

Oracle BI Publisher was originally developed to solve these reporting problems. It was first released with Oracle E-Business Suite 11.5.10 towards the end of 2005. The original release was called XML Publisher 5.5 and consisted of 2 products – Oracle XML Publisher (Desktop) and Oracle BI Publisher (Server). Oracle BI Publisher Desktop was a Microsoft Word Add-in that provided the ability to import an BI or SQL data source, lay out a report template using standard Microsoft Word tables and formatting, and then run the report from within Microsoft Word. This initial release lacked several key features that stopped it being a complete end-to-end solution; as previously mentioned the server element was more of a “toolkit” and you had to build the actual application yourself, and you still had to define some parts of the report – the “report definition” file with an “.xdo” suffix, that brought together the dataset, the template and details of any report parameters – by hand.

Following the initial success of Publisher 5.5, an updated client version (XML Publisher 5.6) was released in April 2006. The client-side install consists of an updated Template Builder for Microsoft Word, with a new BI Publisher Report Definition Wizard. This latest version of Oracle BI Publisher brings the Desktop and Server versions of the product into a new release called Oracle BI Publisher Enterprise, and support is now extended to Oracle PeopleSoft and J.D. Edwards.

The Oracle BI Publisher Enterprise server elements are available as an “out of the box install” using the Oracle Universal Installer, and is available for download from <http://edelivery.oracle.com>. The client-side install consists of an updated Template Builder for Microsoft Word, with a new BI Publisher Report Definition Wizard. A plug-in for Microsoft Excel was expected to be provided, to allow report analysis via this spreadsheet tool, but has been delayed until Release 6.0

BI Publisher Enterprise is a complete web-based reporting application that allows users to:

- Manage reports in a folder directory
- Create reports combining multiple databases as well as web services
- Define the Report Layout in Microsoft Word or Adobe Acrobat
- Extract and translate the boilerplate text in reports
- Schedule and deliver reports via a variety of methods (email, fax or print)
- Generate volumes of reports with minimal loading on the database engine.

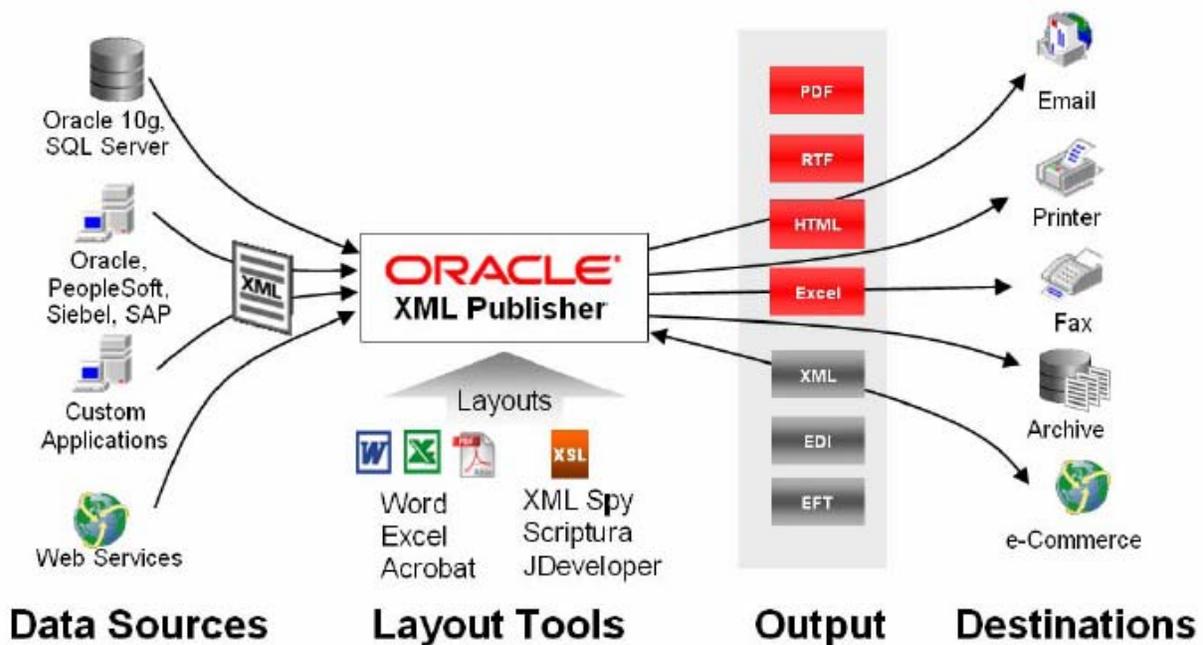


Figure 1 - BI Publisher Enterprise Capabilities

Installing BI Publisher Template Builder

Patch 5472959 provides the latest version of BI Publisher, and is applied to the application server. After the application of the patch and pre-requisites, PDF Printing needs to be enabled in the instance. The “PDF Publisher” print style and PASTA_PDF print driver provide the capability to print PDF files. The “—Pasta Universal Printer” type is associated with the style and driver.

