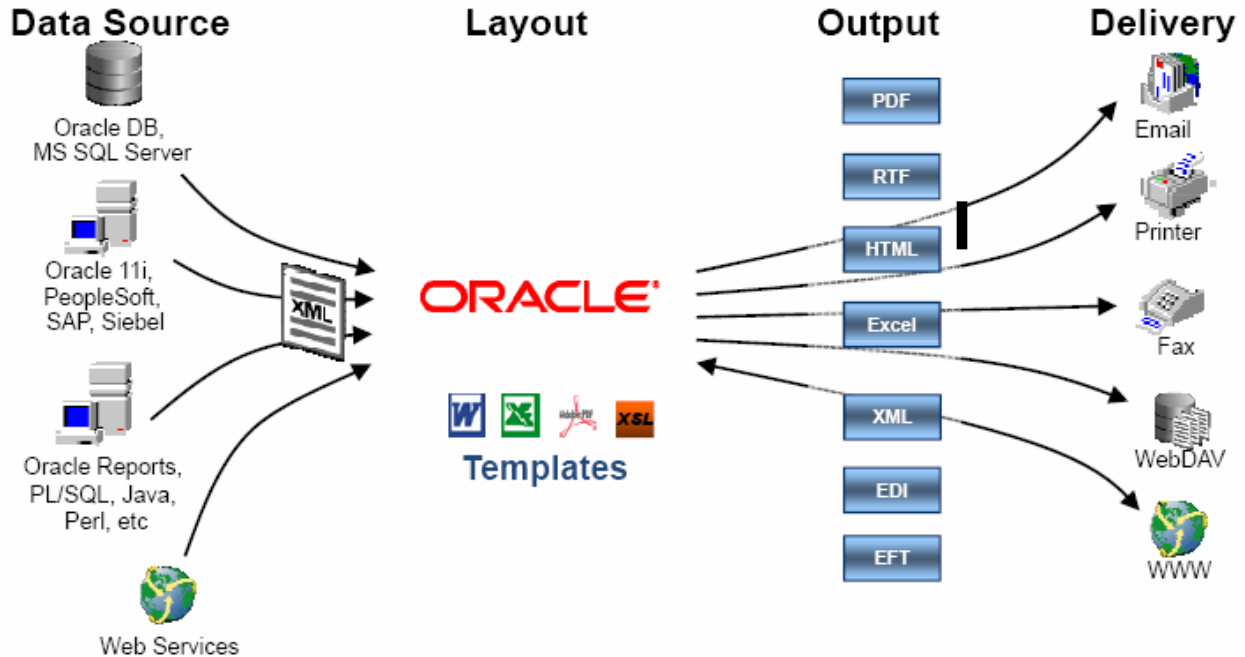


Figure 1 – Oracle BI Publisher Processing

## FSG reporting

While BI Publisher is a fairly new technology, the Financial Statement Generator has been an integral part of Oracle Financials for a long time and FSG reports are used by organizations world-wide. It quickly became evident that it was going to be necessary to deliver FSG reports using BI Publisher technology without requiring users to re-write every FSG report as a BI Publisher template. In order to blend the two technologies, patching, setups, and some tips and techniques developed through trial-and-error became required steps.

## Patching

Patching is a fact-of life in the marriage of FSGs and BI Publisher. Even for organizations on 12.1.2, a recommended patch has already been released. The key is to find what the patches need to be applied quickly and efficiently. The following My Oracle Support Documents are the first documents to check if you are implementing BI Publisher or if FSG reports are not working correctly with BI Publisher

- Apply Latest FSG Patch - How to Find It And Verify It's Been Applied [ID 189282.1]
- How To Determine Which FSG Patches Have Been Applied In Release 11i, 12.0 and 12.1 [ID 549289.1] – This document contains script will identify which FSG patches have been applied
- FSG XML Publisher FAQ [ID 370358.1] – Release 11i
- FSG Functional FAQ [ID 108236.1] – this document provides tips on how to troubleshoot common issues related to Financial Statement Generator (FSG)

As of 3rd February, 2010 the latest FSG patches are:

- 11.5
  - 8640288 (with XML)
  - 9013928 (without XML)
- 12.0
  - 9182320:R12.RG.A

