EBS Reports and Dashboards for the People and by the People

Patrick Cimolini
Insum

Abstract:

Learn how to create flexible ad-hoc reports and dashboards by using Oracle Application Express (APEX) to present your Oracle E-Business Suite (EBS) data. EBS uses older reporting technologies that tend to present static report layouts. Also, EBS drilldown dashboards are difficult to implement. APEX circumvents these shortcomings by allowing you to easily create an EBS reporting environment by using APEX Interactive Reports and Dashboards that have powerful drilldown capability.

About Insum

Founded in 2002 and dedicated to APEX since 2004, Insum is a center of expertise and innovation specialized in Oracle databases and the APEX development tool.

Insum's main office is located in Montreal, Canada where its development group is concentrated. Insum also have an executive office in South Burlington, Vermont, and multiple developers and analysts located across Canada and the US.

Recently, Insum has been promoting the integration of APEX with EBS. The goals for this integration are:

- Make it easier to customize EBS at lower cost.
- Add enhanced functionality that is not native to EBS.
- Make EBS upgrades easier by building customizations externally so that the customizations do not get overwritten by the EBS upgrade process.

This integration strategy also enables EBS organizations to leverage their existing SQL and PL/SQL skills because APEX is largely based on these existing Oracle technologies.

Introduction

This white paper first describes, at a high level, how the integration between APEX and EBS is set up. Secondly, the paper shows how APEX can enhance EBS through the use of APEX's Interactive Reports and graphic dashboards.

APEX Development in an EBS Environment

APEX is used as a "bolt-on" technology when it is used to customize EBS. APEX takes advantage of the rich Application Programming Interface (API) that EBS exposes to the external world. This architecture (Figure 1) separates customizations from the main EBS installation which avoids the problem of internal EBS customizations being overwritten when EBS is upgraded. This is a great advantage because the external customizations do not need to be re-applied after EBS is upgraded, they only need to be adjusted if the EBS API changes. This strategy dramatically lowers the cost and time it takes to upgrade EBS.

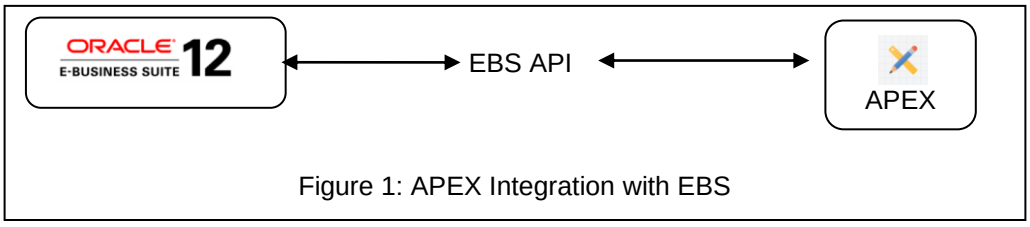

Figure 1: APEX Integration with EBS

What are some of the use cases for customizing EBS with APEX? Five obvious areas are (Figure 2):

- Lite Interfaces
- Mobile
- Interactive Reports
- Drill-down Dashboards
- RESTful Web Services

