Engineering Management Field Project

BI (XML) Publisher Conversion from Third Party Software in E-Business Suite: An ERP (Enterprise Resource Planning) Reporting Framework Conversion Model

By

Shuai Wang

Spring Semester 2012

An EMGT Field Project report submitted to the Engineering Management Program and the Faculty of the Graduate School of the University of Kansas in partial fulfillment of the requirements for the degree of Master of Science.

_____

Herbert Tuttle,   **Date**
Committee Chair

_____

John Bricklemyer   **Date**
Committee Member

_____

John Conard   **Date**
Committee Member

Date accepted: _____

Acknowledgements

First, I would like to thank God for guiding me as I pursued my studies in Engineering Management.

Second, I would like to thank my company for supporting me financially.

Third, I would like to thank the entire staff of the Engineering Management program at The University of Kansas, Edwards campus who provided this excellent educational opportunity. I am grateful to have Herbert Tuttle as my mentor and committee chairperson, and John Bricklemyer, and John Conard as committee members. They have provided me with their excellent opinions throughout the project and have guided me to complete this project in the specified time.

Finally, I lovingly dedicate this paper to my wife, who supported me each step of the way. I am extremely blessed to have such a wonderful wife who will be the mother of my twin girls in July.

Table of Contents

Table of Acronyms

EBS - Oracle E-Business Suite

EDI - Electronic Data Interchange

EFT - Electronic Fund Transfer

BIP - Business Intelligence Publisher

ERP - Enterprise Resource Planning

CRM - Customer Relationship Management

HTML - Hyper Text Markup Language

JVM - Java Virtual Machine

MICR - Magnetic Ink Character Recognition

OOP - Object Oriented Programming

RDF - Resource Description Framework

RTF - Rice Text Format

XML - Extensible Markup Language

XSL - Extensible Style Language

Executive Summary and Overview

Oracle Business Intelligence Publisher (BI Publisher) is an enterprise reporting framework to develop, manage, and deliver all types of highly formatted documents. It eliminates the need for costly point solutions. End users can easily design report layouts directly in a Web browser or using familiar desktop tools, dramatically reducing the time and cost needed to develop and maintain reports. In addition, it is extremely efficient and highly scalable because it can generate tens of thousands of documents per hour with minimal impact to transactional systems. Furthermore, it is "a template-based publishing solution delivered with the Oracle E-Business Suite, PeopleSoft, Enterprise and JD Edwards EnterpriseOne" ("Business Intelligence Publisher Core Components Guide," 2008).

Today, many companies who are using third-party ERP reports framework want to convert their reporting framework to Oracle BI Publisher framework because they want to reduce the overall cost of development, customization, and ongoing maintenance of their ERP reports. However, converting third party software to BI Publisher is not easy. It is technically challenging, can be costly, and can even fail if the project is lacking a thorough plan and careful implementation.

This paper exhibits a case study and constructs a step-by-step conversion model for others to follow. The intended audiences are those companies who

are planning on converting their reporting framework to BI Publisher in the Oracle

EBS (E-Business Suite) environment.

CHAPTER 1

Introduction

Without any consulting services, Company XYZ converted approximately seventy percent of its ERP (Enterprise Resource Planning) reports to BIP from third-party software (Appendix C). During this conversion process, Company XYZ faced both business and technical challenges such as users' resistance, multiple language/region issues, monitoring of the bursting and delivering performance, print driver setup, and MICR (Magnetic Ink Character Recognition)/barcodes font selection.

This paper shows how the Company XYZ's ERP team handled these challenges during the conversion process. More importantly, it exhibits a case study and constructs a step-by-step conversion model for others to follow. The intended audiences are those companies who are planning on converting their reporting framework to BI Publisher in the Oracle EBS (E-Business Suite) environment and those software engineers who are working closely with enterprise level ERP systems.

About Company XYZ

Company XYZ was founded in 1989 with over 9,000 employees spread across the United States, Europe and Asia. In 2010, the company brought in $2.69 billion in sales with 16 million units sold.

CHAPTER 2 – Literature Review

About ERP

ERP (Enterprise Resource Planning) is an integration of information technology and business practice. One of the primary key objectives of an ERP system is to integrate people, data and processes from all business functions vertically and horizontally. In addition, an efficient ERP system improves business productivity, employee's performance, and empowers an organization to dynamically align the business processes to its strategic objects. Business functions such as order management, account receivable, human resource management, CRM (Customer Relationship Management) are chained by ERP. "In simpler words, an ERP is a massive software architecture that supports the streaming and distribution of geographically scattered enterprise information across all the functional units of a business house. It provides the business management executives with a comprehensive overview of the complete business execution, which influences their decisions in a productive way." (Tech-FAQ. 2012)

Some companies have decided to implement their own ERP framework because they want to have full control of the ERP system and make sure that the ERP framework is 100 percent tailored to the business process. Other companies prefer to purchase a pre-built ERP system and customize it to meet their business specification. Regardless, an ERP system needs to have reporting functions which generate ERP reports such as invoices, purchase orders, and pick slip to support daily business activities.

ERP Reports

Any documents generated from an ERP system and used for reporting purposes are considered ERP reports. Some types of documents are used by an internal functional group to assist in its daily operational activities. For example, a packing list report (Appendix C) assists a warehouse shipper to pack required components of a customer order on the shipping station. It describes the contents of the order by size, color and quantity as well as the date the order was received and shipped. The document also explains any changes to the order such as back orders or out-of-stock items.

Other types of documents are sent to the external customers for communication purposes. For example, a purchase order report (Appendix C) is a document that specifies products and/or services ordered from a specific vendor. It is usually delivered according to contractual agreements with a supplier, specifying payment terms, delivery dates, item identification, quantities, and freight terms and all other obligations and conditions. Both the packing list report and purchase order report can be generated and printed manually or automatically from an ERP system. In addition, other ERP documents can be utilized by internal users for data analysis purposes such as monthly sales reports and ledger consolidation reports.

These reports, such as the packing list report and purchase order report, which are sent to the external customers for communication purposes, need to

be designed carefully. Providing precise information and displaying the data in a professional layout are very important when designing the report.