- 9276305:R12.GL.A for FSG drilldown
- 12.1.1 and 12.1.2
  - 9182320:R12.RG.B
  - 9276305:R12.GL.B for FSG drilldown

Additional patches that might be required include Patch 5887917. This is the Desktop Add-in for Word. It is BI Publisher Desktop 10.1.3.2 which also contains the XML Publisher 5.6.3 core libraries and is intended to develop templates for XML Publisher 5.6.3. Also, if a requirement exists to produce XBRL reports, patch 5745329 will be required. This is the Server Side, XBRL Publisher which marries XML Publisher with XBRL taxonomies to produce XBRL reports as required by outside agencies.

## Setups

After patching is completed, certain setups have to be completed correctly, in order for the application to generate FSGs. Verify the following profile option values

- Concurrent: Report Access Level – cannot be NULL at Site level
- FSG: Create Numbers in XML – set to ‘N’ to block creation of attribute RealNum
- FSG: Database Codeset (conditional)

Determine Codeset by running the following SQL statement

```
SELECT tag
FROM fnd_lookup_values
WHERE lookup_type = 'FND_ISO_CHARACTER_SET_MAP'
AND lookup_code = SUBSTR(USERENV('LANGUAGE'),
INSTR(USERENV('LANGUAGE'), '.')+1)
AND language = 'US';
```

If the result of the SQL statement is UTF-8, then no action is required. However, if any other codeset, then a Database Codeset profile option needs to be created and set.

To create the Database Codeset profile option, navigate to the Application Developer responsibility and open the Profile Options form. Complete the form with the values shown below:

The screenshot shows the Oracle Profiles form with the following fields and values:

- Name: RG\_XML\_CODESET
- Application: Application Report Generator
- User Profile Name: FSG: Database Codeset
- Description: Database Codeset
- Hierarchy Type: Security

**Hierarchy Type Access Level**

	Visible	Updatable
Site	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Application	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Responsibility	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Server	<input type="checkbox"/>	<input type="checkbox"/>
Server+Responsibility	<input type="checkbox"/>	<input type="checkbox"/>
Organization	<input type="checkbox"/>	<input type="checkbox"/>
User	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

**Active Dates**

Start: 01-01-1951  
End: [Empty field]

**User Access**

Visible  
 Updatable

SQL Validation used for the Profile Option's List of Values

Figure 2 – Creating the FSG: Database Codeset profile option

Then return to the System Administrator responsibility and set the FSG: Database Codeset profile option to the result of the query above.

- FSG: Debug

When Oracle supports asks you to run a FSG with "RG\_DEBUG\_ON" set to yes, or "FSG: Debug" set to yes, the below steps needs to be done. Navigate to the Application Developer responsibility and open the Profile Options form. Complete the form with the values shown below and then set the profile option to 'Y'. Because there is no validation on the value, be sure to set it to 'Y', not to 'Yes'. See How to create and use the RG\_DEBUG\_ON profile option [ID 435029.1]

The screenshot shows the Oracle Profiles form with the following fields and settings:

- Name: RG\_DEBUG\_ON
- Application: Application Report Generator
- User Profile Name: FSG: Debug
- Description: (empty)
- Hierarchy Type: Security
- Hierarchy Type Access Level table:

	Visible	Updatable
Site	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Application	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Responsibility	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Server	<input type="checkbox"/>	<input type="checkbox"/>
Server+Responsibility	<input type="checkbox"/>	<input type="checkbox"/>
Organization	<input type="checkbox"/>	<input type="checkbox"/>
User	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
- Active Dates:
  - Start: 01-JAN-1951
  - End: (empty)
- User Access:
  - Visible
  - Updatable
- SQL Validation used for the Profile Option's List of Values: (empty text area)

Figure 3 – Creating the FSG: Debug profile option

The XML temporary file directory must be set to a directory that application users have both read and write privileges. To set the directory path, navigate to the XML Publisher Administrator responsibility. Select Home function, Administration tab. Open the 'General' property and set the Temporary directory

