The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decision. The development, release, and timing of any features or functionality described for Oracle’s products remains at the sole discretion of Oracle.
Oracle XML Publisher

Integration with PeopleTools
XML Publisher Concept

Separate data / layout / UI translation

- Flexibility
- Reduced maintenance
XML Publisher Development

- Business consultants use familiar desktop tools for layout customization
- Support for Multiple Data Sources

Report Templates

Multiple Data Sources

✓ Rapid Deployment
Integrated Solution

Tools XML Publisher Modules

- Desktop App & Plug-in
- Desktop App & Plug-in
- Template Manager
- Template
- XMLP Engine
- Process Scheduler

Input:
- PSQuery
- PeopleCode
- XML Doc
- Web Services

Output:
- PDF
- Word
- XSL
- Email
- Printer
- Repository
- e-Commerce
PeopleTools Integrated Solution

- Embedded XMLP Formatting Engine inside PeopleTools 8.48
- PeopleCode API for the Engine
- Data Source Manager
- Template Manager
- Report Manager
- PeopleCode API for Template Manager and Report Manager
- PIA pages for managing data sources, templates and reports
**Business Process Flow**

**Design: Data Source**
- **Mock up Report**
  - Determine data requirements for report
- **Identify or create Data Source**
  - PS Query for business user
  - Any XML source for report developer
- **Register Data Source with XML Publisher**
  - Make Data Source available for XML Publishing

**Design: Template & Report**
- **Download Schema & Sample Data**
  - To facilitate template design
- **Create Report Template**
  - Word RTF or Adobe PDF
- **Preview and tweak until complete**
  - Associate Data Source & Template
- **Register XMLP Report Definition**
  - Optional: Create Translation

**Runtime: Request & View**
- **Publish XMLP-based Report**
  - Calls XML Publisher engines
- **View posting in Report Manager**
  - Offline
  - Existing
  - New
Simple online flow

1. Register Data Source
2. Define Report
3. Add Template
4. Add Bursting (advanced)
5. Add Translation (advanced)
6. Publish
Make a data source available for XMLLP

Register Data Source

Source Type: PS Query
Object Owner ID: Receivables

Data Source ID: OOW_CUST_BAL_LTR_PUB
Active

Description: Customer Balance Letter pubqry

Registered Date/Time: 09/13/05 9:49:22AM
Registered By: VP1

Last Update Date/Time: 09/13/05 9:49:22AM
Updated By: VP1

Related Files

<table>
<thead>
<tr>
<th>File Type</th>
<th>File</th>
<th>Last Update Date/Time</th>
<th>Generate File</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample Data File</td>
<td>OOW_CUST_BAL_LTR_PUB.xml</td>
<td>09/14/05 10:49:09AM</td>
<td>Generate</td>
</tr>
<tr>
<td>Schema File</td>
<td>OOW_CUST_BAL_LTR_PUB.xsd</td>
<td>09/14/05 10:49:09AM</td>
<td>Generate</td>
</tr>
</tbody>
</table>
Offline Template Design

- Create in either Word or Adobe Professional
- **Template construction**
  - Create a report or document in Word; save as RTF
  - Download a PDF-based form from government agency
- Download the Schema and Sample Data files from the Data Source that will supply the application specific values
- Using these XML files in conjunction with Word or Adobe Standard, insert and/or map form fields in the template to application data tags
In Adobe, map tags into PDF form template
In Word, insert schema tags into RTF template form fields and preview design.
Define XML Publisher Report

Definition

Report Name: OOW_1099S
Data Source

Data Source Type: PS Query
Data Source ID: OOW_1099S_PUBLIC
Data Source Description: Vendor 1099
Report Properties

Report Description: Vendor 1099s
*Report Status: In Progress
*Report Category ID: QEDMO
Template Type: PDF Form
Archive Report Files: After 44 Days
Registered Date/Time: Registered By:
Updated Date/Time: Updated By:
Download: Data Schema Sample Data
Add template file to Report Definition

- Register Data Source
- Define Report
- Add Template
- Add Bursting (advanced)
- Add Translation (advanced)
- Publish

**Template**

- Report Name: CUSTOMER_BALANCE_LETTER
- Template ID: PRINT_ENGLISH
- Locale Code: en
- Description: Printer Version
- Channel: Printer

**Template Files**

- Effective Date: 06/01/2005
- Version: 1
- Status: Active
- Template File: Balance_Letter_Template_RTF.rtf
Output choices

Report Name: CUSTOMER_BALANCE_LETTER
General
- Report output may be edited
Output Format Options

<table>
<thead>
<tr>
<th>Format Type</th>
<th>Enabled</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTML</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>PDF</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>RTF</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>XLS</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Output Location
- Location: Any

Email
- Allow recipient address entry at report runtime

Bursting
- Generate separate report document files
- Burst Field Name: 