ERP Report Development Framework

The development of enterprising reports is an inevitable process. Regardless of which ERP software a company is using, customizing seeded reports, developing new reports, and enhancing existing reports are part of a developer's daily activities. Previously, it was difficult for developers to write and maintain programs without having efficient report development software. Developers needed to write every line of code to cover data source retrieval, report layout formatting, and delivery functions in the programs (Figure 1).
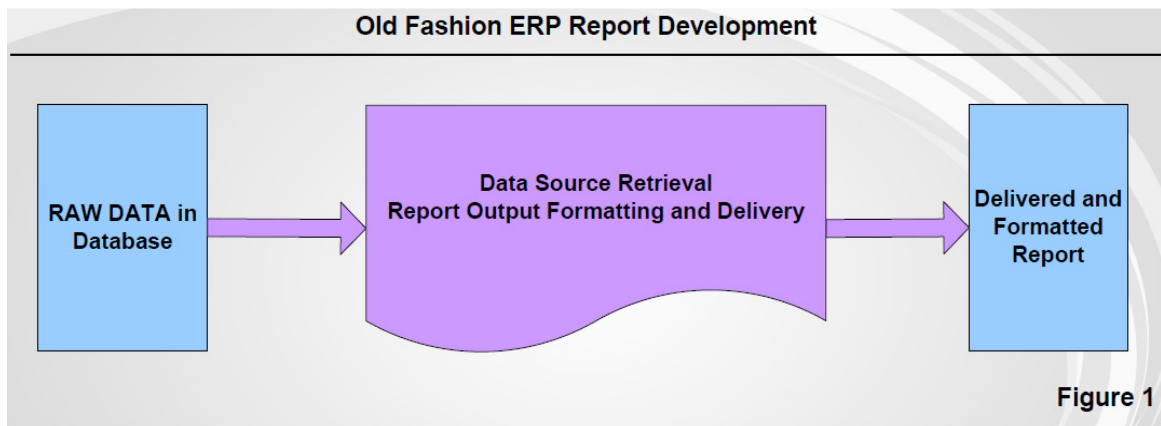
Figure 1 - Old Fashioned Method of ERP Report Development (Shuai Wang 2011)

In recent years, many software tools aimed at assisting the development of ERP reports have emerged. Crystal Reports and Formscape are two examples. Upon scrutiny of the structure of such applications, it becomes

apparent that they utilize similar methods (Figure 2) by introducing a task-oriented structure. Developers' tasks are divided into data source retrieval, report output formatting, and report delivery. XML (Extensible Markup Language) technology makes this segregation possible.
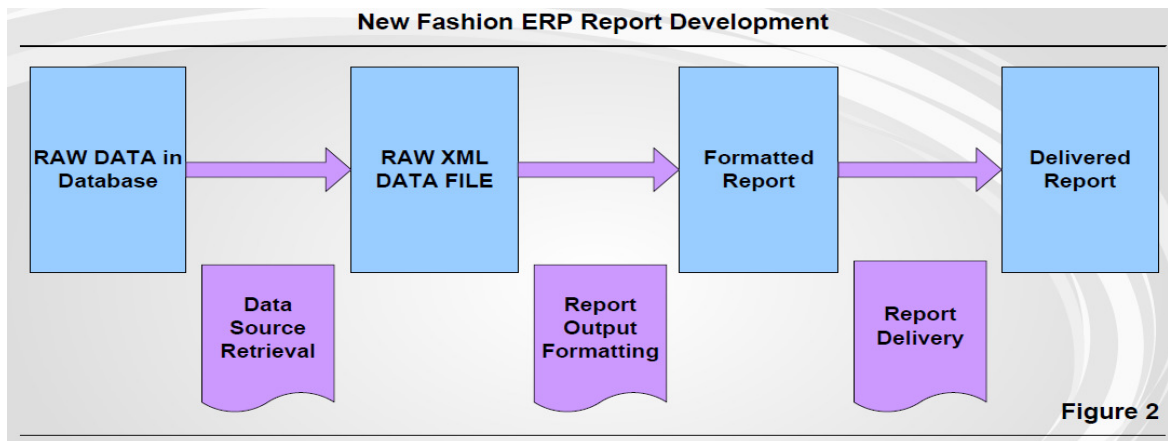

Figure 2 - New Method of ERP Report Development (Shuai Wang 2011)

Oracle EBS Report Framework

Oracle EBS Report framework is Oracle's first attempt to simplify the report development process. Its report builder provides a visual aid (Paper Layout) for a developer to format the report and separates this task from data retrieval (Data Model). The data model within Oracle Report Builder is very easy to learn and use. More importantly, it can produce an output file in XML format (Figure 3).
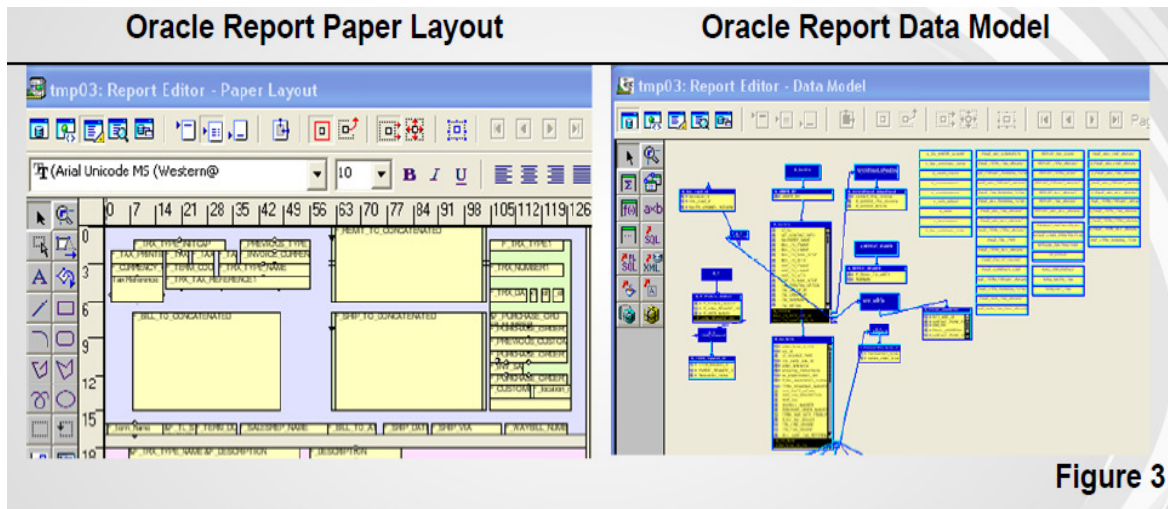
Figure 3 - Old Oracle Report Paper Layout and Data Model (Shuai Wang 2011)