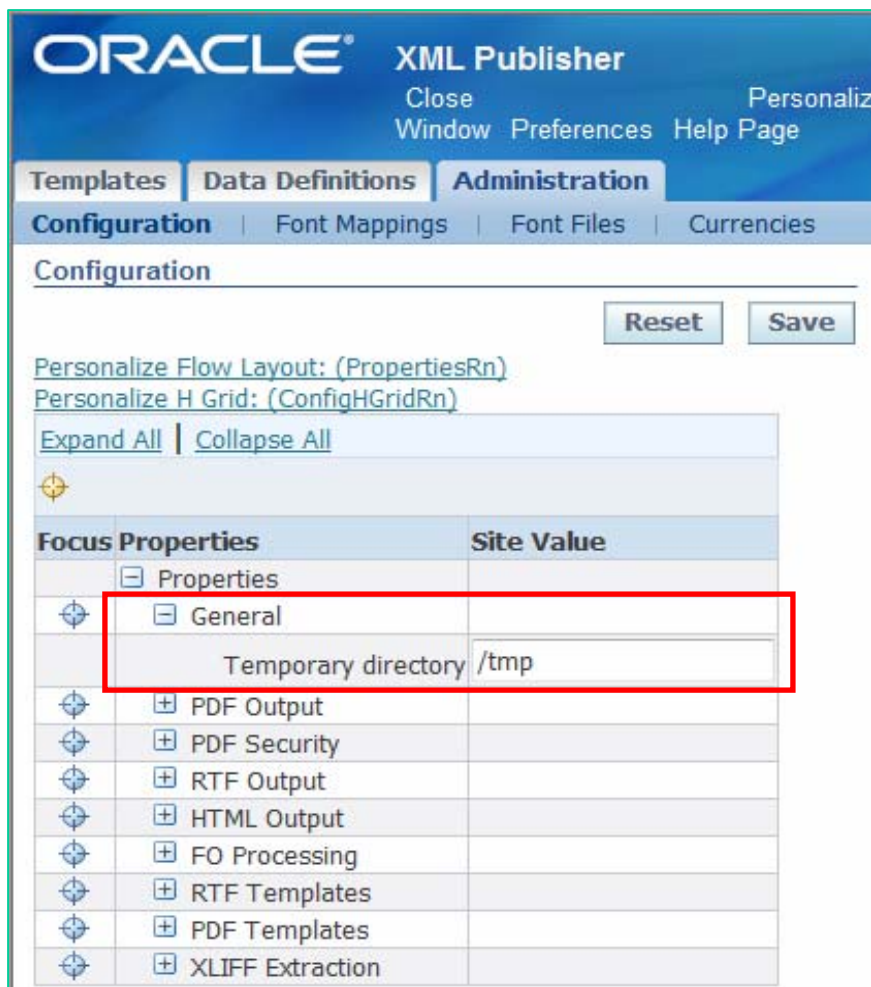


Figure 4 – Setting the Temporary directory

## FSG Template Common Issues

Common issues with FSG Publishing do arise and can often be addressed quickly and efficiently. Some of these issues include:

- FSG Publishing with Report Manager using Content Sets and Parallel Type Processing causes Error [ID 552592.1] – FSG publishing with Report Manager does not support parallel processing with content sets
- RGFSGXML One Step Program - Publish FSG Report Throws ORA-06550 PLS-00306 [ID 787396.1] – This error occurs due to an issue with a mismatch on number of parameters. Be sure to check patch levels.
- Request 'Xml Report Publisher' Erroring With XDOException When Published In PDF Format [ID 417810.1] – The sequence/position of columns must be in ascending order. If they are not, then column width values become negative causing a java error
- Xml Publisher Drops Two Columns And Changes Template Alignment [ID 464418.1] – FSG publishing with BI Publisher does not support overlap of columns and position in ascending order.
- Signal 11 Error – This indicates a memory error but can have several causes. The latest patchset addresses a bug that resulted in a Signal 11 error, but FSG definitions also can cause this error. If on the latest patchset, then search on My Oracle Support for other causes.

## FSG Process Flows

FSGs can be run in several ways. This gives users the flexibility to run the FSGs in a way that suits their organization the best. One process flow involves a two step process. The first step is to request the FSG Report using XML as the output option, either from the Submit Financial Report window or from the Submit Request window (Program - Run Financial Statement Generator). The second step is to run the XML Report Publisher concurrent process, applying the relevant template. Another process flow is a one-step process. Users run the Program - Publish FSG Report from the Submit Request window. Finally, users can select the Financial Report Submission function from Report Manager.