Additionally, Patch 5887917 should be applied to the template designer’s client computer. This patch contains the Template Builder and the Template Viewer, two desktop tools that will increase your productivity using XML Publisher.

The Template Builder is an extension to Microsoft Word that simplifies the development of RTF templates. It automates many of the manual steps that would otherwise be required. The Template Viewer is a Java application that facilitates the rapid development of templates by providing advanced preview capabilities. See the patch readme and help files for user documentation and desktop system requirements.

Once installed, start up Microsoft Word. A new toolbar and menu will be available:

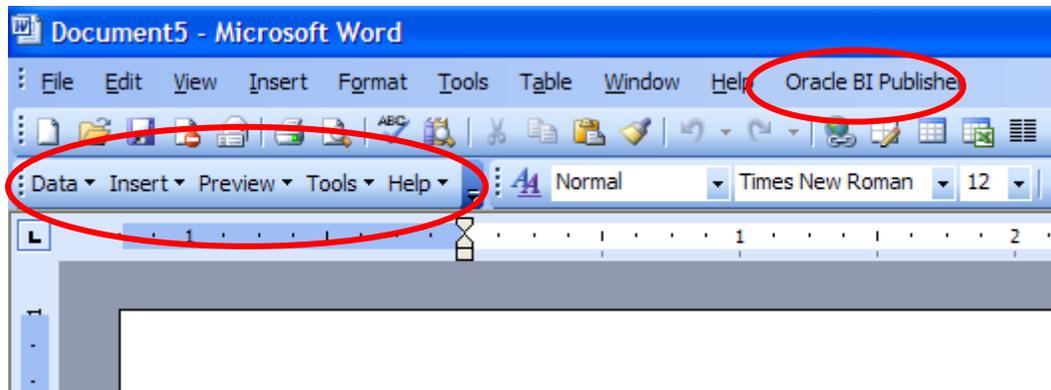


Figure 2 - Microsoft Word with BI Publisher Template Builder installed

Patch 5887917 provides a Template Builder Tutorial, together with example BI output and templates. It also provides the latest BI Publisher User’s Guide

Oracle Application Desktop Integrator (ADI) and Web ADI

Oracle Applications Desktop Integrator (ADI) is an integral part of Oracle Applications. ADI is a spreadsheet-based extension of General Ledger and Oracle Assets that offers full-cycle accounting and asset management within the comfort and familiarity of a spreadsheet.

Oracle ADI is a spreadsheet-based extension to Oracle Applications that offers full cycle accounting within the comfort and familiarity of a spreadsheet. Oracle ADI combines a spreadsheet’s ease of use with the power of Oracle Applications, to provide true desktop integration during every phase of your accounting cycle. You can create budgets, record transactions, and run financial statements and inventory reports all without leaving your spreadsheet.

Oracle ADI was initially released as a stand-alone application, giving you the ability to revise budgets, create journal entries, and define financial reports from any location, without being connected to your server. You only need to connect when you want to transfer data to or from Oracle Applications.