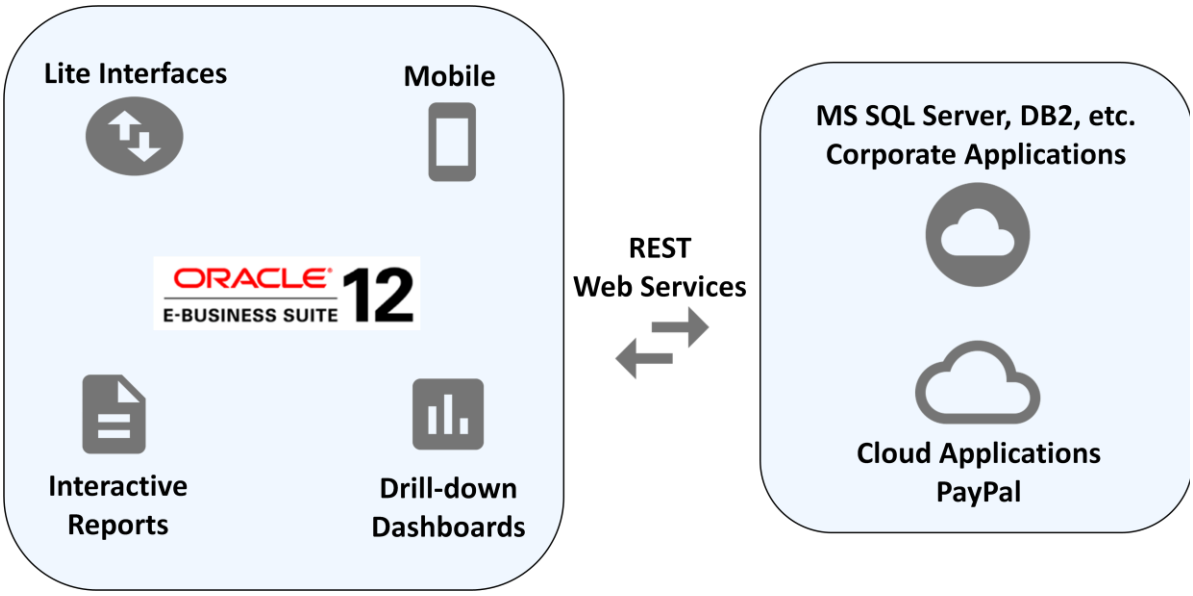

Figure 2: Use Cases for EBS with APEX Integration

Lite Interfaces: Lite interfaces that are used for repetitive data entry make life much easier for the data entry personnel and the EBS administrators. The data entry personnel are given an APEX interface that is tailored to a small subset of EBS transactions. The lite interfaces make the data entry task faster and more accurate when compared to using the out-of-the-box EBS data entry screens. The EBS screens tend to be large, cumbersome, and full of complex functionality that is not needed for routine transactions. The EBS administrators are happy because there are fewer mistakes made by the data entry personnel; in other words, there are fewer messes to clean up at the month end.

Mobile: APEX, out of the box, is mobile ready. Demand for phone and tablet interfaces is strong and APEX can easily supply those interfaces.

Interactive Reports: Interactive Reports are a marquee feature of APEX. These reports give the end users a great deal of power over their data. The reports allow the end users to show/hide columns, sort data, highlight rows and cells, add calculated columns, aggregate data, group data, add charts, and save their reports either publicly or privately. The ability to pivot data was added in APEX 5; the ability to pivot data has made power users extremely happy.

Drill-down Dashboards: APEX allows both reports and charts to drill deeper into more detailed data. In turn, the deeper data is displayed in either report or chart formats that are pre-filtered by the calling page. This is a significant improvement over standard EBS reporting.

RESTful Web Services: The APEX mid-tier uses Oracle REST Data Servers (ORDS). One of the marquee features of ORDS is support for developing REST interfaces between APEX and the external world. This feature allows APEX to exchange data with many other computing environments. This ability to exchange data makes APEX a serious player in the enterprise and cloud worlds.

Why Oracle Application Express (APEX)?

APEX, which is now well over ten years old, is a mature web development tool that produces web applications that are tightly integrated with the Oracle database. The APEX architecture allows developers to easily take advantage of the incredible power in Oracle's database and also serve up functional and scalable web pages that can be tailored into elegantly branded internal or external applications.

Now why would someone use APEX specifically in an EBS environment? Here are a few points:

- APEX is a no cost feature of the Oracle database. It is, in effect, free. EBS users who have purchased an EBS limited license might need to adjust their EBS license so that they can create the custom schemas that are required when adding a bolt-on technology such as APEX.
- APEX is a rapid application development (RAD) tool. Much of the coding is done through a declarative web interface.
- An existing EBS team will be able to quickly get up to speed with APEX because the team members are already experts with SQL and PL/SQL which are the core technologies that are used in an APEX environment.
- Seamless integration between APEX and EBS is readily available.
- APEX reporting and dashboard applications eliminate much of the need for dumping EBS data into spreadsheets. This is an important security point.
- APEX contains an out of the box mobile interface.

Technical Requirements

The technical requirements for adding APEX to an EBS environment is fairly straight forward.

First, you need, of course, a fully licensed Oracle database. Second, you need EBS version 11i or above. Installing EBS is not a simple task; however, this is already done for most of the readers of this document. Third, you need to activate APEX in your database and install ORDS. These last two steps are very straight forward and easy to do.

