

ApexLib

The Open Source Oracle APEX Development Framework

White Paper

Patrick Wolf, Sphinx IT Consulting

ApexLib

Executive Overview	3
Introduction	3
Benefits	4
Improves development speed	4
Adds Web 2.0 features to the user interface	4
Avoids JavaScript programming	4
Reduces code to be maintained	4
Documents your code	5
Proven and tested code	5
Internationalized	5
Integrated into your normal Oracle APEX IDE	5
Full access to the source code	5
Architecture Overview	6
Framework Features	7
Browser/Client improvements	7
Out of the box client side validation	7
Generic AJAX enabled cascading LOV	7
Inline Error Display of Tabular Form and Page Process errors	8
Show Validation errors without page re-rendering	8
Declarative Min-/Max Value Validation	8
Declarative Input Length restriction for Tabular Form Columns	8
Declarative “Case Restriction” for Page Items/Tabular Form Columns	9
Resizable Textarea	9
Non navigable Date Picker-/Lov/Image Button Icons	9
Tabular Form record navigation with Up/Down Cursor Keys	9
Hide “Select All” checkbox of row selector	9
Server improvements	9
Automatic security checks for invalid data	9
Plug & Play Tabular Form Handling	9
Display Null = “Yes” and the %null% problem	10
Documentation	10
Conclusion	11

ApexLib

EXECUTIVE OVERVIEW

Oracle Application Express (APEX), a feature of the Oracle database 10g/11g, combines rapid web application development with the power of the Oracle database. Its easy to use declarative browser based application builder enables developers and non-programmers to develop and deploy data driven web applications in very little time. But as with any tool, there are some areas which can be enhanced to even further speed up the development process of an Oracle APEX application and to make the life of a developer easier. That's where ApexLib, the first Oracle APEX development framework comes into the game. Developed under an open source license and hosted on Sourceforge.net, every Oracle APEX developer can use and integrate it into new or existing applications for free.

INTRODUCTION

Oracle APEX is used by ever more Oracle shops and companies to develop database centric web applications. The growing interest in the tool has also raised questions on how to further improve the already very productive development environment and enhance the application with features used in WEB 2.0 web applications.

This paper describes the benefits, architecture and features of the ApexLib Framework and how it can help you to quickly develop WEB 2.0 enabled web applications.

BENEFITS

Many developers are wasting valuable time by re-inventing the wheel or by re-implementing for each occurrence the same solution patterns (for example AJAX based cascading LOV (List of Values)) again and again in their applications. The ApexLib Framework implements and provides generalised solutions for a lot of the day to day problems of an Oracle APEX developer. It also helps to hide the complexity of JavaScript programming from “normal” developers who probably just have SQL and PL/SQL experience.

Improves development speed

The Framework offers a lot of solutions for the small but sometimes time consuming problems which arise during the development of an Oracle APEX application, especially if end users are demanding enhancements to the user interface which are not offered out of the box by Oracle APEX. Security and input data checking are also a big issue for web applications, which consume a lot of time during development. The Framework does a lot of these checks automatically based on the applications meta data stored in the Oracle APEX repository, helping the developer to concentrate on the business logic and not on technical checks. With its seamless integration into the Oracle APEX IDE and the runtime environment, the ApexLib Framework feels like as an integrated part of the Oracle APEX development environment.

Adds Web 2.0 features to the user interface

State of the art web applications have to have so called Web 2.0 features such as

- cascading LOVs which use AJAX to avoid a full page refresh
- validation in the browser to give immediate feedback when a user leaves a field
- resizable text areas
- keyboard navigation and short cuts for enhanced usability for experienced end users
- tabular form improvements
- enhanced error handling/display for tabular forms

Avoids JavaScript programming

Oracle APEX developers often have good SQL and PL/SQL skills, especially if they have an Oracle Forms background. But only a few have Javascript skills which are required for enhanced browser programming, for example to use AJAX in an application. The problem with Javascript programming is that it can be very time consuming to make it work in all the common browsers (Internet Explorer 6/7, Firefox, Safari). That’s why the framework includes a JavaScript library which works with all the browsers and offers a PL/SQL wrapper for a lot of the functions so that the developer can use PL/SQL to do the programming.

Reduces code to be maintained

Because the framework does a lot of “technical” checks, for example,

- Is the item/column required based on the assigned label template?
- Does the entered value match with the date- or numeric format mask?
- Is the value within the defined min-/max boundary?
- Does the length of the entered text in a tabular form column match with the database column length?

The developer doesn’t have to define “Validation Processes” for all of these checks to make the application secure against invalid data. If you don’t have to write it, you do not have to maintain it.