## FSG Seeded Templates

Three seeded templates are available: FSG: Basic Template (RGFSGXPB.rtf); FSG: Fixed Number of Columns (FSGTEMP.rtf); and Generic FSG Template (GENERICFSG\_en.xsl). The Generic FSG Template is the seeded template for Report Manager. This template should never be modified per How To Modify The Generic FSG Template in Report Manager [ID 388130.1]. If using XML Publisher to create a custom template for use in Report Manager and want to use this as the basis, upload as a new name. Note: same My Oracle Support ID states that Report Manager does not provide method of creating generic template. FSG-specific templates can be generated with the Report Manager Financial Report Template Editor

The screenshot displays the Oracle Report Manager interface for creating a Financial Statement Generator (FSG) template. The page title is "Financial Statement Generator Templates". The main content area is divided into three sections: "Report", "Template Selection", and "Editor Selection".

- Report Section:** Contains a text input field for "Report Name" with the value "Balance Sheet SSC".
- Template Selection Section:** Features two radio buttons: "Create New" (which is selected) and "Modify Existing". There is also a "Language" dropdown menu currently set to "Arabic".
- Editor Selection Section:** Includes a dropdown menu for "Editor" with "Excel Template Editor" selected.

Navigation and utility elements include "Diagnostics", "Home", "Logout", "Preferences", "Help", and "Personalize Page" links at the top and bottom of the page. "Cancel" and "Create" buttons are positioned at the top right and bottom right of the form area, respectively.

Figure 5 – Financial Report Template Editor

Enter the FSG Report Name that you want to generate a template for. Select Excel Template Editor as the Editor. Click on Create. Wait while the Excel template is created. Notice that the Rows and Columns correspond to the Report Definition for the named report.