Just one note, check the version of APEX that came with your database and upgrade it to the latest version. The upgrade process is quick and easy.

Both APEX and ORDS downloads are free on their Oracle Technology Network (OTN) home pages.

Recommended Architecture

Oracle recommends using a traditional three tier architecture (Figure 3). This is the fundamental structure. The configuration of the middle tier varies slightly depending on the version of EBS. An Oracle white paper is available that clearly documents the variations. The latest version (at the time of writing this document) is dated March 2015 and is located at:

`http://www.oracle.com/technetwork/developer-tools/apex/learnmore/apex-ebs-extension-white-paper-345780.pdf`

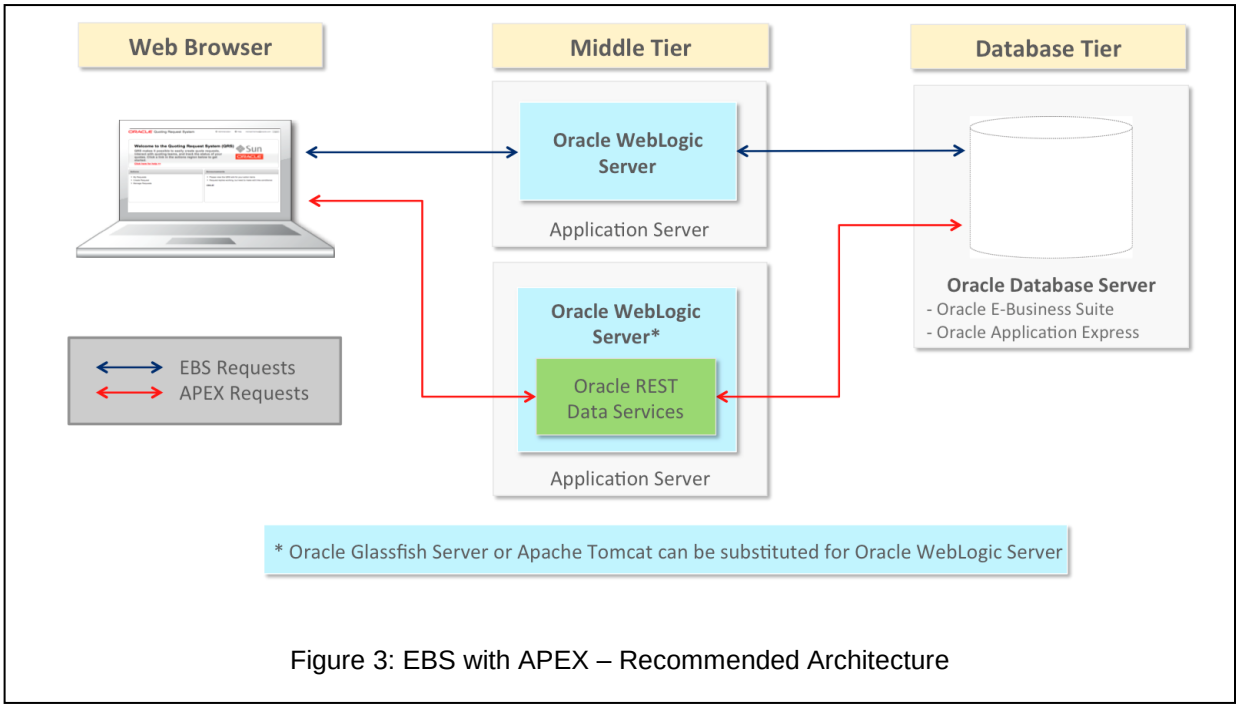

Figure 3: EBS with APEX – Recommended Architecture

Accessing EBS Data from APEX

The mechanical hookup between EBS and APEX for data and process sharing is straight forward (Figure 4).

An APEX application, from the end user's perspective, runs in a browser that is installed on a desktop computer, a tablet, or a smart phone. The APEX application communicates with the APEX engine through ORDS.

