

Oracle E-Business Suite and BI Publisher using PLSQL

James W. Hanon
O2Works LLC

The trend to using Oracle's BI Publisher as the means to format and produce complex, dynamic and rich business documents is growing all of the time. The Oracle E-Business Suite has been tooled up to send reports output to the BI Publisher for formatting, and its use is expanding.

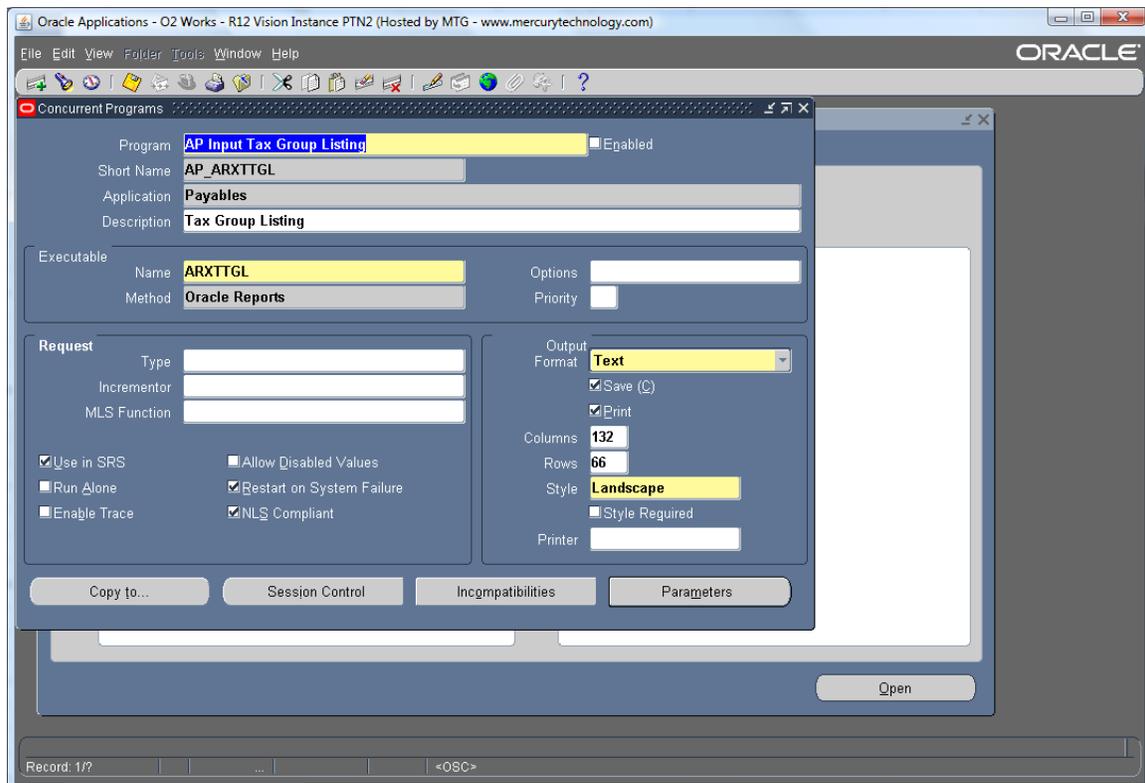
For years, the standard reports development tool for the Oracle E-Business Suite has been Oracle Reports. Oracle Reports is a tried-and-true means of producing everything from daily reports, to invoices, to purchase orders. However, the generally accepted output format has been straight text, with neither graphics nor control over printed fonts.

The Oracle BI Publisher has changed all of that.

