

# BI Publisher

## Seven Small Steps for a Consultant, one Giant Leap for Userkind

By Simon Tomey, Oracle Functional Consultant

Oracle BI (XML) publisher is a rich new piece of functionality in E-Business Suite Financials and one of the easiest to implement. You can forget the text based reports you have been struggling with and delight your users with clean, attractively formatted documents and spreadsheets.

This is a story (fictional) about how your organisation could apply BI publisher to good effect to its reports. The article gives an outline as to how you could do the same and concludes with where to get more information.

### Delighting your users with BI publisher...

“What’s the point of paying all this money for a system, if we can’t even get output we can send to our customers?”, said Colin the credit controller.

I had implemented receivables some months ago and we used third party software to format the text based dunning letters from Oracle. When I designed the customer number, it seemed like a good idea to base it on an existing reference which gave a length of 12 characters. I had checked the length of the field in Oracle and it had worked very well with the rest of the team. Only now did I discover that the text based output only printed the first 11 characters.

“The director of marketing is going to make an issue of this one”, continued Colin in his inimical way – it’s KOLRC! (Kicking off left right and centre).

I never really liked the idea of using third party software. We had to buy another print server to host it, the set up didn’t really work and no one else knew how to configure it. I remembered a conversation with Oracle support where they mentioned “This thing we have called XML publisher.” I wonder if I could get that to work. It’s integrated and it’s supported.

That lunchtime I went to the library to get some books on XML. “XML for idiots.” Well, I felt like an idiot at that moment; what was I going to do about the customer number on the dunning letters? My wife Gilly called that evening; “the kids are desperate to see you”.

“I have to stay a bit later tonight, something’s come up,” I replied, distracted by my thinking, “XML is all very interesting; a very flexible, self defining language with tags, but I just don’t see how it fits with Oracle – the files don’t even work without a schema”.

Gilly interrupted my thoughts. “We need to decide about the extension, we can’t

have three children in one bedroom much longer, but the builder said it’s going to be expensive”.

“Yes, of course, we need to do something soon” I replied. Goodness if I can’t sort this out, that ‘something soon’ might be going down to the Job Centre.

The next day I went to the Finance meeting. Surprisingly for a Finance meeting, I actually looked forward to these meetings. Jack, the Head of Finance, and Sarah were enthusiastic users of Oracle and they always appreciated what I could do, which made me feel better. I was trying to make sense of the XML user guide, so got to the meeting a little late.

“The meeting was getting worse; it was not what I was looking forward to...”

The new girl Janice was in full flow... “When I worked for the Big Corporation, they had SuperFinanceSoftware and it was great. In my last company, you could download everything you wanted from the GL straight into Excel... I don’t understand why we’ve got Oracle. The reports don’t print properly and when you copy them into Excel they go all funny. While the RX reports are better, it is a hassle to run them and the description that we type into the GL gets cut off at only 15 characters so I can’t get all the information.” Clare chipped in “Yes, I’ve got a problem with reports too... Marketing need a list of payments by ‘invoice paid’, but I don’t know how to get this”. The meeting was getting worse; it was not what I was looking forward to, to cheer me up....

Later that afternoon, Jack took me to one side. Jack was very good with people and knew how to encourage others when they were struggling. Perhaps he had news on my promotion and I could tell Gilly to go ahead with the extension to the house.





“Michael, I’ve had a chat with John.” Not only was John the Finance Director, but he had influence at Group Headquarters. “It seems a number of users have been going over my head and complaining about the issues with the Oracle reports... John has given me a deadline of the end of the month to get this sorted, or he will be looking to outsource the Oracle support. We need to deliver something for him, or I’m afraid we’re going to have to put you on the re-deployment register,” Jack continued. “I’m a bit worried about how hard you’ve been working recently. I’m supposed to be going to the UK Oracle User Group Financials SIG tomorrow, but perhaps you want to take tomorrow off to attend in my place”.

The rest of the conversation disappeared into a mist as I tried to maintain my composure. Well at least a day out of the office will be a good chance to think. Perhaps I’ll get some sympathy from others at the SIG – unlike my users, they understand the difficulty of the standard Oracle reports. Perhaps I’ll get some leads on other jobs going.

“Financials”; “Special”; “Interest”: they’re words that don’t go together easily...” I joked with the taxi driver as he dropped me off. My interest was piqued by one of the agenda items, “Transforming your reports with BI (XML) publisher – possibly the richest and easiest to implement new piece of functionality.” I wondered, could this help me?

The presenter seemed to have great enthusiasm for the subject and opened by saying “Using BI (XML) publisher isn’t easy, but it’s easier than it appears”. He asked for a show of hands “Who is actually using BI (XML) publisher?” (very few hands) “...And who thinks they’d like to implement it but have been put off because they don’t know how to start?” (over 50 people put their hands up).

“What I hope you’ll get today is enough knowledge to be able to get BI (XML) publisher working on at least some of your GL reports, but I’ll also be using ‘Receivables customer open balance letters’ as an example”.

I jotted down notes from the presentation as follows:

### The Seven Steps to Transforming your Reports with BI (XML) publisher.

#### Step 1 – Chose a report and save the output.

Chose your first report, run it and save the text based output somewhere. Make a note of the report code (see the log file). E.g. customer open balance letter in receivables (ARCOBXL).

#### Step 2 – Change the output format to XML

Log in to systems administrator, concurrent programs, define and change the output format from “text” to “XML”.