APEX requires a "parsing schema". This Parsing schema (XXAPEX) contains all of the database objects that the APEX applications use directly. Typically, these objects consist of views and PL/SQL packages that communicate with the EBS APPS schema either directly or indirectly through other custom EBS schemas (XX... schemas). The interaction with the EBS schemas is done via GRANTS to a rich set of PL/SQL APIs, interface tables, and REST services.

How is authentication handled in this environment? Easily if Oracle Single Sign On (SSO) is employed; in this case EBS and APEX are registered as partner applications. In addition, code is added to APEX so that APEX can access authorization information after APEX is authenticated. If SSO is not employed then authentication is a bit trickier. Happily this situation has been solved by Insum Solutions Inc. (www.insum.ca) and Concept 2 Completion (concept2completion.net) who have made a solution available through open source. The code can be downloaded for free from Insum's website. This secure solution involves the use of electronic tokens that are valid for a few seconds.

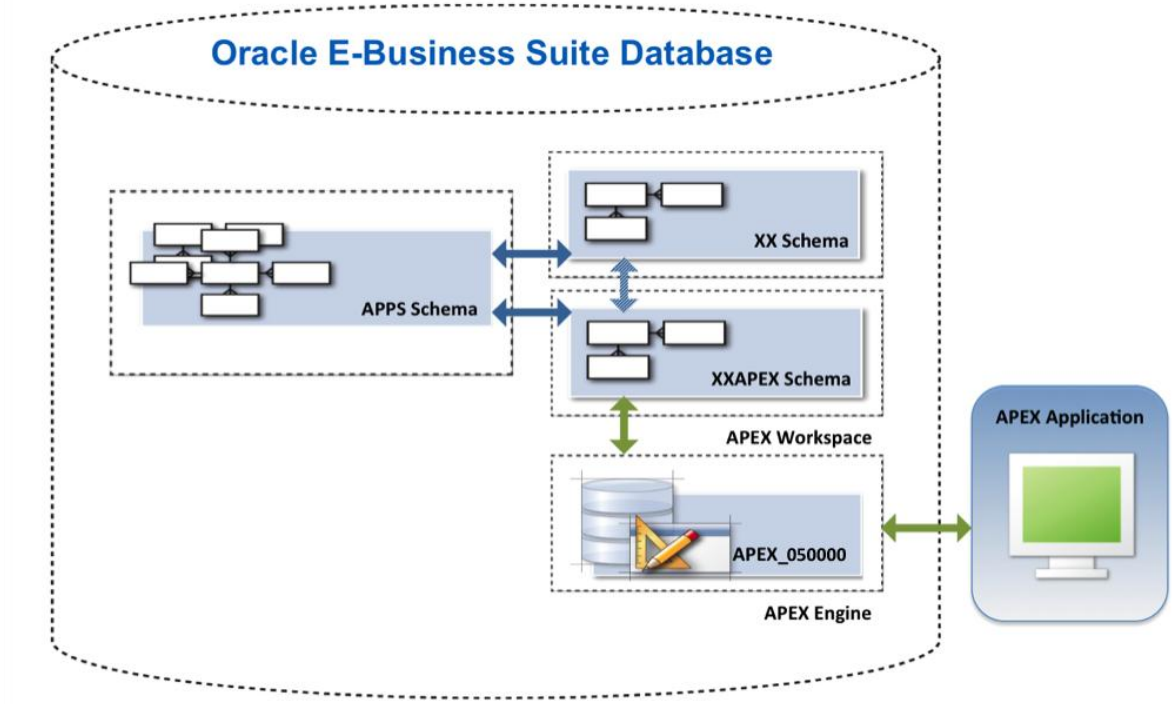

Figure 4: Accessing EBS Data from APEX

EBS Reports and Dashboards for the People and by the People

Now the main part of this paper begins. The following sections discuss how APEX interactive reports and dashboards can be used to great effect in an EBS environment. It also discusses the division of labor between the end users and APEX developers.

Interactive Reports – End User Interface

When you first look at an interactive report you immediately see that a rich set of functionality is available to the end user (Figure 5).

