

Check Printing using XML Publisher

An Oracle White Paper
June 2005

Check Printing using XML Publisher

Executive Overview	3
Introduction	3
Process Overview	3
Setup the XML Publisher Responsibility.....	4
Responsibility Assignment	4
Add the XML Report Publisher Program to Request Groups.....	5
Register the Report as a Data Definition with XML Publisher.....	5
Create the Check Print Template.....	7
Check Overview.....	7
Basic Layout.....	8
Creating the MICR data placeholders.....	8
Creating the signature image placeholder.....	9
Mapping the MICR font.....	10
Register the template in the Template Manager	11
Set the E-Business Suite report to generate XML output.....	11
Run the format payment process to obtain the XML data file	12
Submitting the XML Report Publisher Program	13
Printing the Check Format Output	14
Conclusion.....	14

Check Printing using XML Publisher

EXECUTIVE OVERVIEW

Generating payment checks from the E-Business Suite has traditionally required either a third party or considerable customization of the delivered check print reports. The check print format is very specific to each banking organization, requiring specific fonts and pixel-perfect layout. The existing E-Business Suite reports generate flat text output and do not support these features. Oracle XML Publisher now offers an integrated, cost-effective method to generate the check output directly from the E-Business Suite.

INTRODUCTION

Check printing from the E-Business Suite has historically required a third-party solution that can either query the necessary data and format the output or accept the existing flat text output generated by the provided check print programs. To then generate the checks, the format components had to be placed with pixel-perfect precision onto the page using the necessary MICR font to format the bank routing number and bank account of the payee. Alternatively customers could take the existing check print program and customize it to format the output correctly. These solutions are costly in terms of their implementation, their ongoing maintenance and their complexity. This all adds to the total cost of ownership for the E-Business Suite.

Oracle XML Publisher offers integrated functionality that allows customers to design their own check format using familiar desktop tools that can then be applied to a check print data source, whether that is the seeded Oracle Report or a custom solution. This format is easily maintainable and can include all the necessary components required in the final output to be printed onto blank check stock.

Taking this approach reduces the complexity of the systems surrounding the check generation and thereby reducing the total cost of ownership of the E-Business Suite.

Process Overview

This solution will work with XML Publisher 4.5 and forward. Later releases offer more functionality in the templates to make the process even simpler. This white paper assumes that the reader is familiar with the Template Manager and the templates themselves.

1. Set up the XML Publisher responsibility.
2. Register the E-Business Suite report as a Data Definition with XML Publisher.
3. Design the check print template.
4. Register the template in the Template Manager.
5. Set the E-Business Suite report to generate XML output.
6. Run the format payment process to obtain the XML data file.
7. Run the XML Report Publisher request to merge the data with the template.

SETUP THE XML PUBLISHER RESPONSIBILITY

XML Publisher provides a responsibility and a concurrent program that require set up in the E-Business Suite.

Responsibility Assignment

The XML Publisher Administrator responsibility allows access to the Template Manager. Assign the responsibility to the appropriate users as follows:

1. From the System Administrator responsibility, navigate to the Users form. Navigation path: Security > User > Define
2. In the Responsibilities region, select XML Publisher Administrator from the Responsibility list of values.

The screenshot shows the Oracle Users form with the following details:

- User Information:**
 - User Name: OPERATIONS
 - Password: [Redacted]
 - Description: Pat Stock
 - E-Mail: pstock@visionops.us.com
- Person Information:**
 - Name: Pat Stock
 - Business Group: Vision Corporation
 - Contact Of: [Redacted]
 - Fax: 555-1212
- Effective Dates:**
 - From: 01-APR-1996
 - To: [Redacted]
- Responsibilities Table:**

Responsibility	Application	Description	Security Group	From	To
Oracle Installed Base Adm	Install Base	Oracle Installed B	Standard	14-OCT-2002	
Oracle Installed Base User	Install Base		Standard	19-FEB-2002	
Application Developer Con	Oracle Common Mo		Standard	28-JUL-1999	01-OCT-2000
XML Publisher Administrat	Oracle XML Publish	Oracle XML Publi	Standard	01-APR-2004	
Order Management Super	Oracle Order Manag		Standard	15-AUG-2000	

Figure 1: Adding the XML Publisher Administrator responsibility to a User

ADD THE XML REPORT PUBLISHER PROGRAM TO REQUEST GROUPS

With the 11i10 CU1 patch the concurrent manager application added tighter integration with XML Publisher, allowing users to select a template as a runtime option when they submitted their concurrent requests. If you have installed this patch on your system then this step can be ignored.