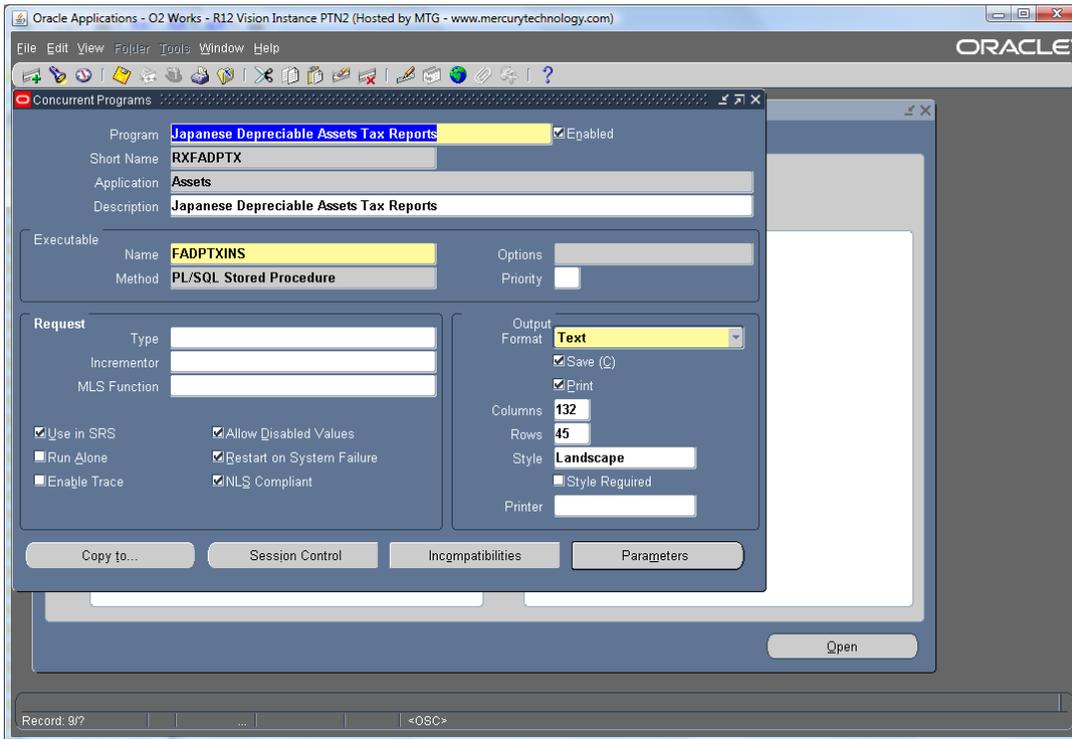
However, the standard means of generating the XML data used in a BI Publisher report is still Oracle Reports.

Producing XML output from Oracle Reports is a fairly simple matter, especially for simple datasets. For complex XML output, PL/SQL may be a better answer. However, defining a PL/SQL concurrent program for a BI Publisher report is not so straight forward.

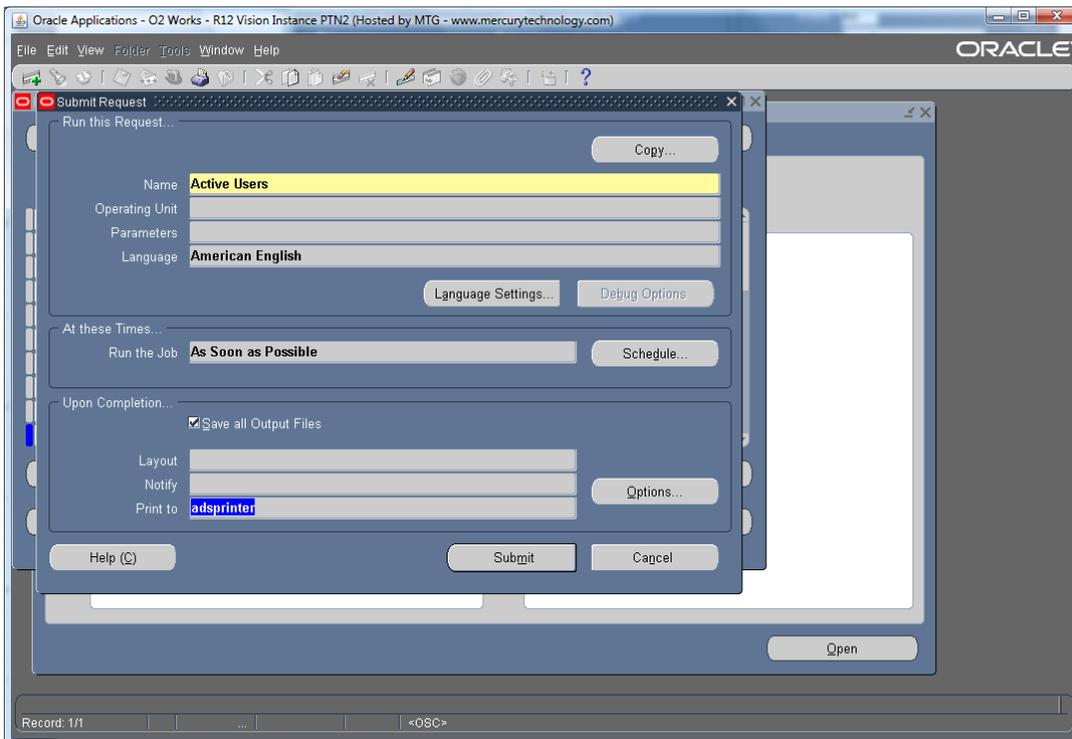
In a standard Oracle Reports concurrent program, the output format is Text, and the number of rows and columns defined per the report.

