Learning R Series
Session 5: Oracle R Enterprise 1.3 Integrating R Results and Images with OBIEE Dashboards

Mark Hornick
Oracle Advanced Analytics
### Learning R Series 2012

<table>
<thead>
<tr>
<th>Session</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Session 1</td>
<td>Introduction to Oracle's R Technologies and Oracle R Enterprise 1.3</td>
</tr>
<tr>
<td>Session 2</td>
<td>Oracle R Enterprise 1.3 Transparency Layer</td>
</tr>
<tr>
<td>Session 3</td>
<td>Oracle R Enterprise 1.3 Embedded R Execution</td>
</tr>
<tr>
<td>Session 4</td>
<td>Oracle R Enterprise 1.3 Predictive Analytics</td>
</tr>
<tr>
<td>Session 5</td>
<td>Oracle R Enterprise 1.3 Integrating R Results and Images with OBIEE Dashboards</td>
</tr>
<tr>
<td>Session 6</td>
<td>Oracle R Connector for Hadoop 2.0 New features and Use Cases</td>
</tr>
</tbody>
</table>
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 decisions. The development, release, and timing of any features or functionality described for Oracle’s products remain at the sole discretion of Oracle.
Topics

• Introduction
• Integration with OBIEE RPD for structured data
• R PNG output generation through SQL
• Setting up the RPD for image data
• Dashboard parameters for user-customized graphs
• Demonstration video
• Summary
Introduction
Architecture: R Script Execution through RPD
Integration with OBIEE RPD

• Invoke R calculations from OBIEE
  – Define ORE-based SQL query via RPD table definition
  – Embedded R script execution
  – Reference R scripts from database table repository

• Retrieve results from R
  – Structured tabular output
  – Graphic PNG stream output
## Options for integration with OBIEE and BI Publisher

<table>
<thead>
<tr>
<th>Result Type</th>
<th>Format</th>
<th>OBIEE</th>
<th>BI Publisher</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structured results from R</td>
<td>Oracle table in RPD</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td></td>
<td>XML</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Images from R</td>
<td>XML</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>PNG image stream in RPD (BLOB column)</td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Images and structured results from R</td>
<td>XML</td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>
Of the 36 busiest airports, which are the best/worst for Arrival Delay?

- This boxplot is generated using the Oracle R Enterprise transparency layer.
- The data reside in a table stored in Oracle Database.
- Access from R occurs using an ore.frame.
- Statistical computations to generate the boxplot are performed in-database, avoiding data movement.
begin
  sys.rqScriptCreate('BusiestAirports',
    ' function(dat){
      ontime <- dat
      n <- 36
      aggdata <- aggregate(ontime$DEST, by = list(ontime$DEST), FUN = length)
      minx <- min(head(sort(aggdata$x, decreasing = TRUE), n))
      busiest_airports <- aggdata$Group.1[aggdata$x >= minx, drop = TRUE]
      delay <- ontime$ARRDELAY[ontime$DEST %in% busiest_airports '||'|' ontime$YEAR == 2007]
      dest <- ontime$DEST[ontime$DEST %in% busiest_airports '||'|' ontime$YEAR == 2007, drop = TRUE]
      dest <- reorder(dest, delay, FUN = median, na.rm = TRUE)
      bd <- split(delay, dest)
      res <- boxplot(bd, notch = TRUE, col = "gold", cex = 0.5,
        outline = FALSE, horizontal = TRUE, yaxt = "n",
        main = paste("2007 Flight Delays by Airport, top ",n, " busiest",sep=""),
        xlab = "Delay (minutes)", ylab = "Airport")
      labels <- levels(dest)
      text(par("usr")[1] - 3, 1:length(labels), srt = 0, adj = 1,
        labels = labels, xpd = TRUE, cex = 0.75)
      topbusiest <-
        data.frame(UNIQUECARRIER=res$names,
          MIN=res$stats[1,], Q1=res$stats[2,], CONF_LOW=res$conf[1,], MEDIAN=res$stats[3,],
          CONF_HIGH=res$conf[2,], Q3=res$stats[4,], MAX=res$stats[5,])
    }
  end;
/
36 busiest airports data

```sql
select UNIQUECARRIER, MINI, Q1, CONF_LOW, MEDIANI, CONF_HIGH, Q3, MAXI
from table(rqTableEval(cursor(select * from ONTIME_S),
    cursor(1 "ore.connect" from dual),
    'select cast('a' as varchar2(10)) UNIQUECARRIER, 1 MINI, 2 Q1, 3 CONF_LOW, 4 MEDIANI, 5 CONF_HIGH, 6 Q3, 7 MAXI
    from dual', 'BusiestAirports'));
```
Start with the RPD

