

## Erweiterung der Oracle EBS mit APEX: schnell, sicher & einfach!

**Yves Chassein**  
**PROMATIS software GmbH**  
**Ettlingen**

### Key Words

APEX, Oracle E-Business Suite, Customization, Rapid Application Development, Integration

### Introduction

With Oracle APEX (Oracle Application Express), it is possible to implement database applications quickly and only with the use of a web browser. These applications can be desktop or mobile ones. With its easy and declarative approach, Oracle APEX is perfect for a rapid but still professional development of database centralized web applications.

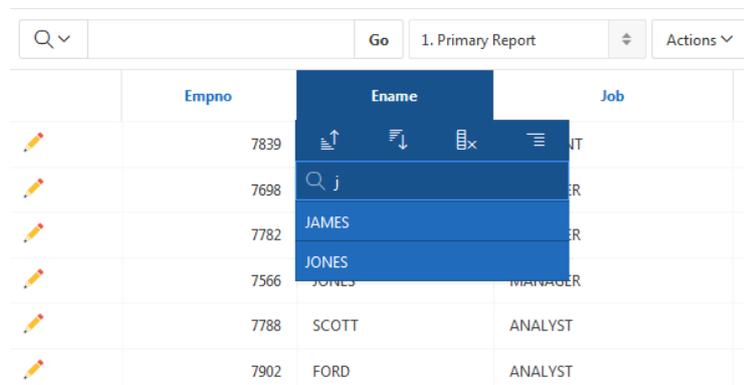
Using Oracle APEX, the functionality of the Oracle E-Business Suite can be easily extended. Since the end of forms support is announced and Oracle application framework customizations are difficult to learn for PL/SQL developers, APEX is an easy way to put well-known knowledge (SQL and PL/SQL) to use and extend the functionality of the Oracle E-Business Suite.

### What is APEX?

Oracle Application Express (APEX) is a development platform which uses SQL, PL/SQL, JavaScript and HTML to create applications on top of an Oracle database. Development in APEX only requires a browser. As a low-code platform, it enables users without much programming experience to create simple web applications quickly and without the need to code, but it is also possible to implement web applications with a high complexity like e.g. a customer portal. APEX applications are easily scalable and responsive. In case APEX uses SQL and PL/SQL as the main programming language, and in contrast to Oracle Application Framework (OAF), it is easy to learn for Oracle Forms developers.

Main features of APEX are:

- Creating reports on database tables, which users can modify themselves (interactive report) and export as .csv or .xls files.



	Empno	Ename	Job
	7839		IT
	7698	j	ER
	7782	JAMES	ER
	7566	JONES	MANAGER
	7788	SCOTT	ANALYST
	7902	FORD	ANALYST

Figure 1: Sorting and searching in an interactive report (Source: <https://www.oracle.com/>)

Empno	Ename	Sal	Loaded Salary
7839	KING	5000	\$5,500.00
7782	CLARK	2450	\$2,695.00
7834	MILLER	1300	\$1,430.00
			\$9,625.00
Deptno : 20			
Empno	Ename	Sal	Loaded Salary
7566	JONES	2975	\$3,272.50
7788	SCOTT	3000	\$3,300.00
7902	FORD	3000	\$3,300.00
7369	SMITH	800	\$880.00
7876	ADAMS	1100	\$1,210.00
			\$11,962.50

Figure 2: Color highlighting of rows (Source: <https://www.oracle.com>)

- Creating forms to enter, modify or delete data in database tables.

Employee Id	First Name	Last Name	Email	Phone Number	Hire Date	Job Id	Salary	Commission Pct	Manager Id
100	Steven	King	SKING	515.323.4567	17-JUN-1987	AD_PRES	24000	4	-
101	Neena	Kochhar	NKOCHHAR	515.323.4568	21-SEP-1989	AD_VP	17000	-	100
102	Lex	De Haan	LDEHAAN	515.323.4569	13-JAN-1993	AD_VP	17000	-	100
103	Alexander	Hunold	AHUNOLD	590.423.4567	03-JAN-1990	IT_PROG	9000	-	102
104	Bruce	Ernst	BERNST	590.423.4568	21-MAY-1991	IT_PROG	8000	-	103

Figure 3: Interactive grid to enter, modify or delete data (Source: <https://www.oracle.com>)

APEX is part of every Oracle database and does not require a separate license, which means it can be used without additional costs.