One of the major benefits of Oracle ADI is the Analysis Wizard. The Analysis Wizard helps you to drill down to relevant financial information within Oracle Applications to analyze spreadsheet-based FSG reports in detail and perform multidimensional data analysis. While drilling down, you can use drag-and-drop capabilities to pivot accounting dimensions and quickly reorganize financial data to perform 'what-if' and 'what-happened' analysis of both summary and detailed financial information. The Analysis Wizard helps you quickly detect trends and exceptions in the financial data that underlies your reported amount.

```

<?xml version="1.0" encoding="utf-8" ?>
- <MasterReport xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:fsg="http://www.oracle.com/fsg/2002-03-20/"
  xsi:schemaLocation="http://www.oracle.com/2002-03-20/fsg.xsd">
  <fsg:SOBName>Vision Operations (USA)</fsg:SOBName>
  <fsg:ReportName>Budget Financial Income Statement</fsg:ReportName>
  <fsg:RowSetName>Income Statement</fsg:RowSetName>
  <fsg:ColSetName>Budget Columns</fsg:ColSetName>
  <fsg:RowSetId>1001</fsg:RowSetId>
  <fsg:ColSetId>2982</fsg:ColSetId>
  <fsg:ContSetId>1287</fsg:ContSetId>
  <fsg:ParamSetId>3150</fsg:ParamSetId>
  <fsg:ReportCurr>USD</fsg:ReportCurr>
  <fsg:ReportDate>1999/10/31</fsg:ReportDate>
  <fsg:RepCurPeriod>Oct-99</fsg:RepCurPeriod>
  <fsg:RepSysDate>22-FEB-2008 12:14:10</fsg:RepSysDate>
  <fsg:PeriodLabel>Current Period</fsg:PeriodLabel>
  <fsg:DateLabel>Date</fsg:DateLabel>
  <fsg:CurrencyLabel>Currency</fsg:CurrencyLabel>
  <fsg:PageLabel>Page</fsg:PageLabel>
  <fsg:NlsNumChar>.,</fsg:NlsNumChar>
- <fsg:RowContext RowId="r100001">
  <fsg:RowName>INCOME</fsg:RowName>
  <fsg:RowChngSign>0</fsg:RowChngSign>
  <fsg:RowSeq>15.0000000000000</fsg:RowSeq>
</fsg:RowContext>
- <fsg:RowContext RowId="r100002">
  <fsg:RowName>Revenue Net of Discounts</fsg:RowName>
  <fsg:RowChngSign>1</fsg:RowChngSign>
  <fsg:RowSeq>25.0000000000000</fsg:RowSeq>
</fsg:RowContext>
- <fsg:RowContext RowId="r100003">
  <fsg:RowName>COST OF SALES</fsg:RowName>
  <fsg:RowChngSign>0</fsg:RowChngSign>
  <fsg:RowSeq>35.0000000000000</fsg:RowSeq>
</fsg:RowContext>
</MasterReport>

```

Figure 3 - Example of XML Output that is used by XML Publisher

XML output files are composed of elements. Each tag set is an element. For example, in the figure above, <fsg:SOBName></fsg:SOBName> is the Set of Books Name element. The data between the brackets is the value of the element. In the figure above, the value for the Set of Books Name element is "Replaced Set of Books Name".

When a report template is designed, you define placeholders for the elements. The placeholder maps the data field to an element in the XML output file. The figure below shows the layout of the FSG: Basic Template that General Ledger provides for a FSG report.

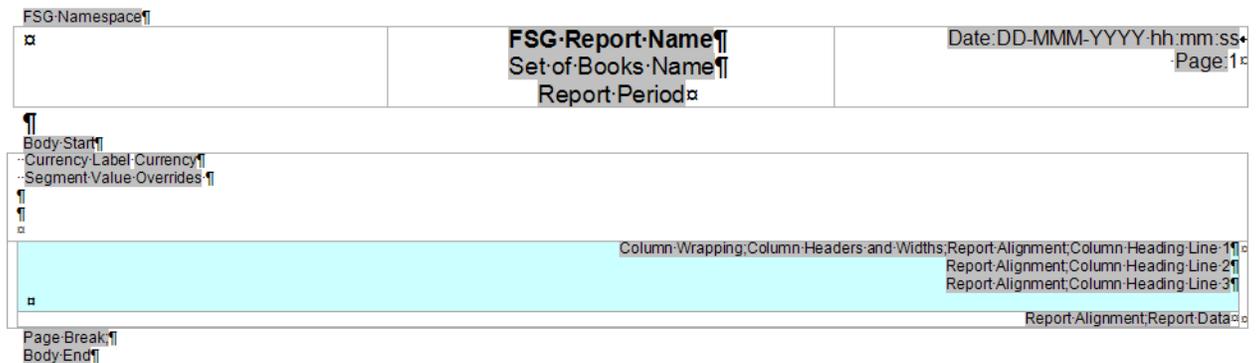


Figure 4 - Layout of the FSG: Basic Template

Placeholder Syntaxes Provided in FSG: Basic Template

The table below lists the placeholder syntaxes provided in the FSG: Basic Template.

Placeholder	Syntax	Meaning
FSG Namespace	<?namespace:fsg=http://www.oracle.com/fsg/2002-03-20/?>	Identifies the elements associated with the FSG. This placeholder does not display anything on the published report
FSG Report Name	<?fsg:ReportName?>	Inserts the FSG Report Name
Set of Books Name	<?fsg:SOBName?>	Inserts the set of books name

In addition to the seeded BI Publisher template, Oracle has provided a new Concurrent Request to generate the FSG with a template as a one-step process – “Program - Publish FSG Report”.