Register Data Source -> Define Report -> Add Template -> Add Bursting (advanced) -> Add Translation (advanced) -> Publish
Viewer security

Register Data Source → Define Report → Add Template → Add Bursting (advanced) → Add Translation (advanced) → Publish

Report Name: CUSTOMER_BALANCE_LETTER

Report Viewers

- Allow viewer ID assignment at report runtime

Web Report Repository/Email Viewers

<table>
<thead>
<tr>
<th>ID Type</th>
<th>ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>VP1</td>
<td>Atamian,Wes</td>
</tr>
</tbody>
</table>

ORACLE
THE INFORMATION COMPANY
Generating the report

• Developer creates process and Run Control page to connect data extraction process with XMLP report generation process.

• For Query data sources, Business Power User utilizes the delivered Schedule Query Report page or the online Query Report Viewer.
Process Scheduler publishes report to Report Manager

Schedule Query Report

Run Control ID: Letter

*Report Name: CUSTOMER_BALANCE_LETTER

Description: Customer Balance Letter

Template Type: RTF

Process Scheduler Request

User ID: VP1

Server Name: Run Date: 09/14/2005
Recurrence: Run Time: 11:56:20AM

Time Zone: Reset to Current Date/Time

Process List

Select Description Process Name Process Type *Type *Format Distribution

☑ Fusion Publisher Query Report PSXPQRYRPT Fusion Publisher Web PDF Distribution
Dear Sirs/Madam,

According to our records as of 01-JAN-04, we show the following open balances.

Your CAD balance is $1,500.00 made up as follows:

<table>
<thead>
<tr>
<th>Invoice Number</th>
<th>Date</th>
<th>Trans Amount</th>
<th>Trans Amount Remaining</th>
</tr>
</thead>
<tbody>
<tr>
<td>602444</td>
<td>06-DEC-00</td>
<td>10,126.00</td>
<td>10,126.00</td>
</tr>
<tr>
<td>602445</td>
<td>06-DEC-00</td>
<td>12,376.00</td>
<td>12,376.00</td>
</tr>
<tr>
<td>10016003</td>
<td>16-NOV-03</td>
<td>132,733.84</td>
<td>132,733.84</td>
</tr>
<tr>
<td>10020178</td>
<td>20-NOV-03</td>
<td>71,077.42</td>
<td>71,077.42</td>
</tr>
<tr>
<td>10020181</td>
<td>21-NOV-03</td>
<td>86,344.81</td>
<td>86,344.81</td>
</tr>
<tr>
<td>502394</td>
<td>22-NOV-03</td>
<td>11,250.00</td>
<td>11,250.00</td>
</tr>
<tr>
<td>10020280</td>
<td>24-NOV-03</td>
<td>128,654.96</td>
<td>128,654.96</td>
</tr>
<tr>
<td>10020310</td>
<td>25-NOV-03</td>
<td>120,853.20</td>
<td>120,853.20</td>
</tr>
<tr>
<td>10020319</td>
<td>26-NOV-03</td>
<td>147,328.21</td>
<td>147,328.21</td>
</tr>
<tr>
<td>294</td>
<td>02-DEC-03</td>
<td>53.35</td>
<td>53.35</td>
</tr>
<tr>
<td>10020402</td>
<td>04-DEC-03</td>
<td>146,775.07</td>
<td>146,775.07</td>
</tr>
<tr>
<td>10020403</td>
<td>04-DEC-03</td>
<td>172,482.45</td>
<td>172,482.45</td>
</tr>
<tr>
<td>10020404</td>
<td>04-DEC-03</td>
<td>147,740.26</td>
<td>147,740.26</td>
</tr>
<tr>
<td>10020405</td>
<td>04-DEC-03</td>
<td>71,577.42</td>
<td>71,577.42</td>
</tr>
<tr>
<td>10020406</td>
<td>04-DEC-03</td>
<td>86,344.81</td>
<td>86,344.81</td>
</tr>
</tbody>
</table>

Private and Confidential
Advanced Options

Bursting
Content Library
Translation
Bursting – separate report files

Use for security & for distribution

```xml
<PayChecks>
  <PayCheck>
    <EmpId>1000</EmpId>
    <DeptId>01</DeptId>
    <Salary>1000</Salary>
  </PayCheck>
  <PayCheck>
    <EmpId>1001</EmpId>
    <DeptId>01</DeptId>
    <Salary>1000</Salary>
  </PayCheck>
  <PayCheck>
    <EmpId>1002</EmpId>
    <DeptId>02</DeptId>
    <Salary>1000</Salary>
  </PayCheck>
</PayChecks>
```

Bursting by EmpId

- EmpId: 1000
- EmpId: 1001
- EmpId: 1002

Bursting by DeptId

- DeptId: 01
- DeptId: 02
Bursting provides multiple features:

- **Data attribute to burst by**
- **Template or translation** for each burst file assigned at runtime per defined data value
- **Report file security** assigned at runtime per application’s existing row level security data
- **Search key data values** stored at runtime for more intelligent search

**Optional**

Report Name: MENU_DESCR

**Bursting**

Burst field Name: PORTAL_PRNTOBJNA

**Template**

Template Control Field: PORTAL_PRNTOBJNA

**Template Application Criteria**

- **Field Value**
  - PASSWORD_CONFIGURATION
  - PT_XMLPUBLISHER

**Template ID**

- MENU_DESCR_1

**Language**

- Spanish
- French

**Burst security**

Security Join Table: PSOPRALIAS

**Security Field Name**: CPRID

**Security ID Type**: User ID

**Field Mapping**

<table>
<thead>
<tr>
<th>Security Join Table Field Name</th>
<th>Data Source Field Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPRID</td>
<td>CPRID</td>
</tr>
</tbody>
</table>

**Search Keys**

Search Field Name

- PORTAL_LABEL
- PORTAL_NAME
Content Library

- Content library is a sub-template within a primary template: either text or image, can be used as page header, footer, legal clause or company logo.

- Syntax in primary template:
  - `<?import:psxmlp://SUBTEMPLATE?>`
  - `<?call-template:"header"?>`

- Content in a sub-template is translatable.
Translations

- Label is extracted using the engine function and saved in an XLIFF
- Download XLIFF file
- Have translation service translate XLIFF
Example XLIFF file for translation

- Note source and target locale and translation units

```xml
<?xml version="1.0" encoding="utf-8" ?>
<xliff version="1.0">
  <file source-language="en-US" target-language="fr-FR" datatype="XDO" original="orphen.rtf" product-version="orphen.xlf" product-name="">
    <header />
    <body>
      <trans-unit id="4d3eb24" maxbytes="4000" maxwidth="15" size-unit="char"
                 translate="yes">
        <source>Total</source>
        <target>Total</target>
        <note><text>Text located: body/table</text></note>
      </trans-unit>
      <trans-unit id="a2bad3c3" maxbytes="4000" maxwidth="22" size-unit="char"
                 translate="yes">
        <source>Seq Name/</source>
        <target>Seq Name/</target>
        <note><text>Text located: body/table/table header</text></note>
      </trans-unit>
      <trans-unit id="59c932c3" maxbytes="4000" maxwidth="57" size-unit="char"
                 translate="yes">
```
XLIFF files related to Report Template

**Template Translations**

<table>
<thead>
<tr>
<th>Source Type</th>
<th>PS Query</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Source ID</td>
<td>MESSAGES_FOR_MSGSET</td>
</tr>
<tr>
<td>Report Name</td>
<td>MAIN_SUB</td>
</tr>
<tr>
<td>Template ID</td>
<td>MAIN_SUB_1</td>
</tr>
<tr>
<td>Description</td>
<td>Colored w/SubTemplate</td>
</tr>
<tr>
<td>Language Code</td>
<td>English</td>
</tr>
</tbody>
</table>

**Scroll Area**

<table>
<thead>
<tr>
<th>Template File</th>
<th>main_with_sub_noQuotes_repeat[1].rtf</th>
</tr>
</thead>
<tbody>
<tr>
<td>Effective Date</td>
<td>11/10/2005</td>
</tr>
</tbody>
</table>

**Generate New Translation File**

**Translation Files**

<table>
<thead>
<tr>
<th>Active</th>
<th>Translation File</th>
<th>Language</th>
<th>Preview</th>
<th>Browse</th>
<th>Delete</th>
</tr>
</thead>
<tbody>
<tr>
<td>☑</td>
<td>main_with_sub_repeat[1110_espl].xliff</td>
<td>Spanish</td>
<td>Preview</td>
<td>Browse</td>
<td>Delete</td>
</tr>
<tr>
<td>☑</td>
<td>main_with_sub_noQuotes_repeat[1110_2].xliff</td>
<td>French</td>
<td>Preview</td>
<td>Browse</td>
<td>Delete</td>
</tr>
</tbody>
</table>
Report Manager – enhanced search

- Search for report by the report’s defined search keys to locate one specific bursted report file
- Reports secured for a user or role
Real life goal is reducing IT dependency

- For ad hoc reports and documents
  - Business users can use Query as data source and design their own ad hoc reports and documents

- For production reports and forms
  - Developers supply XML data extraction program from any source and the initial template designs
  - Business users can tweak the template layouts as needed
  - New form versions only require user to map data elements into the new template form; no coding required
Help for report developers

- Registration, storage and management of report definitions and related files
- Schema generator utility for xsd files
- API’s, application classes, and PeopleCode examples to facilitate incorporation of XML Publisher reporting function onto application pages and run controls
- Managed Object support to facilitate movement of new reports from development to production