However, based on the author's own experience, the paper layout function is difficult to learn, tedious to use, and very taxing to maintain. The same sentiment was expressed by Tim Dexter in his blog: I was amazed at just how bad the reports were...Customization! It was painful, we never shipped quite the right report to you folks, there were fields you wanted and had to be added or removed, layouts needed changing - as soon as you touched that report - it was yours to maintain." Furthermore, the common Oracle report error "Signal 11" drives developers crazy.

CHAPTER 3

Project Background

Several years ago, Company XYZ implemented the Oracle EBS (E-Business Suite) as its primary ERP system. At the same time, the company also purchased Formscape (third-party software), as its main ERP reporting software.

The ERP team was able to integrate Formscape with the Oracle EBS report system. Company XYZ was not the only company seeking third party software solution. According to the Oracle E-Business Suite Development & Extensibility Handbook, "the distribution capabilities in the Oracle Reports tool within E-Business Suite are very restrictive. To distribute the output of Oracle Reports, many companies implementing E-Business Suite have to purchase third-party products or even build their own in-house reports distribution framework" (176).

In contrast with the paper layout function in the Oracle Report builder, Formscape makes it much easier for developers to build a report layout. It allows developers to work in a "what you see is what you get" (WYSIWYG) fashion. Anytime during the design process, developers are able to test the work, preview, and print the output as the work progresses. Additionally, no traditional programming background is required for a Formscape developer. Finally, it can deliver the report by fax, email, print, file transfer, or other methods.

The Formscape software runs on a standalone server. It receives raw data in XML format by registering its processes as virtual printers to Oracle concurrent programs. The approach was to use Oracle Report for data source retrieval, and use Formscape for formatting and delivery. Since implementing Formscape, Company XYZ has developed over 100 business critical reports using the combination of Oracle Report and Formscape processes.
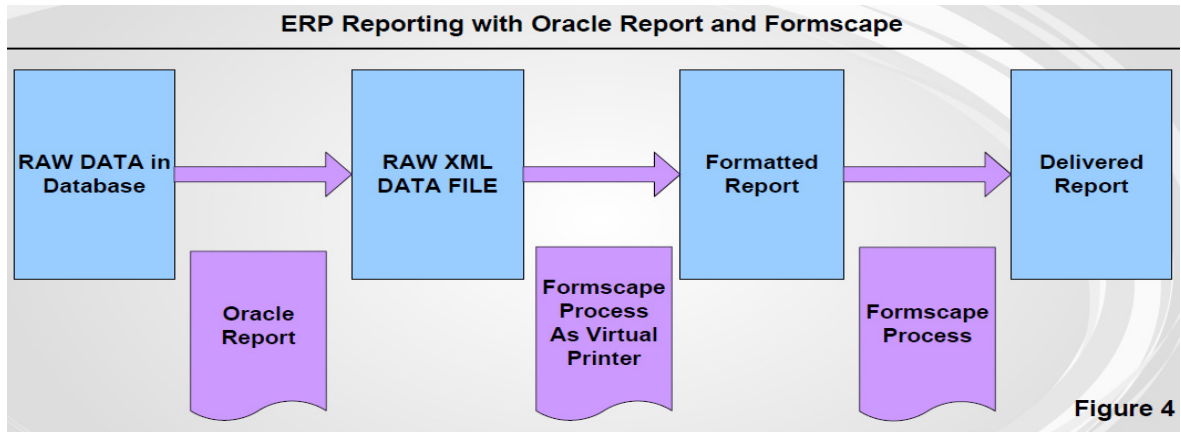
Figure 4 – ERP Reporting with Oracle Report and Formscape (Shuai Wang 2011)

Problem - Statement and Solution

Despite the advantages of Formscape, there are also drawbacks. First, it costs the company to pay maintenance and support fees annually. Second, it takes double the effort to install print servers and printers to both Oracle and Formscape. Third, it relies heavily on the network and print server performance, thus requiring a Formscape server administrator to be on duty constantly. Fourth, it is not fully integrated with the Oracle EBS system. This limits the user to view output in XML format only. If the user wants to view the actual output, he or she needs to either email or print the report. Last, when pages are missing, the users have to re-print the entire batch. There is no way to selectively print missing pages. With these drawbacks in mind, and after evaluating BI Publisher technology, the ERP team decided to convert Formscape to BI Publisher.

The ERP team learned that BIP had overcome these drawbacks while still maintaining the impressive features of Formscape. The team decided to convert

Formscape to BI Publisher. According to XML Publisher User Guide Release 11i, Oracle XML Publisher is a template-based publishing solution delivered with the Oracle E-Business Suite. It provides a new approach to report design and publishing by integrating familiar desktop word processing tools with existing E-Business Suite data reporting. XML Publisher leverages standard, well-known technologies and tools, so the user can rapidly develop and maintain custom report formats.

The flexibility of XML Publisher is a result of the separation between the presentations of the report from its data structure. The data collection is still handled by the E-Business Suite, but now the user can design and control how the report outputs will be presented in separate template files. At runtime, XML Publisher merges the designed template files with the report data to create various outputs to meet a variety of business needs, including:

- Customer-ready PDF documents such as financial statements, marketing materials, contracts, invoices, purchase orders utilizing colors, images and font styles, headers and footers, and many other formatting and design options.
- HTML output for optimum online viewing.
- Excel output to create a spreadsheet of report data.
- "Filled-out" third-party provided PDF documents. Users can download a PDF document, such as a government form, to use as a template for their report. At runtime, the data and template produce a "filled-out" form.