The top menu bar allows the end user to:

- Select columns that will be searched. Searching all text and numeric columns is the default.
- Search text is case insensitive.
- There is a list of saved public and private reports.
- Report and chart views are available if charts have been created.
- The number of rows to display can be picked. This is important when the report can potentially return a large number of rows. Returning a large number of rows can be a lengthy task.
- The Action Menu gives the end user tremendous power over the formatting of the Interactive Report. The Compute feature allows the end user to create calculated columns. This is an incredibly powerful feature than can be creatively combined with the other formatting features to achieve results that are not obvious at first glance; power users who have good MSExcel skills will love this feature. New in APEX 5 is the ability to Pivot the table.
- The report can be expanded to consume the full screen. This is handy for wide reports.

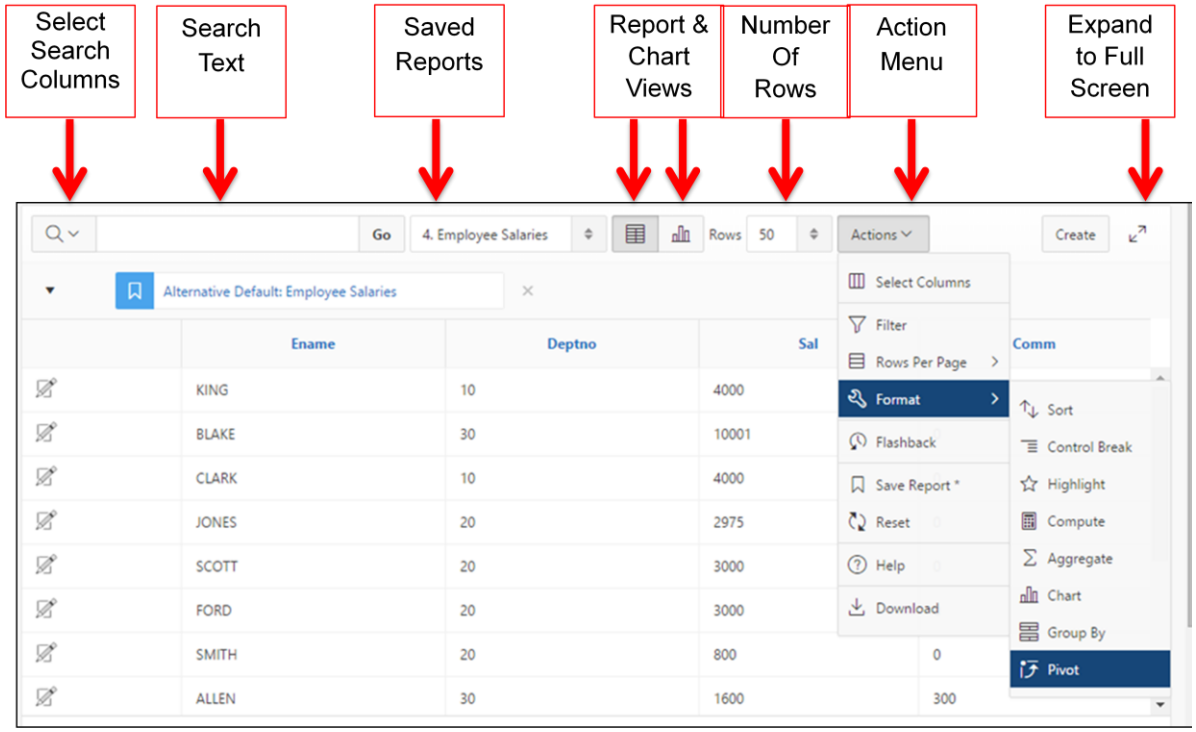

Figure 5: Interactive Report End User Interface

Interactive Reports – Division of Labor

Cooperation between the end user and developer communities is required for an organization to realize the full benefits of Interactive Reports.

End users can heavily customize their reports by using the Action Menu. End users can quickly enhance an interactive report by using the formatting features individually. When the formatting features are combined, then the end users can achieve very creative and effective results that turn the data into easily absorbed information that enhances the decision making process.

Oracle provides Interactive Report documentation that targets end users. The documentation is found on the OTN APEX home page (Figure 6).