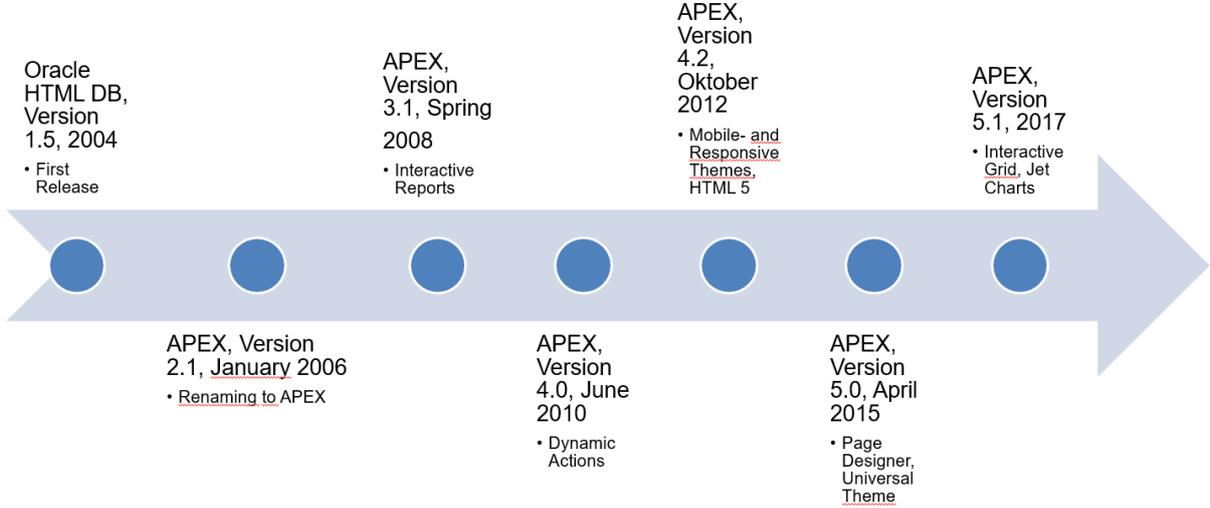
## History

The first version of APEX was released in 2004, then named HTML DB. With version 2.1 in 2006, it was first called APEX.

With version 3.1 and one year later, interactive reports were introduced. These reports enabled users to use special filters and aggregation functions on their reports to further analyze and sort the displayed data.

In 2010, version 4.0 introduced Dynamic Actions and Plugins, enhancing the development tools enormously. With the release of APEX 5.0 in 2015, the whole IDE was overhauled. The development was now done on only one page - the page designer-, instead of on multiple ones (each component used to have its own page). The introduction of Universal Theme enabled developers to easily customize applications and develop responsive user interfaces.

Version 5.1 introduced the interactive grid, which combined the functionality of forms and interactive reports. Since May 2018, the name is no longer APEX 5.1(.4), but instead 18.1, and going for the new global scheme at Oracle.

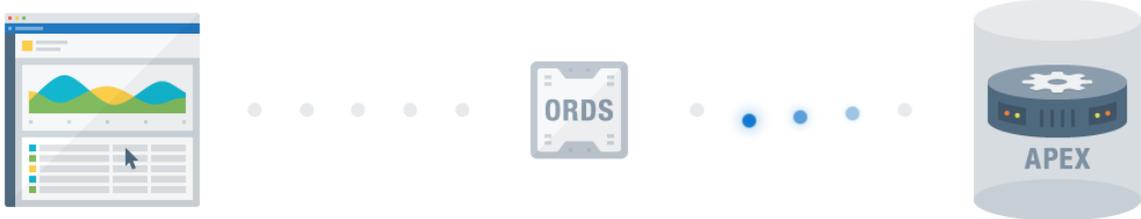


Features like the interactive report or grid are perfectly suitable for extending the functionality of the Oracle E-Business Suite.

**APEX Architecture**

APEX uses a 3-tier architecture. Through a web server, requests are sent from the browser to the database. All business logic is executed in the database. For example on the web server, an Oracle Rest Data Service (ORDS) can be used which takes the request and hands it to the database where it is processed subsequently. Once the processing is finished, the result is sent back to the browser via ORDS. Instead of the ORDS, it is also possible to use the web server which is integrated in the DB. It is not recommended by Oracle to do so on a productive instance. Another but deprecated possibility is to use the Oracle HTTP Server together with mod\_plsql. The recommended way is to use ORDS. It is possible to use ORDS in standalone mode (only for test instances) or deployed on an application server like Tomcat or Weblogic.