- Flat text files to exchange with business partners for EDI and EFT transmission.

CHAPTER 4

Project Plan

A thorough plan was essential for this project to be successful. The scope, schedule and resources were clearly defined.
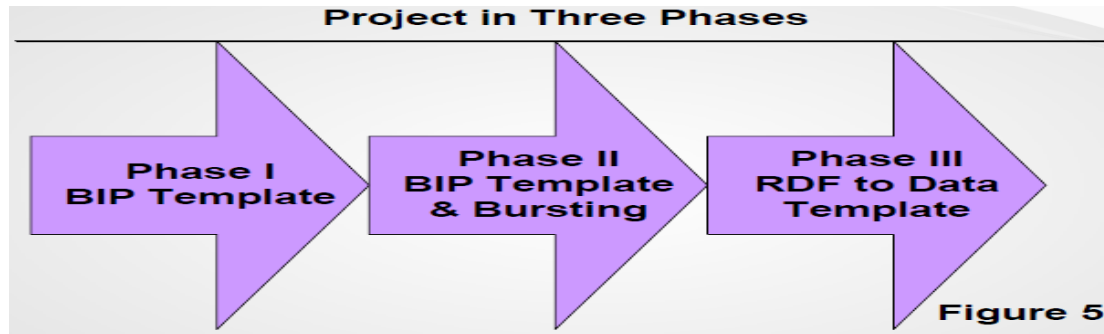
Figure 5 – Project in Three Phases (Shuai Wang 2011)

For the resources, one project manager, three analysts and four ERP developers were assigned to work on the project. Since the project was built to be a long-term solution and was not considered a business critical project, the analysts and developers allocated 15% to 20% of their time to work on it. For the scope, all the in-house developed ERP reports by Formscape needed to be converted. For the schedule, the project team decided on a three-phase approach and planned to complete the project in three years.

The first phase was to collect an inventory of in-house developed ERP reports (Appendix B) and to convert these reports used for printing purposes only. In this phase, developers only needed to focus on creating BIP templates.

The second phase involved converting reports that were used for multiple delivery purposes such as email and FTP. During the second phase, in addition to creating a BIP template, developers also needed to focus on building bursting function. Bursting function is used to break a single large report file into multiple smaller reports based on a bursting key and distribute these reports by email, fax or print to different receipts or systems.

The last phase entailed converting all Oracle RDF programs to BIP Data Template files. In this phase, developers would need to learn how to use the Oracle Report Migration/Conversion Utility and be familiar with the BIP Data Template structure.

Challenges during Project Implementation

The first challenge encountered was the business users' resistance. In theory, we could make the project transparent to users, but in reality, this was almost impossible without their support and cooperation. Even though people are accustomed to change at Company XYZ, the implementing team still had to be careful with how they presented this change to stakeholders. In the beginning, this project was perceived to be unnecessary work by some of the business users. From their point of view, there were already reports meeting their needs

and they did not want to spend more time assisting in the development of similar reports.

One way that helped mitigate the resistance was ensuring that stakeholders were involved at an early stage in the project. We constantly provided project updates and BIP technology education. This greatly reduced stakeholders' fears. Also, we picked a relatively simple report to convert for each user group at the beginning, thus helping to build confidence in both the development team and business users. Today, users cannot wait to convert the rest of the reports to BIP because of their previous good experience with BIP.

The second challenge was to deal with multi-organization and multi-language scenarios. Company XYZ is an international company. Therefore, it was appropriate to build an ERP report in different languages for different operating units. The choice between developing multiple BIP templates and building a XLIFF (XML Localization Interchange File Format) translation file is debatable.

As an example, the pack slip report (Figure 6) presented challenges for. business units in Italy, France, and the UK. These business units wanted the same report in their respective languages. It seemed like a good idea to provide a single BIP template and use an XLIFF file to translate the languages. However, it did not take long to learn that each business unit also wanted a slightly different format and additional data fields. The final decision was to build a template for

each business unit and create an agent program as a controller to use the template accordingly.
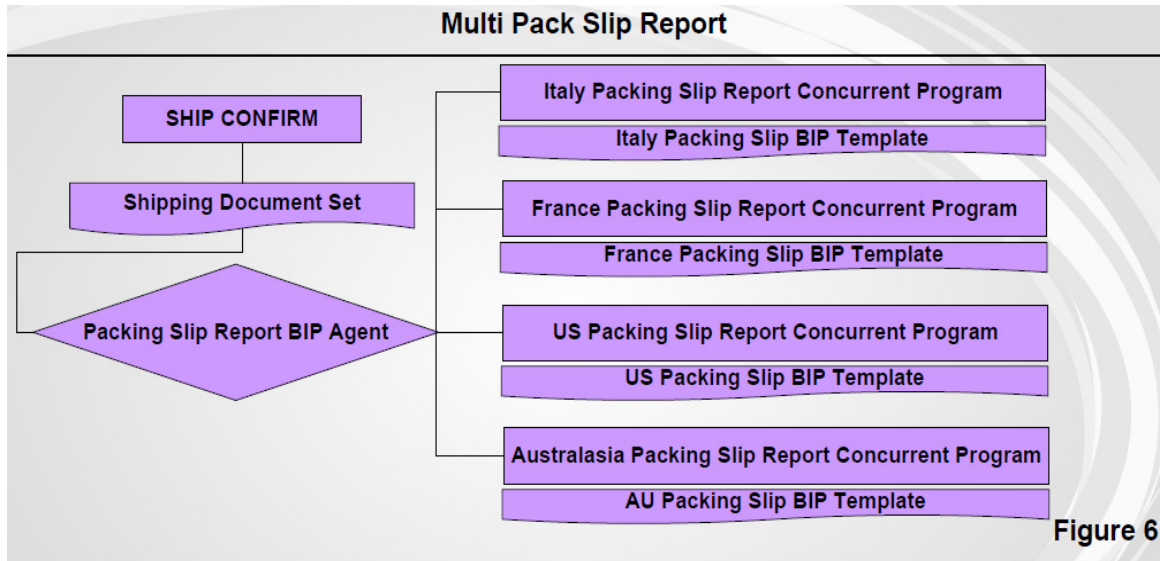
Figure 6 – Multi Pack Slip Report (Shuai Wang 2011)

The third challenge was to generate a legitimate check with MICR (Magnetic Ink Character Recognition) font. The company generates its own checks from the Oracle ERP system on a weekly basis. This program is critical because check payments cannot be delayed. Surprisingly, the development of this project was easier than expected. We bought MICR font from Elfring Soft Font Inc, and registered the font through the XML administrator responsibility. It was then ready to be used. The only obstacle we encountered was that our developers spent a huge amount of time adjusting the MICR position on the check.