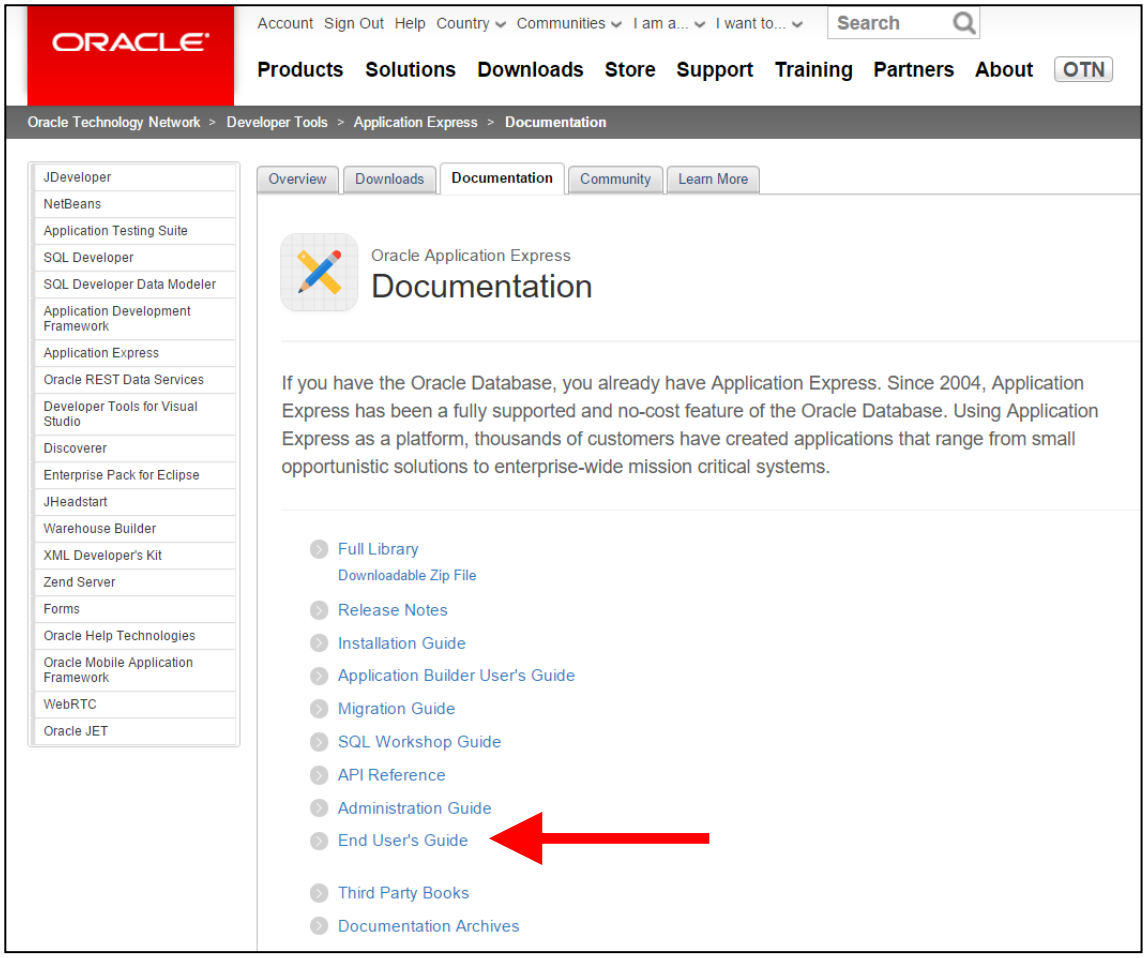

Figure 6: Interactive Report Documentation for End Users

Interactive Reports free developers from the tedious task of constantly reformatting the trivial aspects of reports such as column order, sorting, show/hide columns etc as the end users can easily perform these tasks fro themselves. The developers, however, still must do some work to set up the default Interactive Report so that the end users have a starting point. The developers are responsible for:

- Defining the SQL statement that builds the report. Typically, developers will create a view that contains as much data as possible and then let the end users pick and choose which columns to display.
- Drill down links to other reports and charts are coded by the developers.
- The features that are given to the end users are defined by the developers. For example, not all organizations want end users to have access to the Flashback feature that allows end users to see their data as it was in the past. Turning this feature off is done with a simple check box in the development environment; all of the other features can be turned on or off in the same manner.
- Create the “Default” report for the end users. The default report is based on the SQL statement and the Action Menu features that have been selected. The default report is typically an output from the application’s design documents.
- Developers can add high level filters as dropdown lists outside of the Interactive Report to limit the number of rows that are returned to the Interactive Report. This step is required where there is the possibility of returning tens of thousands of rows to the Interactive Report. Returning too

many rows can be very slow. The filtering design is based on the size of the potential result set and the capabilities of the end user's computer, tablet, or smart phone.