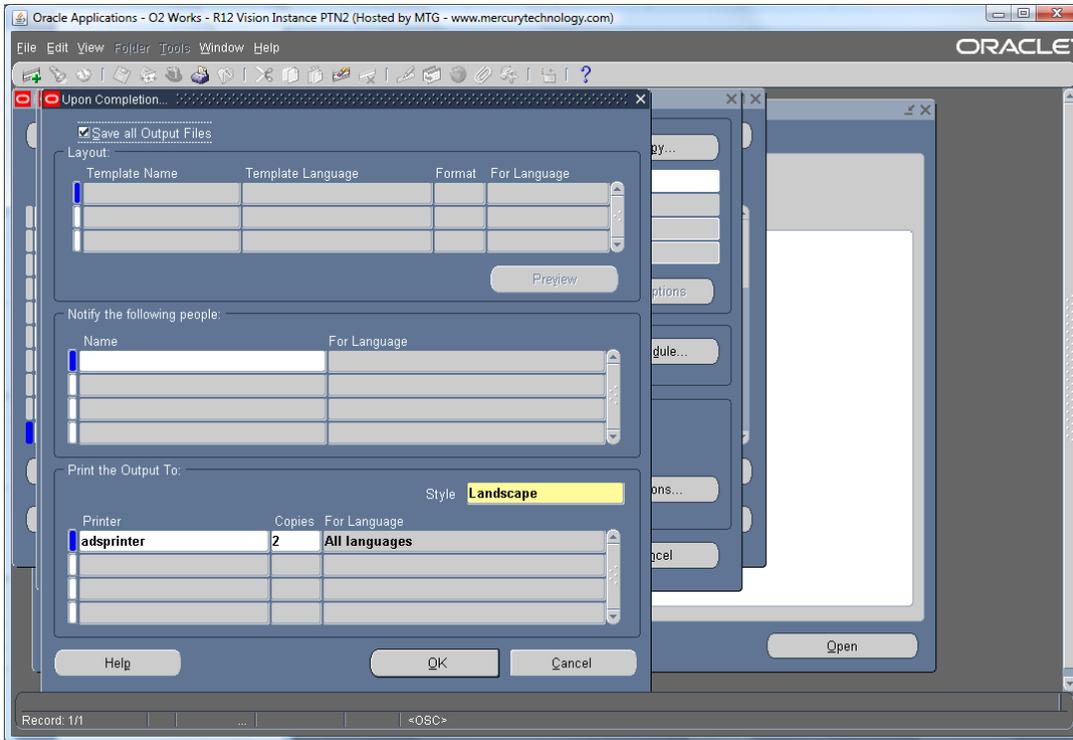


In a standard Oracle PL/SQL concurrent program, too, the output format is Text, and the number of rows and columns defined per the report.