In case of an integration with the Oracle E-Business Suite, it is possible to install APEX in the same database and install an additional web server on the physical application server of the EBS. It is also possible to install APEX on a separate database and application server. To access the EBS data, a database link between the two instances is needed. In many cases, it is a good choice to install an additional Tomcat server on the physical EBS application server where to deploy the ORDS.



**Figure 4: Architecture of Oracle APEX and ORDS (Source: <https://apex.oracle.com>)**

## Integration between Oracle E-Business Suite and Oracle Apex

Innately, APEX comes with several options to implement a user login. Out of the box, it is possible to log into your application with APEX credentials or a database user. It is certainly possible to use one of these login mechanisms to allow users access to the application, but in case of the EBS, this would not suffice. In that case, it should be at least possible to use the same login credentials the user uses in the EBS. In APEX, this can be achieved by using a custom authentication function. For a better user experience however, a single sign on instead of a single authentication is necessary. In addition to the single sign on, the integration should consider these requirements:

- Login/SSO
  - o Coming from the EBS, no password should be needed for the login.
  - o EBS-Session-ID, responsibility-, user- and organization information should be handed over securely to APEX via the URL.
  - o The login mechanism should work without using the session cookie of the EBS (in case the host of the ORDS is not the same as the host of the EBS).
  - o Login-URL should only be valid for a limited time (no passing of the URL to allow other users to sign in).
- Session Management
  - o Every call of an APEX page should tickle the EBS session, if started from there.
  - o If the session timed out, the user should be redirected to the EBS login form.
  - o The logout button in APEX and in the EBS should destroy the corresponding session in the other application as well.
  - o Users should be able to log into APEX itself without logging into the EBS.
- Menu in APEX
  - o The menu which the user can see in APEX should be controlled by the security function within the EBS, meaning that every page in APEX needs a connection to an EBS-function.
- Authorization
  - o Every page should be connected to a function in the EBS.
  - o On page access, the authorization function checks whether the user has access to the corresponding EBS function.
- Context initialization
  - o This should be done by every APEX page if needed.

There are several available approaches which will capture some, but not all of these requirements. A straightforward approach is to deliver the EBS username as a parameter to APEX. The authentication function only checks whether there is a session for the user or not. If that is the case, the “user” is

granted access to the application. If there is no session, the access to the application is denied. This solution is highly insecure, since even without access to EBS, usernames can be guessed and inserted into the URL in order to gain access to the application.

What we require is a safe way without any clear text in the URL. To meet this requirement, we need to create a token for the URL in EBS, which is then sent to APEX and decoded there to log in the user. The token consists of the username and the responsibility, session and organization information. Together with these attributes in the token, we will also use the session ID of EBS in APEX. This will make it possible to end the session in EBS as we log out of APEX, and will also leave our session in EBS untouched if we have logged in there directly from APEX.

To prevent sessions from lying fallow, we built a timeout into APEX that corresponds to the session timeout implemented in the EBS by Oracle. Once the APEX session has timed out, the EBS session will timeout as well. The same functionality is implemented for the reserve situation. If the session is started directly in APEX, it will timeout as well.

Once inside the application, we require some kind of authorization, or otherwise all users, even without the right responsibility in EBS to view the pages, will have access to all pages. In APEX, it is possible to create authorization schemes from PL/SQL, which enables us to check for an EBS function on every page for each user. With this functionality, it is possible to restrict the user to certain pages in APEX. Just like it is described in the requirements, every APEX function should be connected to an EBS function. This authorization can be maintained directly in the EBS.

To access the EBS, APIs and tables, the right session context is needed. In an EBS session, this happens automatically. In APEX, it needs to be set manually on every page to guarantee a correct functionality. To do so, we can use the information from the token. By using the user ID, responsibility ID and organization ID, it is possible to initialize the EBS and the organization context.

By using our approach, every requirement is met and the EBS can be integrated securely with APEX.

### **Resulting advantages**

With APEX on top of Oracle EBS, we are provided with a framework wherein the extension of the functionality is easily learnable for Oracle Forms developers.

The APEX application can be accessed from a browser, which means that no additional tools, most notably no Java or Java Web Start, are necessary for the use of APEX with the EBS. This advantage is of high importance, since all browsers provided exclude Java from their core functionality.

Since APEX contains a number of strong reporting features such as JET charts, interactive reports and grids, we can easily review the data and print it in different formats such as .pdf or .csv. The reports are even customizable, and by using additional tools such as BI Publisher, the printing of .doc files or .xls files is no problem.