Figure 5 - Parameters for Standard Program "Program - Publish FSG Report"

The seeded template does not provide an easy method of formatting columns. Using the seeded template, a 2 column report will have the same formatting as a 3 column report. In order to meet the requirements of most companies, additional templates will be required.

Use of tables within BI Publisher

Introducing tables into the template allows you to control the formatting for each column. An example is provided below.

FSG Namespace				
	FSG Report Name		Date: DD-MMM-YYYY hh:mm:ss	
	Set of Books Name		Page: 1	
	Report Period			
Body Start				
Currency Label	Currency			
Segment Value Overrides				
Col1 Heading Line 1	Col2 Heading Line 1	Col3 Heading Line 1	Col4 Heading Line 1	Col5 Heading Line 1
Col1 Heading Line 2	Col2 Heading Line 2	Col3 Heading Line 2	Col4 Heading Line 2	Col5 Heading Line 2
Col1 Heading Line 3	Col2 Heading Line 3	Col3 Heading Line 3	Col4 Heading Line 3	Col5 Heading Line 3
Column 1	Data Start	Column 3	Column 4	Data End
	Column 2			Column 5
Body End				

Figure 6 - Use of tables within BI Publisher

This example provides for a report with upto 4 columns of data. The shading of the columns alternates between columns, leading to a clearer representation of the report.

Body Start

Col2 Heading Line 1	Col3 Heading Line 1	Col1 Heading Line 1	Col4 Heading Line 1	Col5 Heading Line 1
Col2 Heading Line 2	Col3 Heading Line 2	Col1 Heading Line 2	Col4 Heading Line 2	Col5 Heading Line 2
Col2 Heading Line 3	Col3 Heading Line 3	Col1 Heading Line 3	Col4 Heading Line 3	Col5 Heading Line 3
Data Start	Column 2	Column 3	Column 4	Data End

Body End

Figure 9 - Reordering Columns of the FSG

In the example above, the fields for Column 1 have been inserted between columns 3 and 4. To ease readability, the background has also been changed.

Budget Financial Income Statement			Date:22-FEB-2008 12:14:10		
Vision Operations (USA)			Page:4		
Current Period: Oct-99					

PTD-Budget	PTD-Budget	Dpt Acct	PTD-Budget	PTD-Budget
0.00	0.00	540	0.00	0.00
0.00	0.00	550	0.00	0.00
0.00	0.00	560	0.00	0.00
0.00	0.00	TOTAL	0.00	0.00
0.00	0.00	110	0.00	0.00
0.00	0.00	404	0.00	0.00
0.00	0.00	420	0.00	0.00
-58,179,440.57	-73,947,335.54	580	0.00	0.00
-58,179,440.57	-73,947,335.54	TOTAL	0.00	0.00
0.00	0.00	Depreciation Expense	0.00	0.00
-60,292.31	-76,855.06	Total Expenses	-291.09	0.00
60,292.31	76,855.06	Net Income (Loss) from Operations	291.09	0.00

Figure 10 - FSG with Columns Reordered

Conditional Font Colors

When FSG reports were generated through Microsoft Excel, users could make use of formatting that is not available in Microsoft Word. This is especially true for showing positive numbers in one color, negative numbers in another.

With some additional knowledge of XML processing, the same result can be achieved using BI Publisher.

Col2 Heading Line 1
Col2 Heading Line 2
Col2 Heading Line 3
Data Start
2,002.00

Figure 11 - Building Conditional Font Coloring

In this cell, I have two form fields – one in red font, and one in black font. The code associated with the red font is as follows:

PTD-Budget	PTD-Budget	Dpt Acct	PTD-Budget	PTD-Budget
0.00	0.00	520	0.00	0.00
0.00	0.00	530	0.00	0.00
0.00	0.00	540	0.00	0.00
0.00	0.00	550	0.00	0.00
0.00	0.00	560	0.00	0.00
0.00	0.00	TOTAL	0.00	0.00
0.00	0.00	110	0.00	0.00
0.00	0.00	404	0.00	0.00
0.00	0.00	420	0.00	0.00
-58,179,440.57	-73,947,335.54	580	0.00	0.00
-58,179,440.57	-73,947,335.54	TOTAL	0.00	0.00
0.00	0.00	Depreciation Expense	0.00	0.00
0.00	0.00		0.00	0.00
-60,292.31	-76,855.06	Total Expenses	-291.09	0.00

Figure 14 - FSG with conditional font coloring

FSGs, BI Publisher and Oracle Report Manager

Oracle Report Manager is an online repository for storing and viewing reports. It provides a secure and convenient location to access point-in-time reports published or uploaded to the database.