The fourth challenge was to learn how to generate barcodes because many of our warehouse labels have different requirements. For the barcode 39, it was easy enough. We bought the Morovia barcode font and registered it to the "font family" through the Oracle XML administrator. For other advanced barcodes that required encoding, it took many hours to research and experiment. We learned how to register a barcode encoding class with XML Publisher, and instantiate the class at runtime to carry out barcode formatting.

In summary, the ERP team handled these challenges during the conversion process very well. The business users were happy with the outcome. Each organization from a different country was able to generate ERP reports in its own language. The team's worldwide ERP system generated checks with MICR font and labels with barcode smoothly on a daily basis.

Lesson Learned during Project Implementation

During this project we learned that whenever possible, a sub-template should be used for a logo, and company information such as addresses and contacts should be pulled dynamically. Developers should not hardcode anything in the BIP layout template even if their business user promises that it will not be changed for 100 years. Over the past few years, Company XYZ's logo has been changed, some of its subsidiary business units have decided to use a different logo, and the parent company moved from the Cayman Islands to Switzerland. All those changes caused the development team extra work to

update reports because the address and business contact information was hardcoded and they did not use sub-templates for the logo.

The OPP log file is a helpful troubleshooting resource. It should be checked first when there is an issue related to BIP concurrent programs. It provides detailed run-time information about BIP engine post-process. Many exceptional error messages are not recorded in the concurrent program log file. On the other hand, error messages such as "Java Heap run of memory," "PDF2PS service stop working," and "Not enough temporary space" are recorded in the OPP log file. This simple script below to find the OPP Log location will save the user time. Other techniques for BIP trouble shooting in Oracle EBS are addressed in the article "Troubleshooting Oracle XML Publisher for the Oracle E-Business Suite [ID 364547.1]" in Oracle Metalink. The article is a must-read technical document.

Whenever the BIP bursting function is called, it generates an empty folder on the server temporary directory. As the number of directories grows indefinitely, it could cause a space issue. The issue can be managed by running a script to delete those folders periodically.

When there is a JVM (Java Virtual Memory) issue when running BIP concurrent programs, the heap space may need to be increased by modifying the option field to the concurrent program definition. There is a good article [ID 1268217.1] on the Oracle Metalink site. It provides everything you need to know

about Java heap size. In addition, it is always a good idea to do stress testing on the concurrent program to make sure Java memory is big enough.

The XML encoding issue, "invalid characters" is problematic. Since Company XYZ business units are spread across the United States, Europe, and Asia, many reports are needed to handle the data in special characters for languages such as Icelandic, Latin, and French. Setting the encoding from "UTF-8" to the xml file does not always work. For example, when a data file is encoded by "ISO 8859-1 West European" character-set and imported through EDI (Electronic data interchange), the "UTF-8" will not be able to encode the characters correctly. The list below provides useful guidelines to developers regarding XML data encoding:

- Always use the encoding attribute and know the XML data character-sets.
- Use an editor that supports encoding to view the XML output.
- Make sure to know what encodes the source data.
- Make sure to use a real source data when developing a BIP template.
- Use the same encoding from the source data generation in the encoding attribute.
- Use the EBS Values Set "ECE_CHARACTERSET" as a reference.

CHAPTER 5

<u>Project Technical Skills Set</u>

For Company XYZ, the conversion project has been successful without any consulting services because its ERP developers are technically competent. The chart below (Chart 1) highlights the technical areas a BIP developer should master. The items highlighted in purple represent challenging tasks that require considerable knowledge from developers. In contrast, those items highlighted in blue are relatively easy to learn. This developer skill-map was built by following a standard BIP report development process in the EBS Environment. To start with the top and left box, a developer needs to know:

- Either Oracle Report Data Model Function or Data Template Structure, and the conversion utility to switch Oracle Report to DT
- How to define a concurrent program and request group
- XML and its Encoding technology
- BIP layout template and knowledge about XSL
- How to define Template and Data definition by login into XML publisher administrator responsibility
- The bursting file and xdodelviery.cfg structure
- How to setup print server in Pasta.cfg or in xdodelivery.cfg
- How to use FNDLOAD and XDO_LOADER to do the deployment
- How to debug when there are issues related to BIP concurrent program. (most important)

In addition, BIP engine is built by Java language. If developers know Java, it is a big plus because they can write custom programs to call the APIs of BIP engine. This knowledge would give the organization much more flexibility to construct a customized version of the BIP framework.