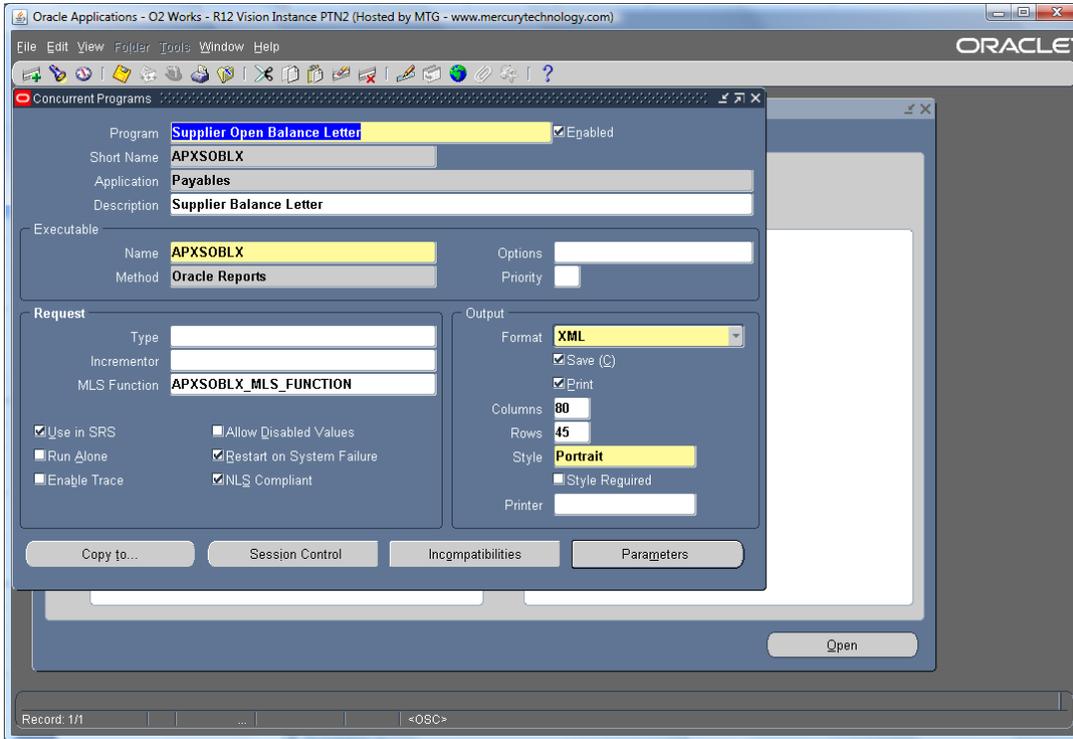


In both cases, the user selects the concurrent program to be run, sets the number of copies to be printed and the destination printer, and then runs the program.

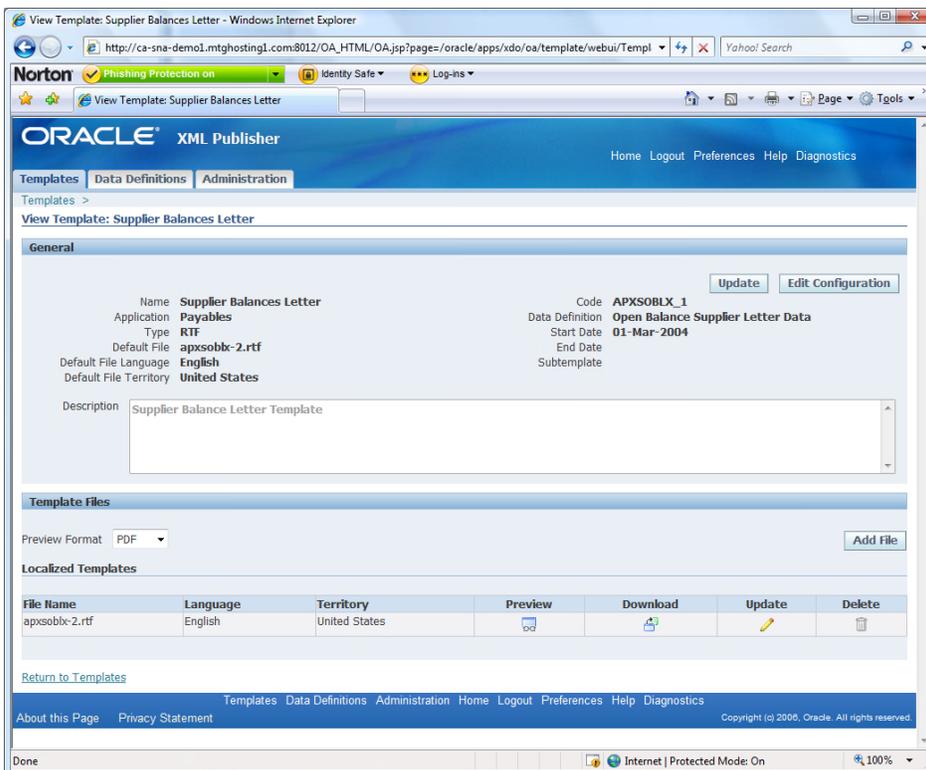
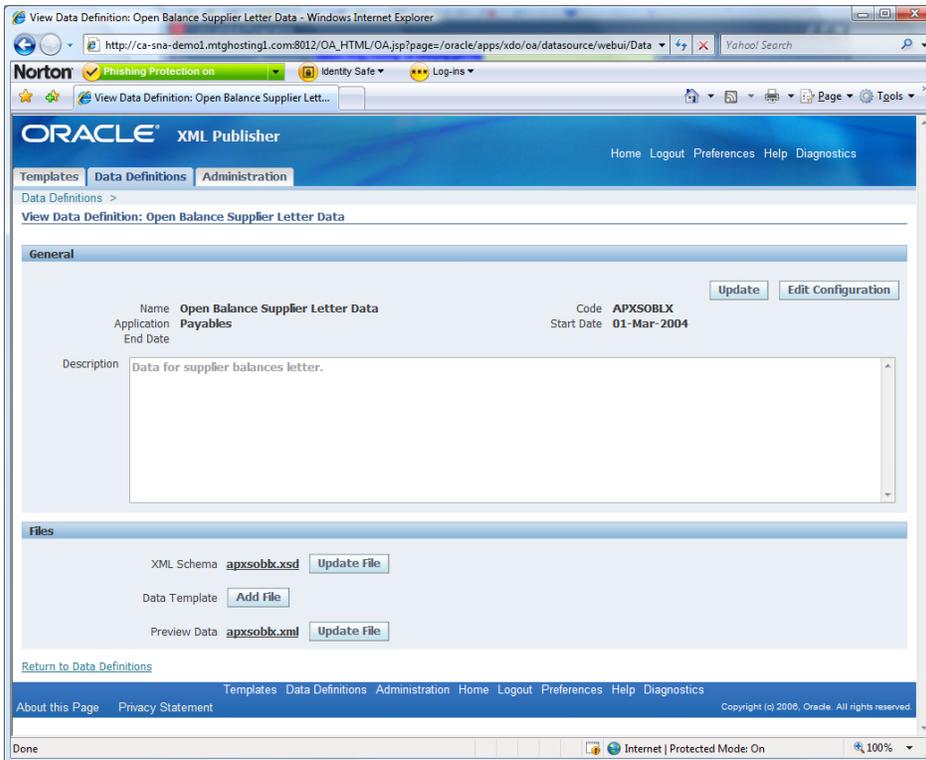