The same applies to the AJAX enabled cascading LOVs. Normally you have to write an “On-Demand Process”, a JavaScript function and code in your page to call these functions, but

with the framework you don't write a single line. It detects the page item dependencies based on the application metadata and injects the necessary code on the fly at runtime. Again, nothing to maintain, it's just there and the developer doesn't have to worry about it.

Documents your code

One of the features of the framework is a "Page Flow Diagram Generator", which generates based on the data in the Oracle APEX repository several "Page Flow Diagrams" of your application. It's all there, somebody just had to analyze and use it. The big advantage of this tool is that you don't have to draw these yourself, which is time consuming. They are also out of date and inaccurate as soon as a change has been made in the application. These diagrams help new developers to understand your application.

Proven and tested code

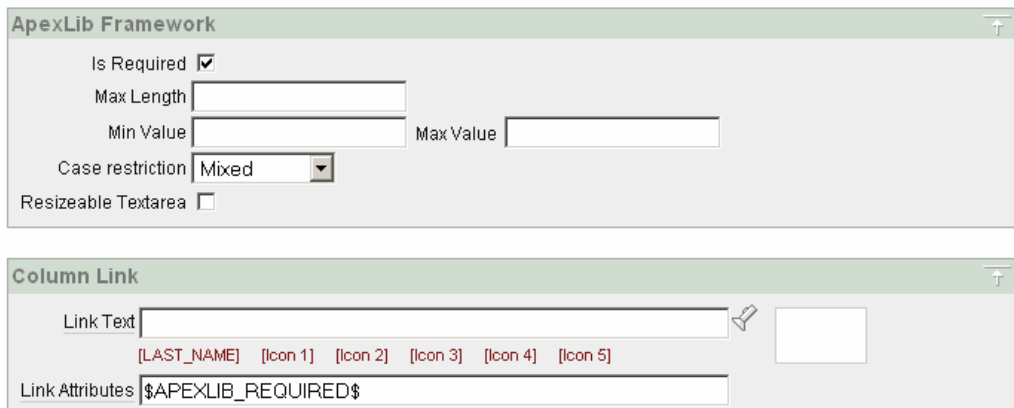
The last version (1.6) of the framework has been downloaded more than 1.600 times (as of end of August 2007) so far. Based on the user feedback it's already used in a lot of applications which work successfully in production. Self written code which is just used in one application can't have the same quality and have undergone the same testing as the framework which has been already been tested by many developers and users.

Internationalized

The framework was developed with NLS in mind. All messages produced by the framework can be translated into other languages. It also considers the NLS settings of the database session when it performed date and number format checks. The framework is already used successfully in production applications around the world.

Integrated into your normal Oracle APEX IDE

By using the Oracle APEX Builder Plugin (another open source project developed by us), the features and switches for the ApexLib framework can be seamlessly intergraded into your normal Oracle APEX Builder IDE. This makes it even easier and faster for the developer to use the features.



ApexLib Framework

Is Required

Max Length

Min Value Max Value

Case restriction

Resizable Textarea

Column Link

Link Text

Link Attributes

Full access to the source code

Because the ApexLib Framework is published under an open source license, developers have full access to the well documented source code enabling them to enhance or fix the Framework themselves.

ARCHITECTURE OVERVIEW

The framework consists of a set of PL/SQL packages, views and JavaScript libraries which can be installed into the applications database schema or into a separate ApexLib schema, if the framework is used in several applications.

The views are used to add an abstraction layer on top of the Oracle APEX repository views. This allows the support of different Oracle APEX versions with the same framework code base. Currently Oracle APEX 2.0, 2.1 (XE), 2.2 and 3.0 are supported. There have also been reports by users that they can even use it with HTMLDB 1.6.

The framework uses regular hooks provided by the Oracle APEX runtime engine to integrate into an application. All the necessary steps to integrate the framework into an application are documented in the How-To Integrate document which comes with the framework distribution. Completing these steps takes about 10 min.

ApexLib's own metadata are stored in the comment properties of the different Oracle APEX objects and are based on simple tags like \$APEXLIB_REQUIRED\$, ... The advantage of using the Oracle APEX repository to store this information instead of creating own tables is that

- Export/Import/Backup is done by Oracle APEX.
- No synchronization/version problems, it's always in sync with the Oracle APEX meta data.
- No extra tables required.

When an application is executed and a page is rendered, the Oracle APEX runtime engine calls the ApexLib framework entry points and based on the application- and ApexLib metadata stored in the Oracle APEX repository, the framework functions fire additional checks or inject additional code to provide a richer user interface experience for the end user.

FRAMEWORK FEATURES

At <http://apexlib.sourceforge.net/> you can get the most up-to-date list of the features included in the framework.

