Improve interoperability with Oracle XML Publisher and HP Output Server

Executive Summary

Use of XML as a primary data format is spreading rapidly. Combining Oracle® XML Publisher with HP Output Server creates an output management system that can effectively process and reliably deliver XML-formatted output to print, fax, email, web, and other output destinations. More and more organizations are turning to XML as a medium to solve interoperability issues because data presented in XML is self-describing, and thus more easily understood by different systems.

HP Output Server (HPOS) is the core software platform of any HP Output Management solution — the intelligent link that gets critical documents from virtually any application to nearly any destination reliably, efficiently, and cost-effectively.

Setting up HPOS to manage Oracle XML Publisher is an effective way to provide output management for an environment that:

- Includes Oracle and non-Oracle applications
- Supports processes that need continuing delivery of high volumes of output
- Needs a single point for control and administration of the complete output process
- Needs job tracking that encompasses the output process from end to end
- Needs an output management system that can cover all of your output devices
- Needs an output management system that is readily extensible to such capabilities as international printing

This paper describes a real-life implementation of HPOS and Oracle XML Publisher. The solution successfully processes thousands of job submissions per hour of XML-formatted data and non-XML-formatted documents from a combination of Oracle and non-Oracle applications.

Introduction

More and more organizations are turning to XML as a medium to solve interoperability issues. One reason is that data presented in XML is self-describing, and thus more easily understood by different systems.

While XML-formatting eases moving reports and data among applications, it is not formatted for presentation. Commonly, output requires a merge step into a presentation form and then a transformation to an appropriate output-device-specific format such as PCL or PostScript® for printers and G3 TIFF for fax devices.

Oracle has been investing in the development of XML-enabled products for several years and continues to add more to its line. Oracle XML Publisher is a recent addition. The combination of Oracle XML Publisher, part of the Oracle middle-tier E-Business suite, and HP Output server (HPOS) can take XML-formatted data from an application and deliver them to printer, fax, email, web, and other output devices throughout your organization's ecosystem. Oracle XML Publisher prepares the raw XML data for publication by merging the raw data into XML-FO templates, then transforming the finished reports into PDF, HTML, and EDI formats. These documents may require further transformation into other formats, depending on the capabilities of the designated output devices. HP Output Server can do these further transformations as well as distribute the documents to the appropriate output devices with very high reliability.

You can install Oracle XML Publisher inside an Oracle ecosystem, as a stand-alone application bridging between one or more applications and the HP Output Management system, or as an ancillary application directly managed by the HP Output Management system.

A general-purpose configuration for XML-formatted output

Deploying Oracle XML Publisher as a process managed by HPOS provides an architecture with the greatest scope and flexibility. In this configuration Oracle XML Publisher becomes a general-purpose XML-formatting engine for reports and labels generated by any applications that uses HPOS as its output management and delivery engine. There are several advantages. This approach can:

- Accept and manage XML data from both non-Oracle and Oracle applications
- Accept input in XML and in other formats
- Facilitate shifting to an environment that uses XML for inter-application data exchange by allowing you to use your existing connections between applications and HPOS and between HPOS and your output devices and storage systems with only a minimal change to the HPOS-input interface
- Take advantage of HPOS's high-volume spooling capabilities
- Provide HPOS's reliability, job tracking, and flexibility to output from these applications
- Enable centralized management for output-related processes, including transformation of XML data to device-ready output
- Enable straightforward, centralized upgrading to your output capabilities, such as gaining the ability to print virtually any document in any language (including multiple languages in the same document) in any location without altering your output devices in any way by adding the HP International Printing for HP Output Server module

Applications send output requests containing batches of XML-formatted data to HPOS. An engine within the HPOS ecosystem divides the incoming data into sub-jobs depending on the output destination and on the form being filled out. HPOS logs these sub-jobs, places them into its Job Queue, then directs them to Oracle XML Publisher for transformation. XML Publisher merges the XML-formatted data with the correct form (invoice, purchase order, product label, and so forth) and transforms the completed form into PDF. HPOS performs further transformations as necessary to convert these PDF-formatted documents into page description languages appropriate to the output

devices such as PCL, PostScript, or G3 TIFF. HPOS then delivers the finished documents to the output devices.