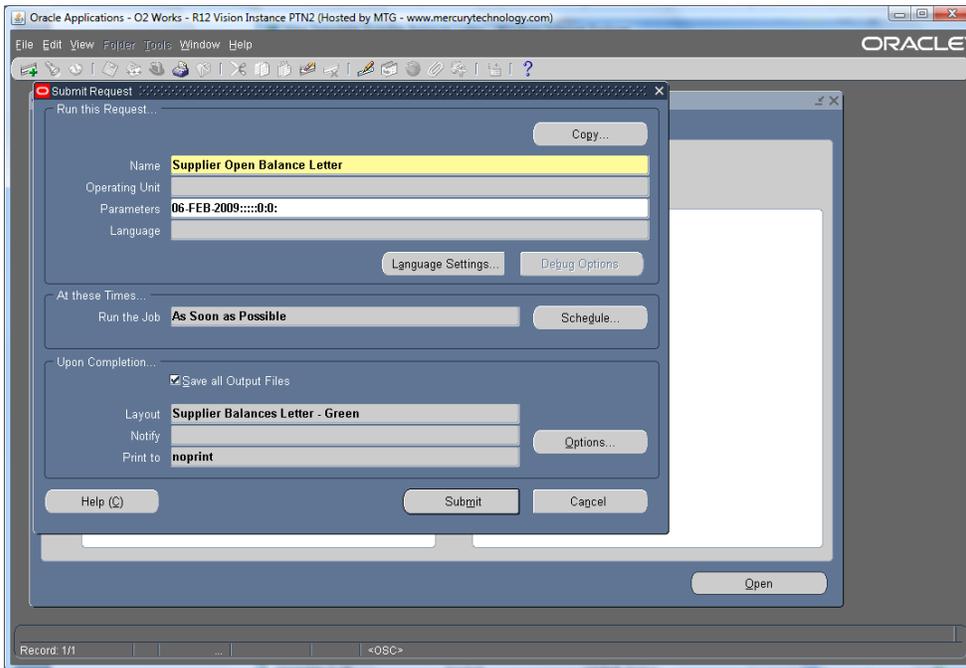
For a BI Publisher Reports concurrent program, the output is set to XML and a BI Publisher Data Definition and Template are created and assigned to the concurrent program.