Oracle Report & Data Template & Conversion Utility	→	Concurrent Program & Request Group	→	XML & Encoding	→	BIP Template & XSL
XML Publisher Administrator	→	Template & Data Definition	→	Bursting & xdodelivery.cfg	→	Print Server Printer & Pasta.cfg
Deployment FNDLOAD & XDO_Loader	→	BIP Debug	→	JAVA		
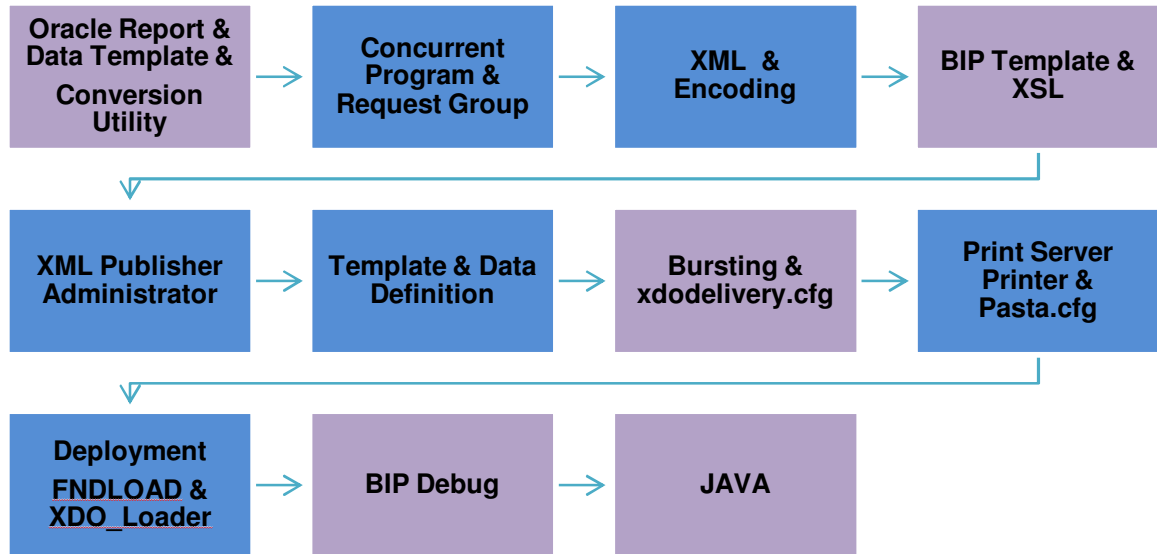
Chart 1 - Required Technical areas (Shuai Wang 2011)

For those companies planning to upgrade Oracle EBS to R12 and eventually to Fusion, they should make a plan to eliminate any in-house built Oracle RDF (Resource Description Framework) programs. As Oracle Report technology will fade away, developers should use BI Data Template when defining a new data source generation program or use the Oracle Report Conversion Utility when updating an existing Oracle Report. However, because the Conversion Utility is not perfect and has limitations, the developer needs to be cautious. There are many features in the Oracle RDF program that cannot be

converted. Sometimes, it is easier to build a BIP from scratch instead of converting.

It is difficult for developers other than the original programmer to maintain an existing template because there are many conditional statements behind the "Text Form." The developers should be trained to use "<!--comments -->" and color the "Text Form Field" when developing a BIP template. In addition, a separate file containing pseudo-codes to mimic the actual logic and conditional statements would simplify template maintenance in the long run.

Because the XSL is not an OOP (Object Oriented Programming) language, it is extremely hard to maintain. Aside from simple "if-else" statements, we always recommend that developers put programming statements in data retrieval programs such as BIP Data Template and PL/SQL packages.

Project Deployment Automation

In Company XYZ, the BIP deployment process was initially handled manually. After completing BIP project development, the developer had to write a step-by-step migration instruction manual. It was an extremely tedious process. Sometimes, developers missed an important step or provided wrong instructions. Other times, database administrators did not follow the instructions correctly.

.
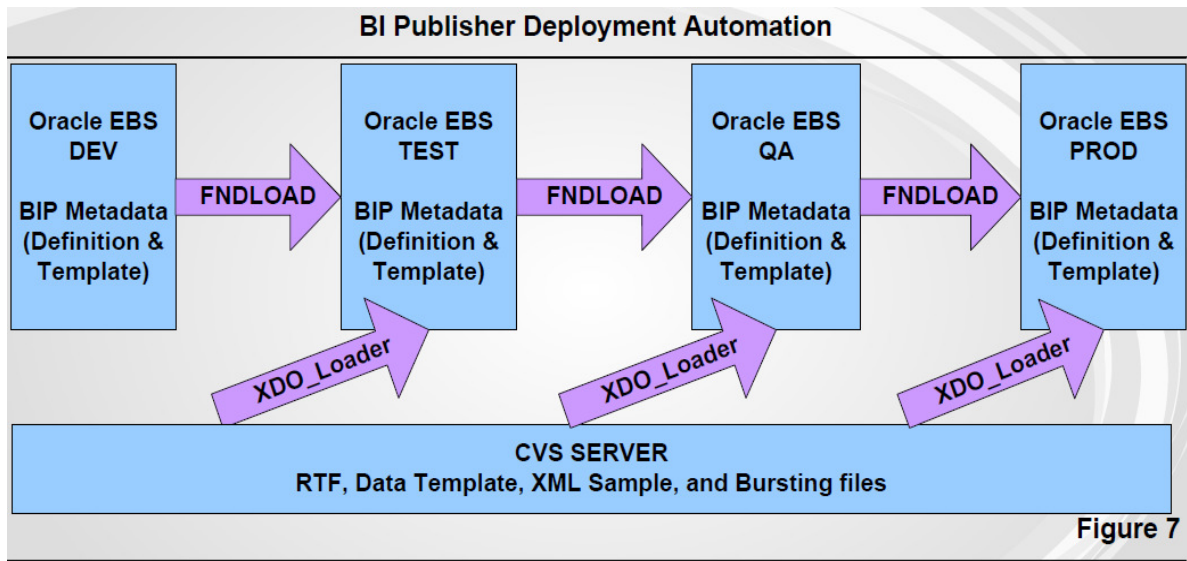
Figure 7 – BI Publisher Deployment Automation (Shuai Wang 2011)

Today, the BIP migration process is fully automated without a single manual step. It uses the FNDLOAD utility to download the metadata (BIP data definition and BIP template) from an Oracle application instance, and it is uploaded to the next application instance. This process then checks out the BIP files (RTF template file, bursting file, and Data Template file) from a CVS server, and uses the XDO_Loader utility to load to the next application instance.