Example: Handling XML data in a high volume environment of Oracle and non-Oracle applications

An HP customer has an environment containing:

- Oracle E-Business Applications
- Oracle Warehouse Management Systems
- Non-Oracle applications

These applications are configured to output reports and labels formatted in XML. HP Output Server is the output management system for the entire environment, distributing reports and labels to printers, EDI, fax, FTP, email, and a fleet of Zebra Label barcode printers. It is a high-volume environment, processing several thousand jobs per hour.

Figure 1 below shows the solution. The applications send raw XML data for mission-critical label and customer-facing documents to HPOS, where the data go into the Job Queue. XML Publisher provides the forms-merge, creating the finished documents and barcode labels, and the XML-to-PDF reformatting. Forms templates are created in Microsoft® Word and are stored in Rich Text Format (.rtf) in a forms repository.

Documents destined for archiving and email are now in their proper format (PDF), and HPOS delivers them without any additional processing to the appropriate destinations. Documents being sent to output devices undergo further transformation to PostScript or PCL.

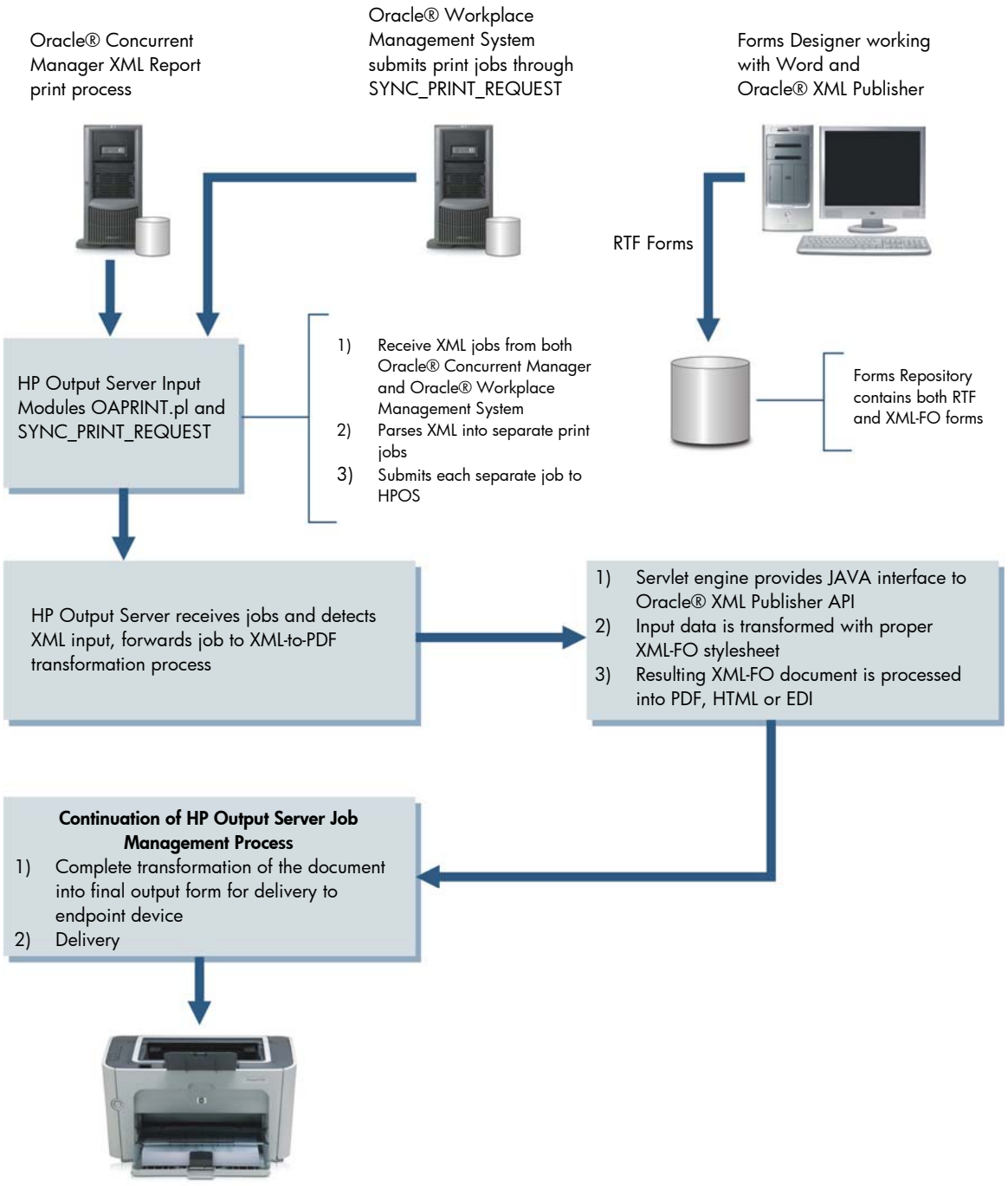
Input processes

Submitting data from Oracle E-Business applications using Concurrent Manager

The Oracle E-Business applications use the Oracle RAC architecture. This implementation has five Concurrent Managers. There is one instance of HPOS, with binaries mounted and shared across the five Concurrent Managers. The HP Output Server interface transfers requests from the Concurrent Managers into HPOS. Each request consists of a batch of raw XML-formatted data. Each batch contains multiple sets of data. Normally these data go into different forms and to different output devices. For example, a batch submission may contain data to fill out invoices for three different customers, an updated Bill of Materials that reflects recent engineering changes, and a list of parts in the next shipment from a supplier in China. These are to be put into the proper forms and routed to fax, print, web, and email respectively.