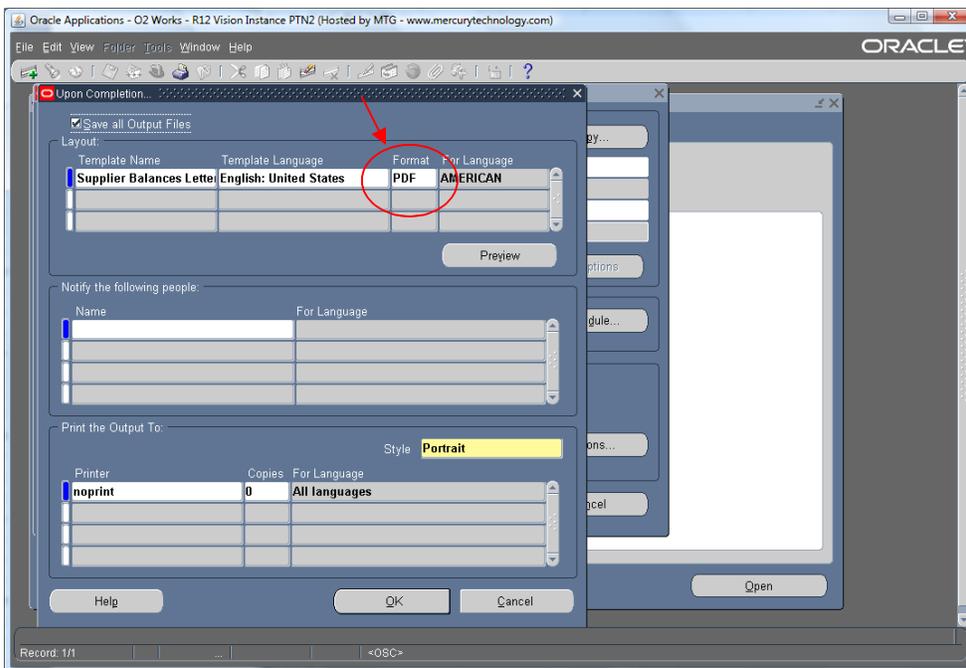
The Code used in the BI Publisher Data Definition must match the concurrent program short name. This is the link between the data definition and concurrent program, allowing the E-Business Suite to know when to call BI Publisher and produced output.



In the case of a BI Publisher Reports concurrent program, the user again selects the concurrent program to be run, sets the number of copies to be printed and the destination printer, and then runs the program.



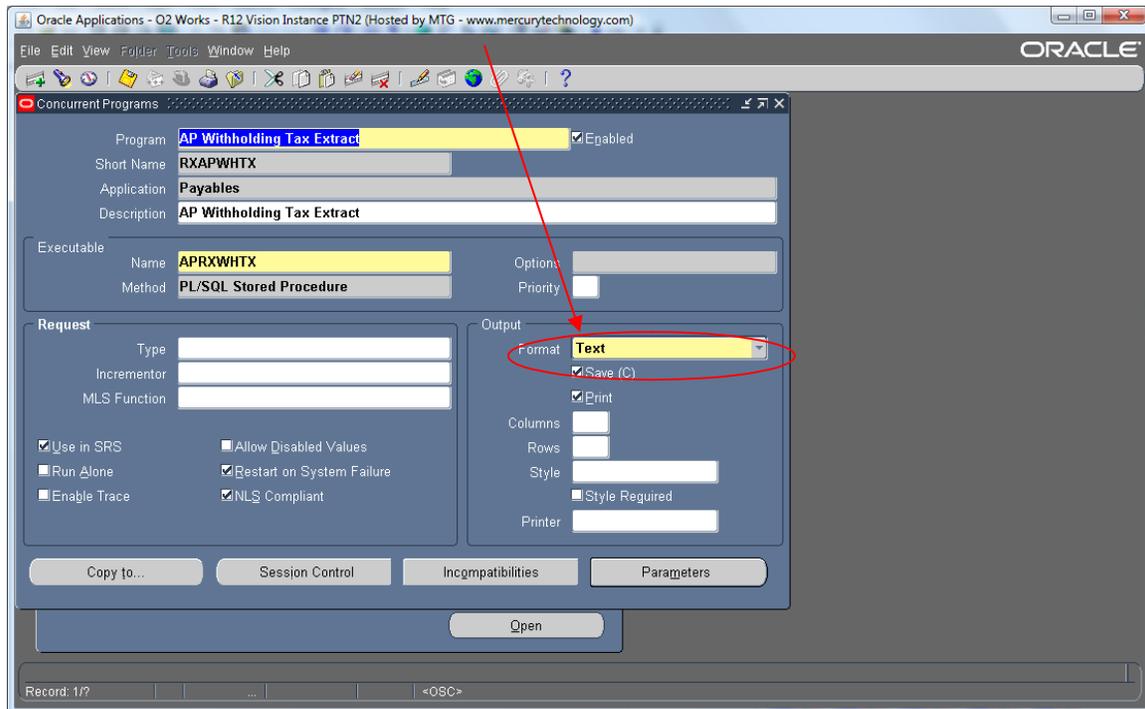
However, a BI Publisher Reports concurrent program also shows the BI Publisher Template to be used. The user can also choose the output format, most commonly either PDF or MS Excel.