BI Publisher Printing

There are two ways to print a BIP report from Oracle EBS system. Both use the standard Oracle EBS concurrent program as the user interface. The author refers to the first mechanism as "conventional printing" (Figure 8). In this mechanism, users print BIP reports by selecting a pre-installed printer, and the Oracle concurrent program handles the printing process. The author refers to

the second mechanism as "burst printing" (Figure 9). The bursting file provides the printing command and the BIP engine handles the actual printing process.
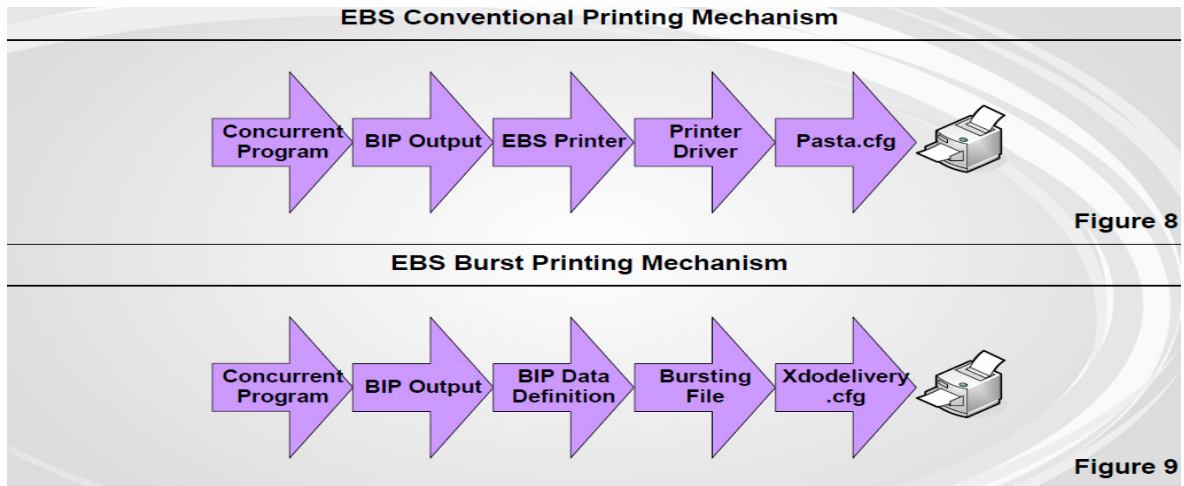
Figure 8 & Figure 9 – EBS Conventional Printing Verses Burst Printing Mechanism (Shuai Wang 2011)

Conventional printing is very easy to setup. The drawback is that the user has to print either all or nothing. Burst printing takes extra work to configure "xdodelivery.cfg" and bursting files. However, it allows the user to print reports selectively. The drawback for "burst printing" is that only the IPP print server is allowed.

If organizations implement the Oracle ERP system worldwide, it will be a challenge for them to accommodate the BIP report printing process with multi-print servers and multi-paper size requirements. For the conventional printing mechanism, the recommendation is to setup different types of printer drivers.

Instead of providing many print configuration files (pasta.cfg), it is easier to setup a UNIX printing script command as the "Arguments" on the "Printer Drivers."
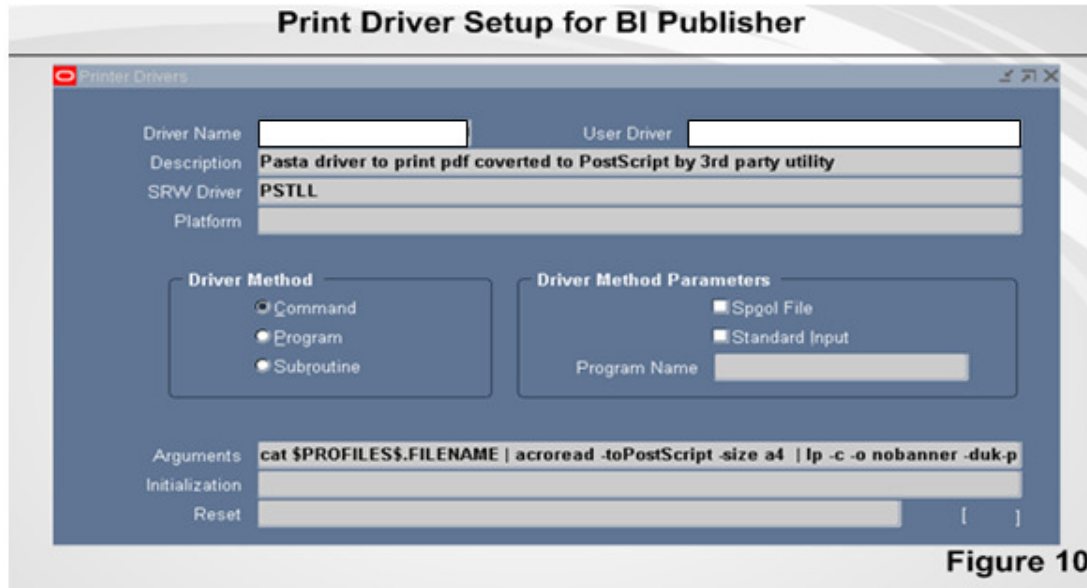
Figure 10 – Print Driver Setup for BI Publisher (Shuai Wang 2011)

CHAPTER 6 - Performance Monitor

Many people have concerns about BIP program performance because it is a relatively new technology. Therefore, Company XYZ built a customized "Oracle Form" inside the Oracle EBS system (Figure 11). This dashboard can be used to monitor, troubleshoot and study the performance of any BIP concurrent program. It also provides the average run time of a BIP concurrent program over a certain period. System administrators can view the log and output file anytime without logging into a specific responsibility. Until today, we have not found any performance issues with BIP concurrent programs.

Figure 11 – BI Publisher Concurrent Program Performance Monitor (Shuai Wang 2011)

CHAPTER 7 - Summary

Company XYZ has benefited from the project because it has reduced the overall cost of development, customization, and ongoing maintenance of ERP reports. It has also saved the company's direct business cost by not spending on third party software maintenance and training of developers.

This ongoing conversion project is considered successful. Without any consulting services, Company XYZ has converted approximately seventy percent of its ERP (Enterprise Resource Planning) reports to BIP from third-party software. The ERP management is pleased with the outcome. Project team members gained valuable knowledge and learned important lessons. Both business and technical challenges were handled very well. More importantly,

business users appreciate the project, and they cannot wait to convert the rest of

ERP reports to BI Publisher technology.