Add the “XML Report Publisher” program to the appropriate request group depending on what concurrent program is to be published and which responsibility is being used. The XML Report Publisher program should be added to the request groups that the base programs reside in. In the case of check printing, the program should be added to at least the Payables All Reports reporting group.

To add the XML Report Publisher program to a request group:

1. From the System Administrator responsibility, navigate to the Request Groups form. Navigation path: Security > Responsibility > Request
2. Query the request group.
3. Add XML Report Publisher as a Program.

The screenshot shows the 'Request Groups' form in Oracle EBS. The form is titled 'Request Groups' and has a blue header bar. Below the header, there are several input fields: 'Group' (All Reports), 'Application' (Payables), 'Code' (empty), and 'Description' (All Payables SRS reports (including update programs)). Below these fields is a section titled 'Requests' which contains a table with three columns: 'Type', 'Name', and 'Application'. The table lists several programs, with 'XML Report Publisher' selected. Below the table is a 'Description' field with the text 'Generate PDF from existing XML output and XML Publisher Template'.

Type	Name	Application
Application	Payables	Payables
Program	XML Report Publisher	XML Publisher
Program	Workflow Background Process	Application Object Library
Program	Mass Additions Create Report	Assets
Program	Publish RX Reports	Assets
Program	Invoice Register	Payables
Program	Invoice on Hold Report	Payables
Program	Cash Requirement Report	Payables
Program	Payment Register	Payables
Program	Missing Document Numbers Report	Payables

Figure 2: Adding XML Report Publisher program

REGISTER THE REPORT AS A DATA DEFINITION WITH XML PUBLISHER

XML Publisher provides a Template Manager to register your report data definitions and templates. Register the report data definition in the Template

Manager to make the data source available to XML Publisher for use with the designed templates. This will depend on the check format program you are currently using. The currently seeded payment formats are:

Check Print Program Name	Concurrent Program Name
Format Payments (Bank Transfer Request)	APXPBFTR
Format Payments (Evergreen)	APXPBFEG
Format Payments (Evergreen, Form Feed)	APXPBFEF
Format Payments (Evergreen, Laser)	APXPBFEL
Format Payments (Standard Oracle)	APXPBFOR
Format Payments (Standard Oracle, Form Feed)	APXPBFOF
Format Payments (U.S. Treasury)	APXPBFUS

To register the check print Data Definitions:

1. From the XML Publisher Administrator responsibility, navigate to the Create Data Definition page. Navigation path: Data Definitions > Create Data Definition.
2. Enter the required fields. The data definition Code must match the concurrent program's short name. At runtime this allows the XML Report Publisher program to provide a list of reports that can be published; that is, those that have been registered as data definitions with XML Publisher and the list of appropriate templates for those reports.

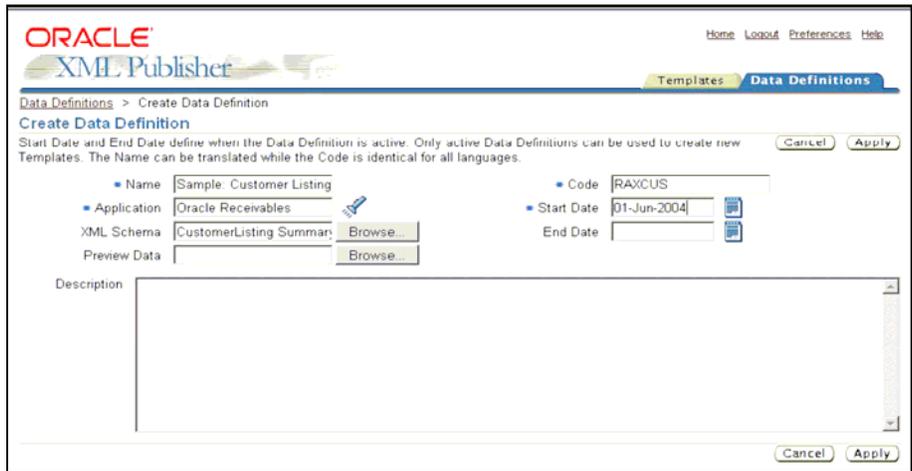


Figure 3: Registering check print program with the Template Manager

For a complete description of the fields required to register your data definition, see the *XML Publisher User's Guide*.