When complete, opening the concurrent program output will display the BI Publisher output, in the format chosen by the user.

Now, to direct the output of a PL/SQL concurrent program to BI Publisher, we will still generate Text output, even if it is, in fact, XML. So, something is needed to allow BI Publisher to take the Text output and utilize it for its own.

First, the PL/SQL concurrent program must produce its output as a valid, well formed XML document. The output format will be Text, but the content will be XML.



Second is a “shell” concurrent program built around the PL/SQL concurrent program. This shell program is called to enable the adding of a BI Publisher Layout to the concurrent request generating the XML output.

```

PROCEDURE start_outbound
(
  errbuf          OUT NOCOPY VARCHAR2
  , retcode       OUT          NUMBER
  , p_debug_mode  IN          VARCHAR2   DEFAULT 'Y'
  , p_organization_id IN      NUMBER
  , p_delivery_date_low IN    DATE
  , p_delivery_date_high IN  DATE
  , p_freight_code IN        VARCHAR2
  , p_delivery_id  IN        NUMBER
  , p_trip_id     IN        NUMBER
  , p_delivery_leg_id IN    NUMBER
  , p_item_display IN        VARCHAR2
  , p_item_flex_code IN     VARCHAR2
) IS

  l_xml_layout      BOOLEAN      := FALSE;
  l_req_id         NUMBER;

BEGIN

  l_xml_layout := fnd_request.add_layout ('XXCUSTOM', 'XXCUS_WSHRDBOL', 'en', 'US', 'PDF');
  l_req_id := fnd_request.submit_request( 'XXCUSTOM' -- application
  , 'XXCUSECBOL' -- program
  , 'XXEDI/Memo Outbound Bill of Lading PDF Output' -- description
  , NULL -- start_time
  , FALSE -- sub_request
  , p_debug_mode -- P_DEBUG_MODE
  , p_organization_id -- P_ORGANIZATION_ID
  , p_delivery_date_low -- P_DELIVERY_DATE_LOW
  , p_delivery_date_high -- P_DELIVERY_DATE_HIGH
  , p_freight_code -- P_FREIGHT_CODE
  , p_delivery_id -- P_DELIVERY_ID
  , p_trip_id -- P_TRIP_ID
  , p_delivery_leg_id -- P_DELIVERY_LEG_ID
  , p_item_display -- P_ITEM_DISPLAY

```

```

, p_item_flex_code                                -- P_ITEM_FLEX_CODE
, CHR(0), CHR(0), CHR(0), CHR(0), CHR(0)
);

END start_outbound;

```

The PL/SQL concurrent program simply sets the appropriate BI Publisher layout Template and then calls the PL/SQL concurrent program which produces the require XML output.

The following line takes the place of the Upon Completion Options and assigns the appropriate BI Publisher Template, hard coding the BI Publisher Output Format to PDF. It is possible that a Value Set could be created for the shell concurrent program allowing the user to select the BI Publisher Output type.

```
l_xml_layout := fnd_request.add_layout ('XXCUSTOM', 'XXCUS_WSHRDBOL', 'en', 'US', 'PDF');
```

The user then calls the shell concurrent program, with all the parameters simply passed on to the main concurrent program. When complete, opening the concurrent program output will display the BI Publisher output, in the format chosen by the user.