Dashboards – End User Interface

Dashboards are a common design pattern that can be built by most common programming environments. What makes APEX attractive in this area? The primary reason is the declarative nature of APEX programming that makes building a dashboard interface quick and relatively easy for the developers. In other words, when a functional end user has a specific need, developers will be able to fill that need in a short period of time when they use APEX.

A typical use case for dashboard drilldowns is illustrated in Figure 7. An end user's attention is caught by an unusually large bar in a bar chart. Clicking on the offending bar takes the end user immediately an Interactive Report that contains the details that built the bar. The end user then uses the power that is built into the Interactive Report to find the offending line item(s).

At this time with APEX 5 there is no way for an end user to build a dashboard environment within the context of an APEX application that is linked to EBS. The division of labor falls heavily upon the developers. The good news here centers on the declarative nature of APEX that allows developers to quickly assemble a dashboard environment.

Note: *End users could conceivably assemble their own dashboard environment using the Websheet feature of APEX. Integrating Websheets to EBS is out of scope for this presentation but would be a fascinating research project.*

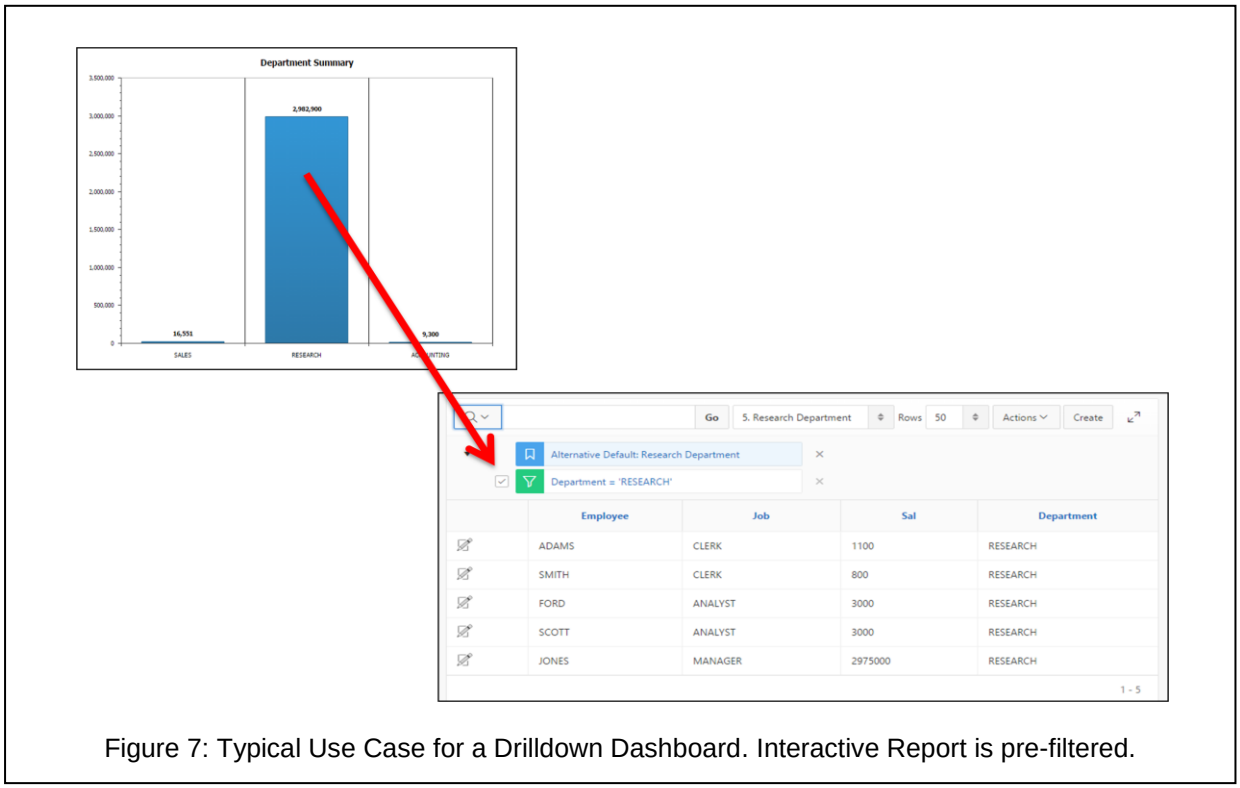

Figure 7: Typical Use Case for a Drilldown Dashboard. Interactive Report is pre-filtered.