References

Studdard, Leslie. 2008. Business intelligence publisher core components guide. Oracle. September. http://docs.oracle.com/cd/E15743_01/psft/acrobat/txlc.pdf (accessed March, 28, 2011)

Dexter, Tim. 2007. A brief history of BIP. Oracle BI Publisher Blog. Oracle, August 31.

Dexter, Tim. 2007. Barcoding 101. Oracle BI Publisher Blog. Oracle, https://blogs.oracle.com/xmlpublisher/entry/barcoding_101 (accessed May, 25, 2011).

Formscape. 2002. Introduction to Formscape project development curse manual book. Formscape. April 2002

Menzigian, Carol, Frank Rovitto, Ingrid Snedecor and Vanessa Wang. 1999. Oracle reports: Building reports, Release 6.0. Oracle. April. http://docs.oracle.com/pdf/A63455_02.pdf (accessed June, 15, 2011).

Oracle. 2011. Metalink Note: 189708.1 - Oracle Reports 6i Setup Guide for Oracle Applications 11i. (accessed May, 20, 2011)

Oracle. 2011. Metalink Note: 356501.1 - How to Setup Pasta Quickly and Effectively. (accessed September 4, 2011)

Oracle. 2011. Metalink Note: 333504.1 - How to Print Concurrent Requests in PDF Format. (accessed September, 4, 2011)

Oracle. 2011. Metalink Note: 1268217.1 - Output Post Processor (OPP) Log Contains Error "java.lang.OutOfMemoryError: Java heap space" (accessed June 1, 2011)

Passi, Anil and Vladimir Ajvaz. Oracle e-business suite development & extensibility handbook. http://blogs.oracle.com/xmlpublisher/2007/08/a_brief_history_of_bip.htm (accessed September, 11, 2011).

Ryali, Kishore. http://apps2fusion.com/at/64-kr/346-bi-publisher-report-migration-utility (accessed April, 20, 2012).

Tech-FAQ. 2012. ERP (Enterprise Resource Planning). http://www.tech-faq.com/erp.html (accessed April, 4, 2012).

Appendix A

About Company XYZ's ERP (Enterprise Resource Planning) System

- Oracle E-Business Suite 11.510.2
- Oracle Database 10.2.0.3
- Oracle Xml Publisher 5.6.3
- Oracle BI Publisher Template Builder for Word 10.1.3.4.1
- Oracle Report Builder 9i
- JRE 1.6

Appendix B

Report\Country	United States	United Kingdom	Australasian	Iberia	Taiwan	Canada	Italy	French
Remittance Advice Report	X		X		X		X	X
Performa Invoice Report	X		X	X	X	X		X
Invoice Report		X	X	X	X		X	X
Return Label		X	X	X	X	X	X	X
Pallet Report	X	X	X	X	X		X	X
Sales Order Acknowledgment Report	X				X	X	X	
Purchase Order Report	X	X	X	X	X	X		X
MRB Report	X		X	X	X	X	X	X
Kanban Card	X	X		X	X	X		X
Statement Report	X	X	X			X	X	X
Shipping Label	X		X	X	X	X	X	
Pick Slip Report	X	X	X	X	X	X		X
Pack Slip Report	X	X		X	X	X	X	X
Bank Payment EFT (Electronic File Transfer) Report	X	X		X	X		X	X
RMA Return Summary Report	X	X		X	X	X	X	X
Rebate Check	X	X	X	X	X	X	X	X
Check	X		X		X	X		X
Move Order Transaction Report		X	X	X	X	X	X	X
Receipt Traveler Report	X		X	X	X	X		X
Discrete Job Routing Sheet	X	X	X		X	X	X	X
Shot Order Report	X	X	X		X	X	X	
Positive Pay Report		X	X	X	X	X	X	X
Shipping Confirm Invoice	X	X	X	X	X	X		X

Commercial Invoice Report	X		X	X	X	X	X	X
Shipping Discrepancy Report	X	X	X		X	X	X	X
Express Logistics Manifest Report		X	X	X	X	X		
Marketing Program Authorization Report	X	X	X	X		X	X	X
Inter-Company PO Report	X			X	X	X	X	X
Inter-Company Sales Order Report	X	X	X	X	X	X		
Price Protection Report	X	X	X	X	X		X	X
Cycle Count Listing	X		X	X		X	X	X
LPN Label	X	X	X		X	X	X	X

Appendix C

Packing List Report Example

SUPPLIERS INC
20 West 34th Street
New York NY 10118
Phone (212)555-5550

PACKING LIST

SHIP TO
0078742033938
WAL-MART DC 6069A-ASM DIS
1200 MATLOCK DRIVE
ST. JAMES, MO 65559

DATE	STORE NO	PO NO	DEPT NO	ACCOUNT NO
08/22/2003	0078742033938	2550028385	00026	360052260

ROUTING	WEIGHT	#CARTONS	
Roadway	81	7	

UPC Code	Buyer Code	Vendor Code	Description	Qty	Units
098601005500	002658827	00550	AP BAYHILL	2	CA
098601103701	002623404	10370	AP PINEHURST	2	CA
098601121033	002625937	12103	AP LATROBE	6	CA
098601203203	002608734	20320	GOLF SHOE	1	CA
098601203401	002609987	20340	GADGET	1	CA
098601205504	002624334	20550	AP PEBBLE BEACH	1	CA
098601400015	002648021	40001	AP AUGUSTA	1	CA
			TOTAL	**14**	

Purchase Order Report Example

ORACLE *E-Business Suite*

Purchase Order

PURCHASE ORDER NO.	REVISION	PAGE

SHIP TO:

VENDOR:

BILL TO:

CUSTOMER ACCOUNT NO	VENDOR NO.	DATE OF ORDER/BUYER	REVISED DATE/BUYER
PAYMENT TERMS		SHIP VIA	FOB
FREIGHT TERMS		REQUESTOR/DELIVER TO	CONFIRM TO/TELEPHONE

ITEM	PART NUMBER/DESCRIPTION	DELIVERY	QUANTITY	UNIT	UNIT PRICE	EXTENSION	TAX

Total

AUTHORIZED SIGNATURE

Oracle E-Business Suite