- Open RPD in offline edit mode
- Data Source Name
  - tnsnames.ora entry, or (no <CR>s)

(DESCRIPTION=
  (ADDRESS=(PROTOCOL=TCP)
    (HOST=adc211-228.us.oracle.com)
    (PORT=1521))
  (CONNECT_DATA=(SID=ore)))
Create new physical table

• Example initialization string

```sql
select UNIQUECARRIER, MINI, Q1, CONF_LOW, MEDIANI, CONF_HIGH, Q3, MAXI from table(rqTableEval(cursor(select * from ONTIME_S), cursor(select 1 "ore.connect" from dual), select cast("a" as varchar2(10)) UNIQUECARRIER, 1 MINI, 2 Q1, 3 CONF_LOW, 4 MEDIANI, 5 CONF_HIGH, 6 Q3, 7 MAXI from dual, 'BusiestAirports'));
```
Access the data

- Update Row Count
- View Data…
Set up business layer

- Drag table to Business Model and Mapping
- Drag again to create reference for relationship
  - Specify key columns
  - Keep only key column(s) in ID table
  - Select both tables and right click for Business Model Diagram → Selected tables only

©2013 Oracle – All Rights Reserved
Business Layer to Presentation Layer

- Click join tool and click/drag from key table to fact table
- Click OK on Logical Join dialog
- Drag fact table to Presentation Layer
R PNG Output Generation through SQL
rqTableEval – 36 busiest airports graph

set long 20000
set pages 1000
select *
from table(rqTableEval( cursor(select * from ONTIME_S),
cursor(select 1 "ore.connect" from dual),
'PNG','BusiestAirports'));

<table>
<thead>
<tr>
<th>NAME</th>
<th>ID</th>
<th>IMAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

• Execute the function that graphs the top 36 busiest airports
• Specify the input data as ONTIME_S
• Specify auto-connect with parameter cursor
• Return the results as a table with name, id and image columns
• View the PNG image data returned, which can be used in an OBIEE RPD
Goal

• Create an dashboard that displays an image generated from R script that is executed in Oracle Database through SQL
Setting up the RPD for Image Data
Create a New Physical Table

• Open the OBIEE .rpd file
  – You may first need to stop your coreapplication_obis using `opmnctl stopproc ias-component=<name>`

• Right click the schema under your ORE-enabled database
• Click “New Physical Table…”
Specify Columns and Keys

• Click the “Columns” tab
  – Add two columns corresponding to the select columns in the query – id and image
  – Id is an INT
  – Image is LONGVARBINARY(32000)
  – Click OK

• Click the “Keys” tab
  – Type “id” in the Key Name field
  – Type “id” in the Columns field
  – Click OK
Specify General Physical Table Properties

- Name
- Table Type = Select
- Default Initialization String
  - This is the SQL query that invokes your R script already stored in the R script repository in Oracle Database
  - Here, we’re using the “BusiestAirports” script introduced earlier
- Click OK
Move to the Business Model and Mapping

- Click and Drag the Schema (RQUSER in this case) to the Business Model and Mapping column of the Admin Tool
- Expand the node
- Right click to copy and paste (duplicate) the “BusiestAirports” table for joining in the next step
- Note: we deleted the NARROW and ONTIME_S tables from this node since they will not be used.
Join the BusiestAirports with “itself”

- Select both BusiestAirports and BusiestAirports#1
- Right click and select “Physical Diagram” and “Selected Object(s) Only”
Join the Two Tables

• Click the join tool with tool tip “New Join”
• Click BusiestAirports and drag to BusiestAirports#1
• Save the result and close the Diagram window
Set up the ‘image’ column

- Expand the “BusiestAirports” table
- Double click ‘id’
- Double click the “Logical Table Source” entry
Specify the image lookup details

- Click image column
- Click the “edit” icon on the right
Specify the lookup expression

• Before the “image” string already present, type `lookup(`
• Add a comma after the “image” string
• Double click ‘id’ to add to expression
• Add closing `)` at end of expression
• Click OK
Set “sort order” and “descriptor id” columns

- Double click ‘image’
- Click “Set…” and select the id column in the dialog window, click OK.
  - Do this for sort order and descriptor id
- Click OK
Create the Presentation Layer Entry

- Click and drag the (in this case) RQUSER schema from the Business Layer to the Presentation Layer.
Save the RPD

- After saving, you will need to restart the coreapplication_obis component

```bash
opmnctl startproc ias-component=<name>
```
Creating the Dashboard
Login to OBIEE and Reload Files and Metadata

- After logging in, scroll down to “Maintenance and Troubleshooting” section
- Click “Reload Files and Metadata”
Create a new Analysis

- Click the “New” menu
- Click “Analysis”
- Click “RQUSER” since this is the schema are working with
Select Columns under the “Criteria” tab

• Click and drag first the ‘id’ column, then the ‘image’ column to the “Selected Columns” section as shown.