CREATE THE CHECK PRINT TEMPLATE

Check Overview



Figure 4: Check Features

The major features of a check are:

- Check Amount – this value needs to be expressed in words as well as in a numeric format.
- Multiple Address blocks – the payee and payer's addresses need to be displayed.
- Banking Information - this appears at the bottom left of the check, it contains the bank routing number, the payer's account number and the check number. These values must be rendered using a specific MICR font and printed using magnetic ink.
- Signature block – for any company generating more than five to ten checks per day, a signature image is the only real solution. This image needs to be secured and only available during a check print run.
- Layout – banks are very specific about how the check objects should be laid out on the page.

The seeded Oracle Reports provide all of the above data information. All that needs to be done is to format the data correctly with regard to position, fonts and signature images.

You can create the check print template just as you would any other XML Publisher template using Rich Text Format (RTF) or the Portable Document Format (PDF). This paper will cover the RTF approach.

There are several variations on typical check print layouts. For example, the check may appear at the top or the bottom of a list of invoices that the check is paying, or the check may appear above or below a letter or form data. No matter what layout is required, an RTF template can be written to handle the format required.

Basic Layout

The basic layout of the check can be created using table structures in the RTF

ORACLE 500 Oracle Parkway Redwood Shores CA 94065		Date: 01-May-2004
Pay to the order of	Abbot Tech 1 The Street San Francisco CA 94404	\$ 000000.00
Amount in Text		
Anonymous Bank Aaa Street New York, NY 10000		

Figure 5: Basic Check Layout

You can see that images and boilerplate can be embedded directly into the layout using tables with selectively hidden borders to organize the layout. The incoming data can be mapped into the MSWord form fields supported by the XML Publisher parser. To get the layout to match the check stock paper layout may require some adjustment to the RTF template. A combination of adjusting table/row/cell properties and the use of carriage returns can be used to achieve this.

Creating the MICR data placeholders

XML Publisher has the ability to use external fonts at runtime; as long as the font is accessible at runtime on the server. XML Publisher does not currently license a MICR font but a quick search on the web will find both freeware and commercial fonts that can be used.

Once you have the MICR font you will need to make it available to MS Word. Do this by installing it in the <<windows root>>/fonts directory. The font can now be accessed from Word. You can check this by looking for it in the font drop box.

To insert the MICR strings into your template, insert the appropriate number of form fields and map them to the XML data that will come in at runtime. Highlight these fields and set the font to the MICR font installed earlier.

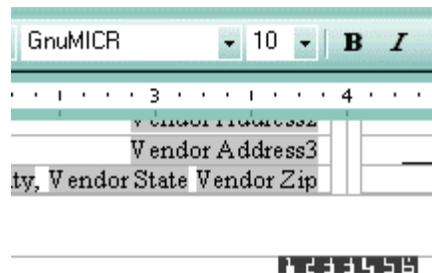


Figure 6: MICR information fields

Once this is completed you have completed creating the MICR placeholders in the template.

Creating the signature image placeholder

If you are likely to be printing more than a few checks per day it does not make sense to have the checks signed personally. XML Publisher provides the means to insert a dynamic signature image on the output.