Report Manager users can either be report producers or report consumers. Report producers can submit, publish and set security for reports. Report consumers can view, download and approve reports.

Reports can be published and presented to certain responsibilities without additional security or a variety of security models can be applied, allowing only authorized users to view entire reports or parts of reports.

Overview of Financial Report Publishing

You can submit any FSG report available to your responsibility. The publishing parameters determine the template to be applied to format the report, the security to restrict access to the report, and when the report becomes available. These templates are XSL:FO based, so any text editor can be used to create them. Templates can be created as RTF files using BI Publisher. More importantly, Microsoft Excel templates are provided in Oracle Report Manager.

You can create or edit FSG templates using Microsoft Excel. These templates are stored in the database as Excel files for editing and as an eXtensible Stylesheet Language Formatting Objects (XSL:FO) document. The style information from the XSL-FO template is applied to the data in the report and controls the formatting of the data.

When you create a new template for a specific report, the template editor retrieves the structure of the chosen report and provides standard static data at the top of the report template in Excel, as well as row and column details. The template that is generated in Excel is fully functional. It can be uploaded immediately to the template manager using the Oracle menu and used to format output for the report for which the template was created.

Select the menu item "Financial Report Template Editor" to build a new template. A new template is fully formatted and ready to work with. When creating a new template, or editing an existing template, you can make the following changes:

- Add and remove report elements
- Change the location of report elements on the page
- Modify row and column sizes
- Use standard Excel formatting, such as bold and italics

Release 12 – BI Publisher is here to stay

With Release 12, more of the reporting is being moved to BI Publisher. Out of the box, more than 870 templates for more than 60 modules are included with baseline release 12. Oracle is currently migrating the remaining reports to BI Publisher, and later this year, more than 2000 templates will be available.

Release 12 also includes a set of libraries to help move your custom Oracle reports to a data template (extract) and a layout template (RTF). These libraries are the same that are being used by Oracle to convert the canned reports.

For current release 11i customers, there are rumors (<http://blogs.oracle.com/BIpublisher/2007/02/23#a163>) that Oracle is considering building a template bank on Metalink to allow the download of Oracle developed templates. Release 5.6.3 of BI Publisher will also include the migration libraries.

Conclusion

Classic reporting approaches combine the data definition, the layout format and translation in a single source. This leads to large numbers of report files; a single data definition (query) could support several layouts but due to the nature of the report definition file the user has to create and maintain a separate report for each combination. Now add the required translation and the number of report definitions has become huge.

Typically the tools used to create these reports are powerful but complex and require a high level of technical training to be used effectively.

If a company takes a development delivered report and has to customize it the only option they have is to take a copy of that report definition, make the changes they require and then use it. If development upgrade the base report then any changes will need to be propagated through to the customized report.

These three factors lead to an increase in maintenance and customization costs and an extremely difficult and complex upgrade.

Oracle BI Publisher simplifies the whole customization process and combines this with reducing costs. The traditional approach of combining the data definition, format and translation in a single entity is thrown out in favor of breaking the three components apart. With the three pieces now separated the whole model is simplified.

- Data Definition - The data definition exists as a single entity, an Oracle Report, PL/SQL package, Service Bean, etc but it now just becomes an 'BI Data Engine' that can service not only a reporting need but can fulfill other requirements e.g. B2B communication along with other reports.
- Report Templates – The technology behind BI Publisher is a W3C standard, XSL-FO. There are many XSL editors currently in the market that can be used to create report formats. BI Publisher also allows the user to create layouts using Microsoft Word, Microsoft Excel and Adobe Acrobat, these familiar desktop tools make report template design and much simpler proposition for users and can be designed comfortably by business analysts rather than an IT specialist.
- Translation – BI Publisher is able to extract the report boilerplate to an XLIFF format. These files can then be translated by 3rd party translation companies.

At runtime the three components are brought together by BI Publisher to generate the required output.

All parties win from report developers to the end user when using Oracle BI Publisher.

End User – The end user is presented with a richer report offering from the e-Business Suite. Reports in multiple formats with multiple delivery options are now available directly from the applications suite; the language and locale options have been improved and this all comes through a consistent user interface.