#### Step 3 – Re-run the report and save the XML file

Go back into the responsibility as before (e.g. receivables manager) and re-run the report. The report will now be produced as

an XML file. Open the output and save it as an XML file somewhere where you can find it again (tip: put the report code in the file name to make it easier to find).

#### Step 4 – Create an RTF template

Use MS Word (or similar program that can create RTF) to create a template. It’s easiest with the BI publisher add-in, but you can do it “manually”.

1. Set up your template so that it looks like the output you want. Include formatting and logos, watermarks, signatures, etc.
2. Put something in which will remind you where you want items to be filled with live data. E.g. put address line1 where you want address line1.
3. Open the XML output you saved in step 3 and put the tag names into the document where you want the live data to be. Surround the tag names by <??> e.g. <?address\_line1?>.
4. Save the file as RTF (not \*.doc!)

#### Step 5 – Link the Oracle output to a data definition

Login to XML publisher administrator and create a data definition (home, data definitions, create). Fill out the boxes marked by asterisks and save the XML output file (saved earlier) as “preview data”. Chose a sensible “name” (free text field), type the report code into “code” (this must be exactly as per the report), application is your module (e.g. receivables) and a start date of today or earlier.

#### Step 6 – Link the Oracle output, the data definition and your RTF template to a “XML template”

Stay in XML publisher administrator and create a template (home, templates, create). Fill out the boxes marked by asterisks and save the RTF file (created earlier) as “file”. Chose a sensible “name” (free text field), type the report code into “code” again (this must be exactly as per the report and as typed into the data template), application is your module (e.g. receivables) and a start





date of today or earlier, language is “English”, territory is “United Kingdom” (assuming you are in the UK and communicate in English). Click “apply” and wait for the message saying that your template has been created. Preview what you have done by clicking the preview icon.

### Step 7 – run the report

Go back into the previous responsibility (e.g. receivables manager) and run the report. The output should now be produced as per your template (and as per the preview). If you get a message saying “unable to find the published output for this request” and that “no output exists for the request nnnnnn”, don’t panic – it’s just that this report needs to be published via the two step process. (See my slides (number 87) or see Metalink Note ID: 364547.1).

I set aside the next day to see if I could replicate the presentation on my test environment. It took me most of the

day to get the hang of it, but once I had done that I tried the same principles on dunning letters.

“These dunning letters are great,” said Colin once I showed him the test output. “When can we implement to live?”

Not only were the letters able to display the full content of the fields, but the letters appeared much clearer and include logos, tables and signatures. I used the standard dunning letter output, so the text could be changed using the normal maintenance screen, and the formatting is easily modified via the RTF template. Best of all, they can be produced as PDF and emailed to a mailing house to print and send.

Encouraged by the success with the dunning letters, I set GL “account analysis report” to run with the Oracle pre-loaded template. Instead of having to copy paginated text into Excel and fiddle with it to clean it up, users could now paste the GL account analysis directly into Excel. Best of all, the full field content was displayed, so

whatever was entered into the journal would appear directly in the Excel report.

The next day John called me into his office. “Michael, Head Office has heard what we have been doing with XML publisher and is impressed. They are talking about setting up a team to migrate what you have been doing to all the other companies in the group. It seems that the money which we will save on maintaining our old Oracle reports will easily fund another two posts, not to mention the time we are saving in our division in not having to populate our MI reports. Our customers are impressed by the presentation of the documents we send out, the board members like the reports we give them and we’re well ahead on converting our old reports for the planned R12 upgrade. Would you consider leading the team – it will mean more responsibility, but of course that will come with the appropriate rewards.”

The End ... (or is it just the beginning?).

## About the Author



**Simon Tomey** is a chartered accountant and Oracle Functional consultant. Simon has significant experience of Oracle BI publisher and is an enthusiastic advocate of its practical application. He would be delighted to hear from anyone with questions.

[Simon.Tomey@BeLife.co.uk](mailto:Simon.Tomey@BeLife.co.uk)

**Note:** BI publisher used to be called XML publisher and is part of the functionality available from the whole Oracle BI Enterprise package. You do not need the whole BI package to transform your Oracle financials reports, as the requisite parts are integrated into Oracle Financials (and other applications).

For more details on the seven steps to setting up your reports in XML publisher you can download a recent presentation from UKOUG’s January E-Business Suite Financials SIG. See [http://www.ukoug.org/calendar/show\\_presentation.jsp?id=8167](http://www.ukoug.org/calendar/show_presentation.jsp?id=8167) or [http://www.belife.co.uk/OracleR12\\_XML\\_Publisher.html](http://www.belife.co.uk/OracleR12_XML_Publisher.html) (username “OracleScene” password “Summertime”).

Developers,  
Look No Further.

Oracle  
Development  
Tools  
User Group

WWW.ODTUG.COM

A Real World User Group for Real World Developers.

ODTUG understands the needs of developers and we deliver - real world support, direct expert access, truly technical resources and seriously practical events. If you're a developer in the trenches, visit us at the User Group Pavilion - 2nd Floor Mosconi West to receive a free CD of 2007 Conference Papers. Then join us in June 2008 for a day of seriously practical community service in New Orleans at Kaleidoscope 2008. Visit [www.odtug.com](http://www.odtug.com) or [www.odtugkaleidoscope.com](http://www.odtugkaleidoscope.com) for more details.