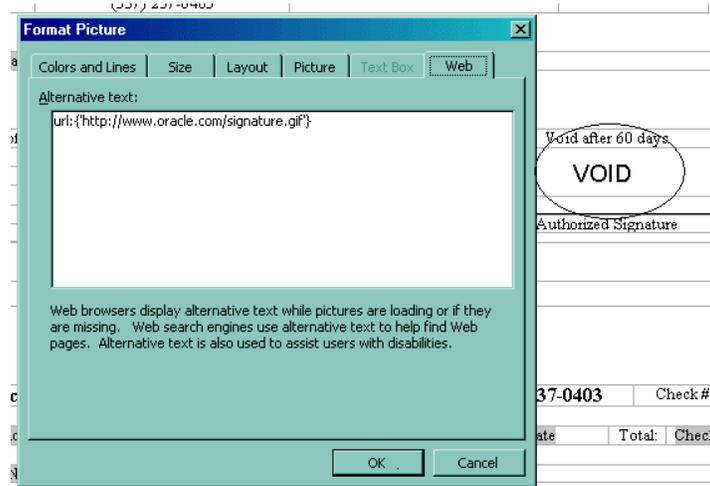


Figure 7: Signature image substitution

The signature image can be secured quite simply by creating a dummy image in the template. You can then use the image substitution features of XML Publisher to have the image rendered at the time of document generation.

Within the web tab of the dummy image enter a URL that points to the location of the image. In this example we use “url:{"http://www.oracle.com/signature.gif"}”. You may not want to hard code the location in the template. In this case you could extract the location into an XML element and then reference that, e.g. “url:{"SIG_LOC"}”. XML Publisher will resolve the value at runtime. This could be in a secured directory on a server, or you could map a floppy drive containing the image, to your web server. At runtime, insert the floppy into the drive and the image is accessible and imported into the output document. If the image is not present, then the check will contain the dummy image only and not the valid signatures.

If you require multiple signatures on your checks the same method applies to all the images. You can use conditional formatting around the images if different people have to sign different checks based on some attribute. This can all be supported in the template.

Your template is now complete. Now load it to the Template Manager so that XML Publisher can access it at runtime.

Mounting a floppy drive on an Apache web server can be achieved using the Alias directive.

MAPPING THE MICR FONT

XML Publisher needs to know the location of the MICR font at runtime so that it can use it in the final output. XML Publisher uses a configuration file to set document properties such as security and font mappings. You will need to create this configuration file if you have not done so already.

For the check printing solution XML Publisher only needs the MICR font location; however, if you wish to secure the output, other properties can be set on the document. The configuration file takes the following format:

```
<config version="1.0.0" xmlns="http://xmlns.oracle.com/oxp/config/">
  <properties>
    <property name="system-temp-dir">/tmp</property>
    <property name="pdf-compression">true</property>
    <property name="pdf-security">true</property>
    <property name="pdf-open-password">user</property>
    <property name="pdf-permissions-password">owner</property>
    <property name="pdf-no-printing">true</property>
    <property name="pdf-no-changing-the-document">true</property>
  </properties>
  <font>
    <font family="MICR" style="normal" weight="normal">
      <truetype path="/usr/fonts/MICR.ttf"/>
    </font>
  </font>
</config>
```

You can see the font entry for the MICR font is pointing to the /usr/fonts directory. This directory must be accessible by XML Publisher at runtime. Now place the configuration file in an accessible runtime directory and make a note of the directory location. For further information on the location of the configuration file please check the *XML Publisher User's Guide*.

If you have XML Publisher 5.0 installed you have an alternative to the configuration file. All of these properties can be specified in the RTF template document as custom properties. Simply enter the properties, prefixing them with 'xdo-', and their values into the property dialog in MSWord.

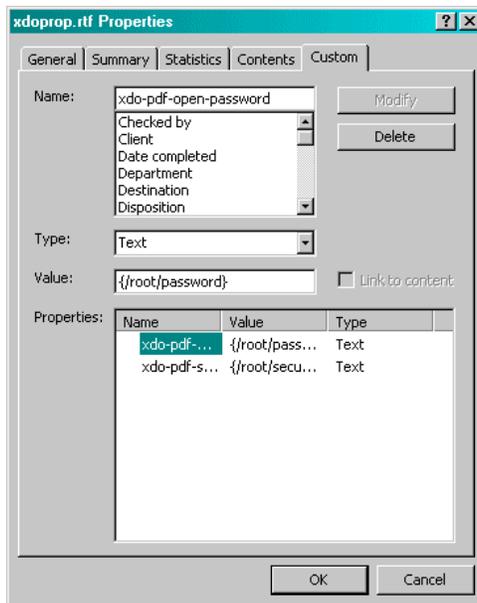


Figure 8: Entering document properties

This feature is very powerful the template now becomes a self-contained document that XML Publisher can understand at runtime. For instance, if you wanted to add the font location to the template properties you would enter

```
xdo-font family="MICR" style="normal" weight="normal"
```

as the property and

```
truetype path="/usr/fonts/MICR.ttf"
```

as the value.