Furthermore, with the responsive design of APEX, all applications are suitable for mobile devices. This increases the availability and accessibility of the application.

Due to the large and helpful community and despite there not being an official Oracle support, problems arising during the development are usually resolved quickly.

## Business Case

A suitable business case for the extension of the Oracle E-Business Suite with APEX can be the creation of a monitoring application for the interface tables. Everybody may be familiar with the situation - how rarely the interface log tables are checked, and the many errors that occur if there are scheduled jobs to import for example items. In the EBS, a form helps to check those interface errors. To use this form, you need in-depth knowledge about the interface, and changing the contained values in it can raise issues.

With the APEX application, the user is given the possibility to, on one hand, monitor the interface, and, on the other hand, to rapidly modify the input data in case of an error. In addition, the user diagram can cluster the errors and give an overview of the interface status. This way of application can be created e.g. in the item interface table, to display the errors and edit or create new interface lines.



Figure 5: Example chart, showing the error per month

Interface line attribute1	Error message	Sales order	Header attribute13	Interface line attribute6	Trx number	Batch source name	Gl date
1003881152	Wenn die Zahlungsart "Automatisch" ist, müssen Sie ein Bankkonto eingeben oder sicherstellen, dass ein primäres Konto für die Transaktionswährung für diesen Rechnungskunden eingerichtet wurde	1003881152	117513224	168500671	117513224	DE_INVOICE_IMPORTED	02-APR-18
1003881152	Wenn die Zahlungsart "Automatisch" ist, müssen Sie ein Bankkonto eingeben oder sicherstellen, dass ein primäres Konto für die Transaktionswährung für diesen Rechnungskunden eingerichtet wurde	1003881152	117513224	168500672	117513224	DE_INVOICE_IMPORTED	02-APR-18
1003881152	Wenn die Zahlungsart "Automatisch" ist, müssen Sie ein Bankkonto eingeben oder sicherstellen, dass ein primäres Konto für die Transaktionswährung für diesen Rechnungskunden eingerichtet wurde	1003881152	117513224	168500670	117513224	DE_INVOICE_IMPORTED	02-APR-18
1003975576	Rechnungs-Nr. bereits vorhanden	1003975576	117580156	169387096	117580156	DE_INVOICE_IMPORTED	14-APR-18
1003975576	Rechnungs-Nr. bereits vorhanden	1003975576	117580156	169387094	117580156	DE_INVOICE_IMPORTED	14-APR-18
1003975576	Rechnungs-Nr. bereits vorhanden	1003975576	117580156	169387098	117580156	DE_INVOICE_IMPORTED	14-APR-18
1003975576	Rechnungs-Nr. bereits vorhanden	1003975576	117580156	169387097	117580156	DE_INVOICE_IMPORTED	14-APR-18
1003612828	Angegebenes Buchungsdatum ist in geschlossener Periode	1003612828	117369762	166293288	117369762	DE_INVOICE_IMPORTED	14-FEB-18
1003612828	Angegebenes Buchungsdatum ist in geschlossener Periode	1003612828	117369762	166293289	117369762	DE_INVOICE_IMPORTED	14-FEB-18
1003612828	Angegebenes Buchungsdatum ist in geschlossener Periode	1003612828	117369762	166293290	117369762	DE_INVOICE_IMPORTED	14-FEB-18
1003612828	Angegebenes Buchungsdatum ist in geschlossener Periode	1003612828	117369762	166293291	117369762	DE_INVOICE_IMPORTED	14-FEB-18
1003612828	Angegebenes Buchungsdatum ist in geschlossener Periode	1003612828	117369762	166293292	117369762	DE_INVOICE_IMPORTED	14-FEB-18
1003612828	Angegebenes Buchungsdatum ist in geschlossener Periode	1003612828	117369762	166293293	117369762	DE_INVOICE_IMPORTED	14-FEB-18
1003612828	Angegebenes Buchungsdatum ist in geschlossener Periode	1003612828	117369762	166293295	117369762	DE_INVOICE_IMPORTED	14-FEB-18

Figure 6: Interactive grid to show detail information and possibility change the errored lines

**Kontaktadresse:**

Yves Chassein

PROMATIS software GmbH

Pforzheimer Straße 160

D-76272 Ettlingen

Telefon: +49 7243 2179 0

Fax: +49 7243 2179 999

E-Mail: [yves.chassein@promatis.de](mailto:yves.chassein@promatis.de)

Internet: [www.promatis.de](http://www.promatis.de)