HP modified its standard interface to parse the XML requests submitted by the Concurrent Managers. While parsing the data, the interface identifies the separation points between sets of data, the destination form or label for each set, and the output destination for the completed form/label. Using this information, the interface divides the XML request into sub-jobs, where each sub-job corresponds to a different combination of form and destination requested. The interface submits these sub-jobs to HPOS one at a time.

Figure 1: Solution Overview



The interface passes information in both directions. It updates the Oracle log with the job-number information generated by HPOS. It also writes delivery notices generated by HPOS reporting the success or failure of each sub-job into the Oracle log.

Submitting data from Oracle Warehouse Management Systems

The Oracle Warehouse Management System (OWMS) does not use Concurrent Manager. OWMS initiates an output request by calling a PL SQL program, SYNC_PRINT_REQUEST. SYNC_PRINT_REQUEST is a stub with no content. HP has written a JAVA program named

“sync_print_request”. The JAVA code wraps the XML data from OWMS in a <transmission_block>data</transmission_block> node and sends it to HPOS via a TCP/IP socket.

The receiving end of the TCP/IP socket on HPOS unwraps the data, splits it into separate sub-jobs, and places the sub-jobs in the Job Queue.

XML data and XML-FO Forms to PDF with Oracle XML Publisher

Adding a JAVA virtual machine

Oracle XML Publisher requires a JAVA virtual machine in order to perform its functions. HP is using Apache-Tomcat as the hosting JAVA virtual machine. The Tomcat server runs an HP-written JAVA servlet that:

1. Merges input XML data with an XSL-FO (eXtensible Style Language-Formatting Objects)
2. Reformats the completed form to PDF.