When the template is processed, XML Publisher will read this information and be able to locate the font. For more information on this feature please consult the *XML Publisher User's Guide*.

REGISTER THE TEMPLATE IN THE TEMPLATE MANAGER

Now you are ready to register the template in the XML Publisher Template Manager. From the XML Publisher Administrator responsibility, navigate to the Create Template page. Navigation path: Templates > Create Template.

When you create your template, you register it against an existing Data Definition. In this case you need to register it against one of the check print programs you registered earlier.

Upload the RTF or PDF template file and select the language and territory.

SET THE E-BUSINESS SUITE REPORT TO GENERATE XML OUTPUT

You now need to set up the check print program to generate XML data rather than TEXT as the output when the concurrent manager executes the report.

1. From the System Administrator responsibility, navigate to the Concurrent Programs form. Navigation Path: Concurrent > Program > Define.
2. Query the check print program report you wish to use. The report must be able to generate XML; Oracle Reports 6i supports XML output and this is the format of the check print programs.
3. In the Output region, select XML from the Format drop down list.

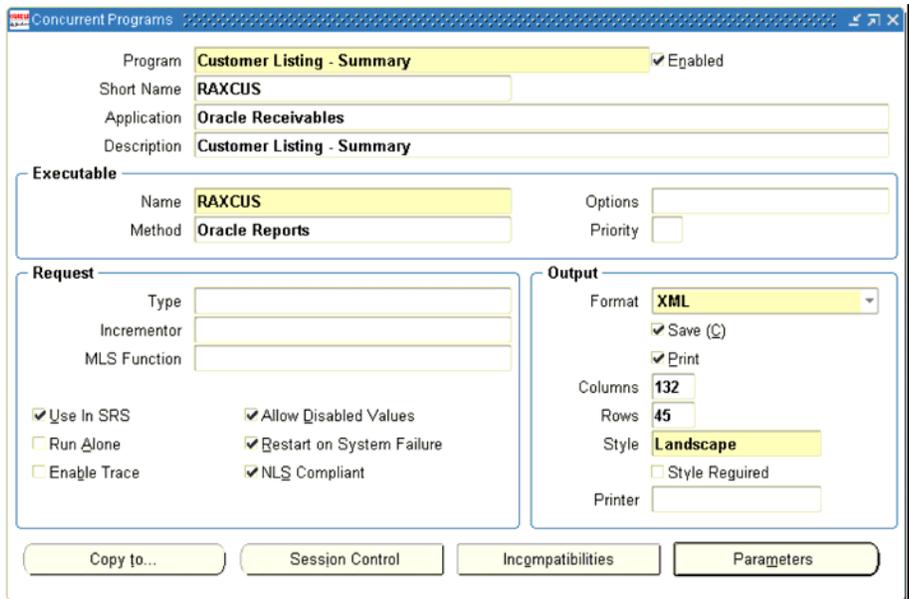
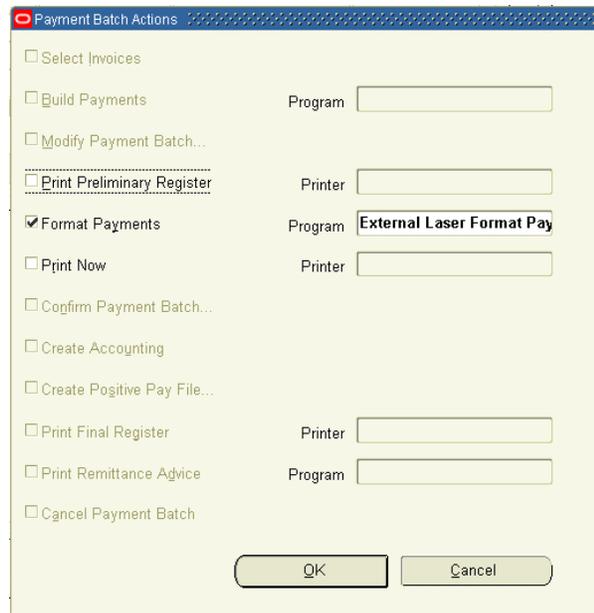


Figure 9: Updating program definition to generate XML

The program will now generate XML data when executed by the concurrent manager.