There are three fundamental tasks that developers need to do when they build a dashboard environment:

- Build charts
- Build interactive reports
- Build links between the charts and reports.

The bulk of the work is done declaratively with just a bit of coding for the links. The coding is simple and well documented.

Building a chart is done by walking through a step by step wizard. One of the steps asks the developer to code the chart's underlying SQL statement that contains a link to another page. The SQL syntax for each chart type varies slightly; however, the wizard always provides SQL snippets that can be used as a copy-and-paste template (Figure 8). The SQL snippets eliminate the need to look up the syntax in documentation.

The syntax for linking a chart to underlying drilldown charts and reports uses the standard APEX URL syntax. This URL syntax (Figure 9) shows you how the syntax can be used to

- navigate to regular APEX pages
- navigate to and pre-filter Interactive Report pages

In both cases, the URL syntax is simple and easy for developers to understand and code. A more complete definition, of course, is found in the Oracle documentation.

Chart Query Example for 3D Column

```
SELECT NULL LINK,
       ENAME LABEL,
       SAL VALUE
FROM   EMP
ORDER  BY ENAME

SELECT 'f?p=&APP_ID.:2:'||:APP_SESSION||'::::P2_ID:'||EMPNO LINK,
       ENAME LABEL,
       SAL VALUE
FROM   EMP
ORDER  BY ENAME

SELECT NULL LINK,
       ENAME LABEL,
       SAL "Salary",
       COMM "Commission"
FROM   EMP
ORDER  BY ENAME
```

Figure 8: SQL Snippet for a Chart – Available in the wizards for all Chart Types

	Place holders, colon delimited		Definition
	f?p=		f is a function p is the parameter
1	APP_ID :		Application ID or Alias
2	APP_PAGE_ID :		Page ID or Alias
3	APP_SESSION :		Session ID
4	REQUEST :		HTML Request
5	DEBUG :		Yes / No flag
6	Clear Cache :		List of pages to clear
7	item1, item2 :		List of items to pass
8	itemValue1, itemValue2 :		Values of the items
9	printerFriendly		Yes / No flag

Syntax for pre-filtering Interactive Reports

4	REQUEST :	(selects a report)	IR[region static ID]_<report_alias>
6	Clear Cache :	(reset the report)	RIR and/or CIR and/or RP
7	item1 :	(filter the report)	IR[region static ID]<operator>_<target column alias>
8	itemValue1 :	(value for the filter)	Filter values

Examples:

This is a handy snippet for copy-and-paste

```
f?p=app:page:session:request:debug:clearCache:items:values:printerFriendly
```

Link to page 200, clear cache on page 200, and pass the value 5555 to page item P200_EMPNO.

```
f?p=&APP_ID.:200:&APP_SESSION.:::200:P200_EMPNO:5555
```

Link to page 200, select the SALARIES saved report in the EMP Interactive Report and return rows where the SAL column is greater than 5000. Reset the report and the pagination.

```
f?p=&APP_ID.:200:&APP_SESSION.::IR[EMP]_SALARIES::RIR,RP: IR[EMP]GT_SAL:5000
```

Figure 9: APEX URL Syntax

Conclusion

Using APEX as a “bolt-on” tool to customize EBS is an attractive strategy. Results can be achieved quickly due to the declarative rapid application development environment that is provided by APEX. APEX customizations are outside of EBS; therefore, they are not overwritten when EBS is upgraded. End users are given a tremendous amount of personal power and control through the use of Interactive Reports. Attractive and functional dashboards can be quickly built by developers to give end users visual information with the ability to conveniently drill down into multiple layers of detail. Finally, it is always good to remember that APEX is a “no cost” feature of the Oracle Database.