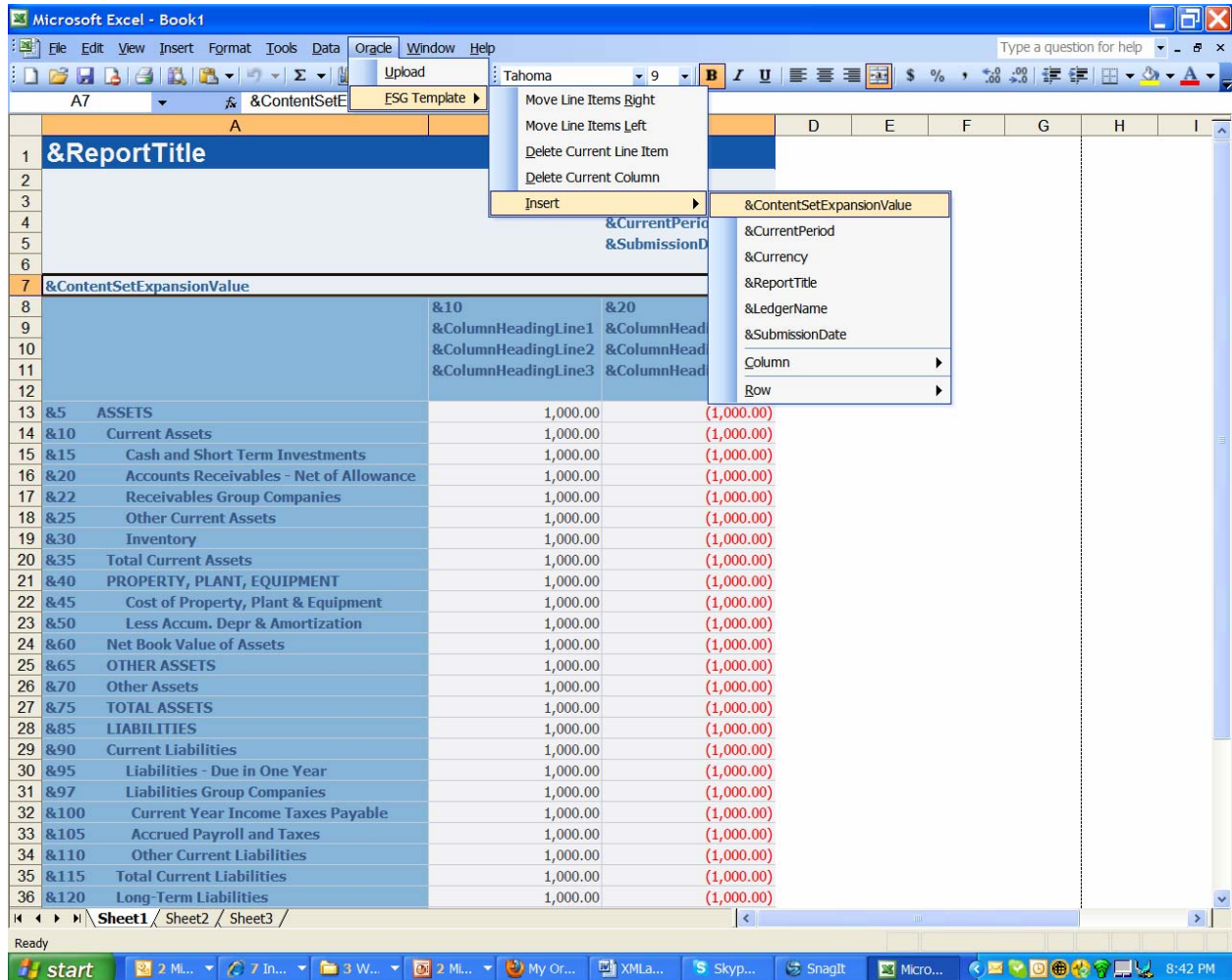


Figure 6 – Excel Template Editor

Select Oracle > Upload to upload the template to the BI Publisher Template Library. No modifications to the template are required; it can just be uploaded to the template library.

The FSG: Basic Template (RGFSGXPB.rtf) text output is designed for four widths: 80, 132, 180, 255. Anything greater than 255 requires export to spreadsheet. XML subtracts margins from these totals using 10 CPI plus cell margin. For example, to calculate the maximum usable space for a landscape 8 ½ X 11 paper, if the template has 1/2 inch margins and a cell has .08 margins. The maximum allowable space is 132 minus 10 (margins) minus 2 (cell margin) minus 1 (currency format) = 119 maximum. Users must allow 1 extra space for any negatives if 'Currency: Negative Format' not set to: -XXX. It is important to use 'Build Column Set' to verify ending position of last column. If the ending position exceeds "allowable end column", then the report wraps to next page.

Additionally, the font size of 'Report Data' element does *not* shift horizontal placement. If the text becomes too big for the cell, then it wraps within the column. Proportional fonts still align numbers. Negative Currency Format can cause number to shift; use the text version to properly align numbers.