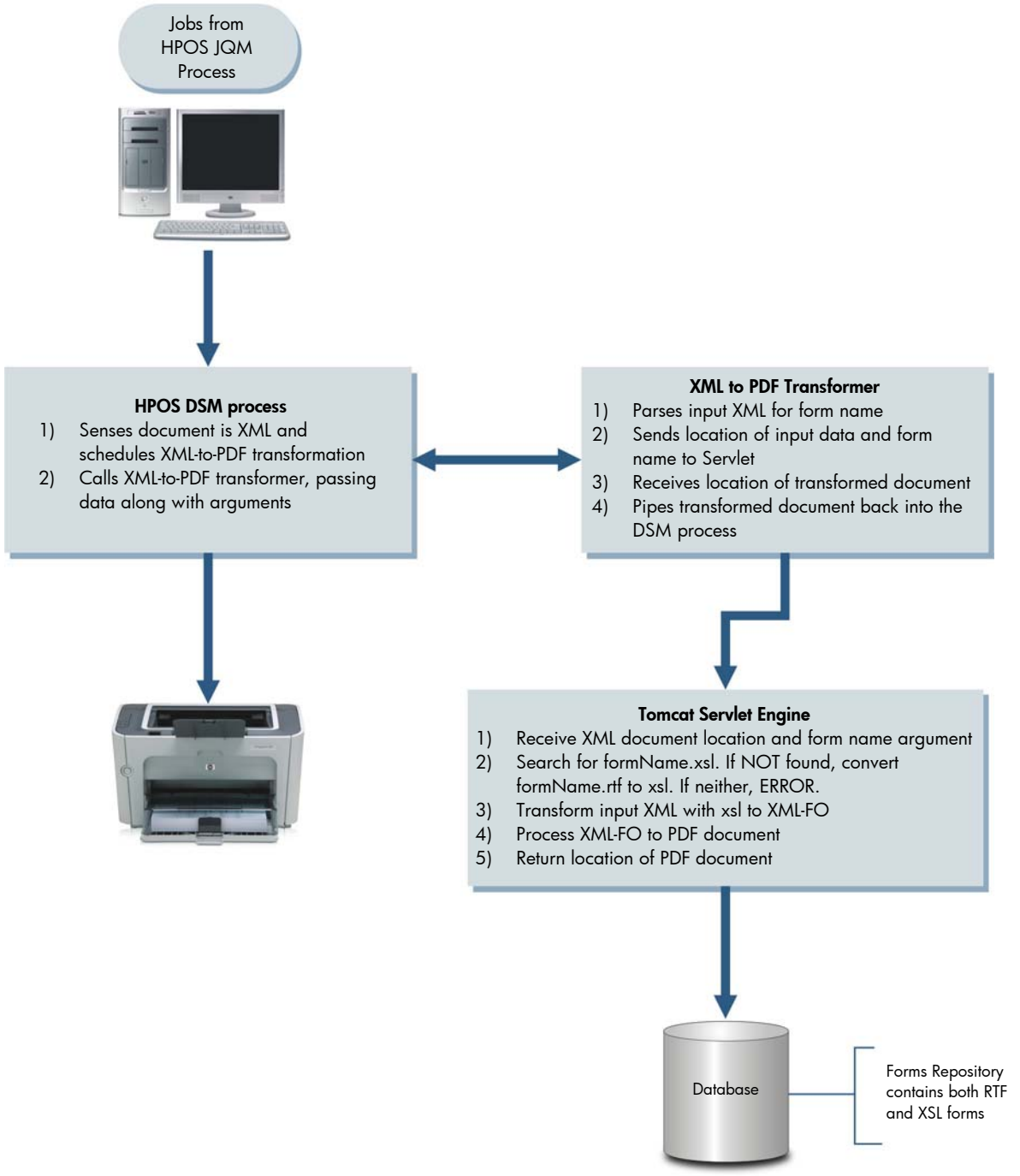
Figure 2 shows the servlet process flow.

The Tomcat server and the servlet reside on the same server as HPOS. Forms designers generate the forms in Microsoft Word and save them as Rich Text Format in a templates directory that is accessible to the servlet engine. Chapter 8 of the Oracle XML Publisher Users Guide provides the API syntax for JAVA applications development.

Linking HPOS and Oracle XML Publisher

HPOS is not a JAVA application. HP can enable communications between HPOS processes and a JAVA application using PERL and the HTTP::Lite module. See Figure 2 below.

Figure 2: The essential features of the XML to PDF transformer as implemented with HPOS.



Both the PERL program and the servlet code are less than two pages each. By keeping the servlet and the HP Output Server on the same physical host we avoid the complications and overhead of SOAP, XML-RPC, and network file transfers. The depicted transformer application processes thousands of submissions an hour faultlessly.

Conclusion

Adding Oracle XML Publisher to HPOS creates a powerful output management system that can fully support the management and delivery of output from an XML-enabled application environment. Linking Oracle XML Publisher to the HPOS Job Queue and letting HPOS manage the overall flow allows Oracle XML Publisher to function as a general-purpose XML-transformation engine within HPOS. This configuration allows HPOS to support an environment of Oracle and non-Oracle applications. It also provides:

- A single location for administering and upgrading the complete output management process
- Job tracking that encompasses the XML-forms merge and XML-to-PDF transformation step as well as the other steps in the output process
- Output capabilities to print, fax, email, web, and more through HPOS
- Reliable output processes

Oracle XML Publisher allows organizations to use Microsoft Word for forms design and an Apache-Tomcat servlet as the forms-merge engine instead of requiring the proprietary products often encountered in this field, with their attendant purchase and maintenance costs.

The example in this paper shows HPOS plus Oracle XML Publisher and the open source components working in a high-volume environment. The system handles bursts of input documents ranging into the thousands and will easily handle a sustained job rate of thousands of jobs an hour.

For more information

For more information about HP Output Management solutions, please send an email to outputmanagement@hp.com or visit: www.hp.com/go/outputmanagement or contact your HP sales representative.

4AA1-3610ENW, 06/2007