RUN THE FORMAT PAYMENT PROCESS TO OBTAIN THE XML DATA FILE

Now run the payment format program via the Oracle Payables interface.



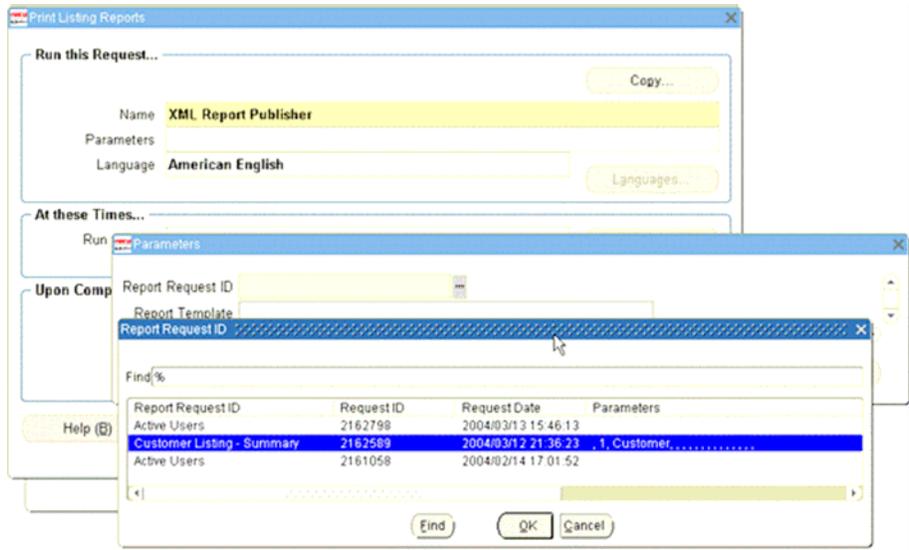
At the time of writing, Oracle Payables have not integrated the XML Publisher functionality into their payment formatting flow. The Payment form programmatically submits the formatting program to the concurrent manager. Because the XML Publisher template is not set during this process, the output will be XML. This will be addressed in a later release.

To generate your checks in PDF output, submit a second request, the “XML Report Publisher” program.

Submitting the XML Report Publisher Program

Use the Submit Request form to run the XML Report Publisher program. You will be prompted for the following parameters:

1. Report Request – the LOV will contain a list of requests that:
 - a. Have produced XML.
 - b. Have been run by the same user or responsibility
2. Report Template – the LOV will contain a list of available templates that are registered with this data definition in the Template Manager.
3. Template Locale – the LOV will contain the language/territory combinations for which templates have been registered.
4. Output Format (appears in release 5.0 and later) - select PDF.



When you execute the program, remember to ensure the signature image is available.

PRINTING THE CHECK FORMAT OUTPUT

You can now print this output using the standard concurrent manager printing functionality or open the resulting PDF and print locally to the check printer.

CONCLUSION

The solution offered by XML Publisher offers an integrated E-Business Suite process that allows customers to design their own check formats using familiar desktop tools that can then be applied to check print data sources to generate check output directly from Oracle E-Business Suite.

Taking this approach will reduce the complexity of the systems surrounding check generation and will help to reduce the total cost of ownership for the E-Business Suite.



White Paper Title

[Month] 2000

Author:

Contributing Authors:

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:

Phone: +1.650.506.7000

Fax: +1.650.506.7200

www.oracle.com

Oracle Corporation provides the software
that powers the internet.

Oracle is a registered trademark of Oracle Corporation. Various
product and service names referenced herein may be trademarks
of Oracle Corporation. All other product and service names
mentioned may be trademarks of their respective owners.

Copyright © 2000 Oracle Corporation
All rights reserved.