• Click the “Results” tab to view the graph that is dynamically generated from the R script.
Save the analysis

• Click the “save” icon at top
• Navigate to the folder where you want the analysis
• Click OK
Create a Dashboard

• Click the “New” menu
• Click “Dashboard”
• Provide a name and location for the dashboard
• Click OK
Populate the Dashboard with the Analysis

- In the catalog, open the folder where you stored the analysis
- Click and drag “BusiestAirports-Analysis2” onto the canvas
- Click the save icon
View the finished Dashboard

- Click Run
Voila!

- The image is dynamically generated when the dashboard is displayed.
- This occurs through the execution of the SQL query that invokes the R function stored in the database R script repository.
Dashboard Parameters for User-Customized Graphs
Goal
RandomRedDots with parameter

begin
    sys.rqScriptCreate('RandomRedDots',
    'function()
    {
        plot( 1:10, rnorm(10), pch = 21,
            bg = "red", cex = 2 )
    }');
end;
/

begin
    sys.rqScriptCreate('RandomRedDots3',
    'function(n)
    {
        plot( 1:n, rnorm(n), pch = 21,
            bg = "red", cex = 2 )
    }');
end;
/

set long 2000
set pages 1000

select id, image
from table(rqEval(cursor(select 100 "n" from dual),'PNG', 'RandomRedDots3'));
Change Query to use Session Variable for Parameter

- Default Initialization String
  - For parameters, use NQ_SESSION variable

```sql
select id, image
from table(rqEval(cursor(select
    valueof(NQ_SESSION.NumRedDots) "n" from dual),
    'PNG', 'RandomRedDots3'));
```
Create variables…

- Under the “Manage” menu, select “Variables…”

- Brings up the Variable Manager
Create initialization block

1. Variable Manager
2. New Initialization Block...
3. Name: NumRedDotVar
4. Data Source:
   - OBIEE Server
     - select 100 from dual
5. Variable Target:
   - Name: NumRedDots
     - Default Initializer: 100
6. Execution Precedence:
   - No execution precedence setting was made
Create a new session variable

- Click on “non-system”
- Right click in the area on the right and select “New Session Variable…”
- Complete session variable by
  - naming it “NumRedDots”
  - Enable any user to set value
  - Choose Initialization Block as NumRedDotsVar
  - Specify the default value, e.g., 100
Create a dashboard prompt

• Click “Dashboard Prompt” under the “New” menu
• Under the “plus” sign, select “Variable Prompt”
Edit prompt
Format prompt
Finished Dashboard
What else is possible?

- **Multiple images**
  - If an R script generates multiple images, these will be returned as multiple rows with distinct ids.
  - These images can be displayed using the same approach detailed in this presentation.

- **Structured data**
  - The SQL query that invoked the R script specified ‘PNG’ as the output format caused any non-image stream data to be discarded.
  - If the R script returns structured data, e.g., a `data.frame`, ‘PNG’ can be replaced with a SQL `SELECT` definition that describes the structure to SQL, e.g., `SELECT 1 id, 'a' name, 1 cnt FROM dual`.
  - This output will appear as a regular table that can be exposed through the RPD as any other table.

- **XML output**
  - Replacing ‘PNG’ with ‘XML’ will generate an XML representation of both structured data results and image results together.
  - Images are expressed as a base 64 encoding of the PNG file.
What else is possible?

• Advanced Calculations
  – R script that tests for statistical significance between processes or products using distribution analysis and weighted t-tests, returning numerical results, p-value scores and computations
  – From OBIEE dashboards, dynamically built predictive models can be executed and fitted to data or a subset of data, with model diagnostics and details streamed back to the dashboard
  – What-if analysis and estimating causal effects based on data can be executed using different R solvers with result sent back to OBIEE for graphing and reporting
  – Markov Chain Monte Carlo Simulations is also a technique widely used across different industries, the results of which can be exposed through OBIEE dashboards summarizing the parallel execution enabled by ORE embedded R execution
Summary

• ORE embedded R execution SQL interface enables integration with OBIEE RPD and Dashboards
• Incorporate structured data as a table or XML
• Incorporate graphics as XML or PNG image stream
• Parameterize R scripts to allow dynamic control of data computations or generated images