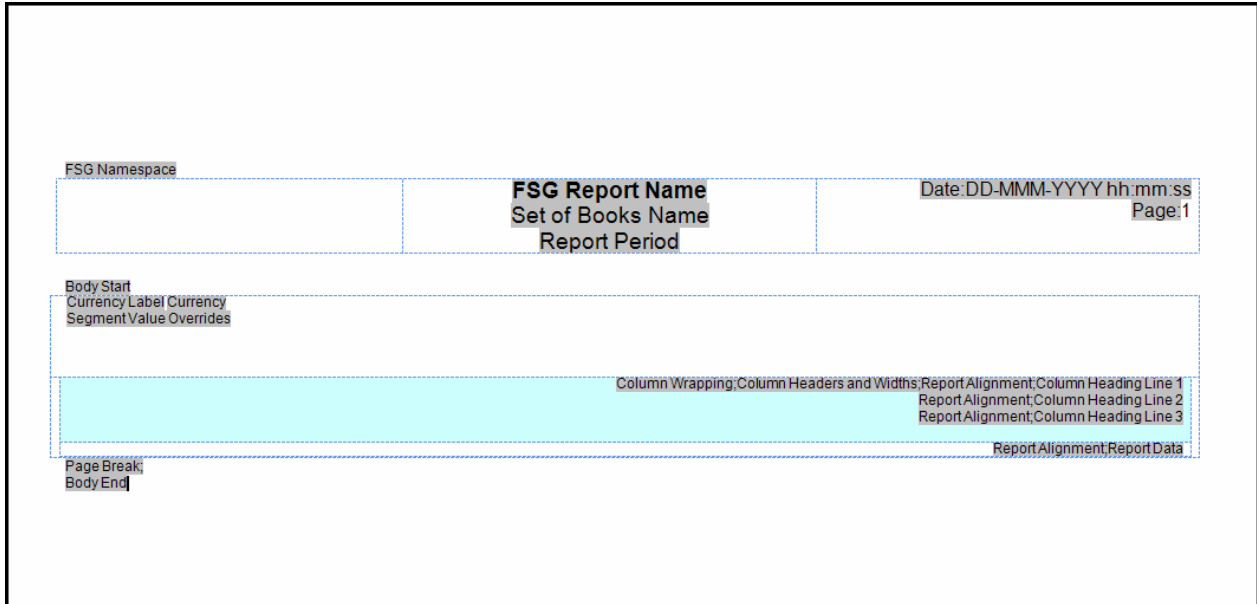


Figure 7 – FSG: Basic Template (RGFSGXPB.rtf)

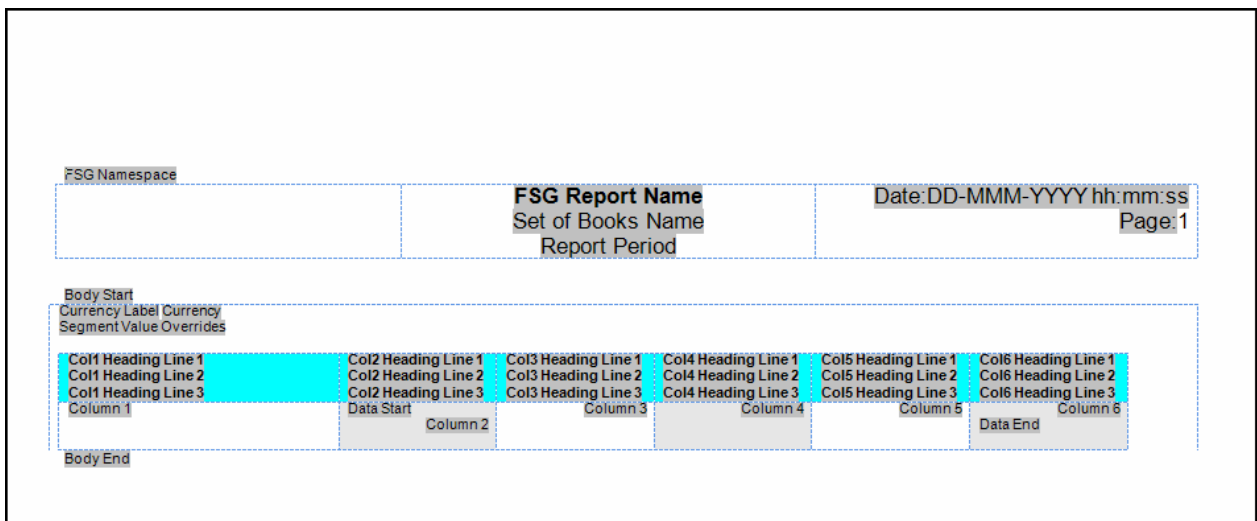


Figure 8 – FSG: Fixed Number of Columns (FSGTEMP.rtf)

In both the Basic and Fixed Number of Columns Templates, the following FSG definitions are not recognized: Lines to Skip, Page Break. The templates do recognize the Underline Character and the Indent. The spaces for the indent will not show up in the XML tag, but will be visible on the report.

## Conclusion

More information can be found at the following websites:

- Publisher Home Page
  - <http://www.oracle.com/technology/products/xml-publisher/index.html>
- Publisher Demo Catalog
  - <http://www.oracle.com/technology/products/xml-publisher/demoshelf/Shelf.html>

- Publisher Blog
  - <http://blogs.oracle.com/xmlpublisher/>
- Publisher Forum
  - <http://forums.oracle.com/forums/forum.jspa?forumID=245>

With the nearly unlimited formatting capabilities BI Publisher provides to report developers, use of BI Publisher with FSG is functionality that report developers and functional users will want to add to their skill sets.