The Feature demonstration web-site which is hosted on <http://apex.oracle.com/> shows all the described features and is the best way to find out more about how they work. You can find it at <http://apex.oracle.com/pls/otn/f?p=33231:40> The same demo application is also included in the framework distribution, so that you can check out how the features are used.

Browser/Client improvements

Features which improve the appearance of your application in the browser.

Out of the box client side validation

By using the ApexLib JavaScript library, the framework automatically adds checks for required, date picker- and numeric items based on the application meta data. These checks are useful, because it avoids an unnecessary page rendering roundtrip and the user gets immediate feedback about data entry errors.

The checks are not just simple checks where an error is displayed in an alert window, instead it shows the error message as inline error next to the field by manipulating the DOM tree of the HTML page. The goal was to have the same error message appearance as if Oracle APEX would generate it when it performs server side checks.

» <http://inside-apex.blogspot.com/2007/01/one-small-step-for-man-one-giant-leap.html>

Generic AJAX enabled cascading LOV

Sometimes there are items on a page which depend on each other. For example, the first item is a select list which shows car manufacturers and the second select list should show just the models of the selected car manufacturer.

In the Web 1.0 world a page re-rendering was necessary as soon as the user had selected a manufacturer from the first select list, because the server had to generate a new page which contains just the models of this manufacturer in the second select list.

In the Web 2.0 world this is no longer state of the art. Instead of re-rendering the whole page, an AJAX call asks the server for just the models. The model select list is changed on the fly by using browser DOM tree manipulation. This technique avoids page flickering and additional programming in the browser.

To use this feature, no additional coding has to be done. The framework automatically analyzes the WHERE clause of the LOV SQL Statements to determine the dependencies between the items and inject the necessary code.

» <http://inside-apex.blogspot.com/2006/11/generic-solution-for-depending-select.html>

» <http://inside-apex.blogspot.com/2006/12/speedup-of-cascading-ajax-lovs.html>

Inline Error Display of Tabular Form and Page Process errors

Tabular Form handling in Oracle APEX is a little bit weak. Errors can just be displayed on a separate error page, because otherwise the entered data is lost. But users want to see an error message with their entered data and not on a separate page without the context needed to fix the error.



This feature tweaks the Oracle APEX error handling so that tabular form errors are displayed as inline messages next to the tabular form. It even highlights the cell with the error.

Email	Hire Date	Job
test	26-10-2097	S
JSEO	12-12-1998	S

- » <http://inside-apex.blogspot.com/2006/12/hacking-apex-error-page.html>
- » <http://inside-apex.blogspot.com/2006/12/hacking-apex-error-page-part-2.html>

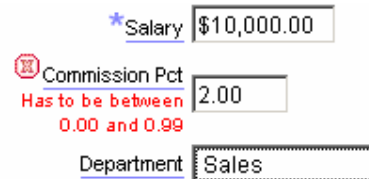
Show Validation errors without page re-rendering

There are sometimes situations where users enter data which you have to validate on the server and which triggers a re-rendering of the page to show the error. But because of the nature of the data (like password fields) you can't include the value the user just entered into the generated HTML page. So the user has to enter the value again. This feature provides a solution for this problem without losing the entered data.

- » <http://inside-apex.blogspot.com/2007/01/apexlib-new-release-v016.html>

Declarative Min-/Max Value Validation

A common requirement during application development is to restrict the data which can be entered into a field. The most common restriction is to have, for example, a minimum and/or maximum value for a number field or to define that a date value can't be in the past.



*Salary \$10,000.00
 Commission Pct 2.00
 Has to be between 0.00 and 0.99
 Department Sales

Oracle APEX doesn't provide a min-/max value property out of the box to do this basic checks, instead you have to create for each case a "Validation Process" where you have to define the check expression, an error message, ...

When defined, the ApexLib framework does the check on the client side and for security reasons also on the server side, so that no invalid data gets into your database.

- » <http://inside-apex.blogspot.com/2007/03/min-max-value-validation-in-browser.html>

Declarative Input Length restriction for Tabular Form Columns

Oracle APEX currently doesn't generate the HTML attribute MAXLENGTH for Tabular Form Columns, instead all are defined as maxlength=2000. The result is that the database raises an ORA-12099: value too large for column ... when the user tries to save a value that's too long

Because the error message and the behaviour isn't very user friendly, the developer has to do a lot of coding to avoid this error, especially because validating Tabular Form Columns is a little bit tricky in Oracle APEX. By using this feature, the framework will add the necessary HTML attributes to restrict the length as the user enters data into the field.

- » <http://inside-apex.blogspot.com/2007/02/restrict-input-length-for-tabular-form.html>

Declarative “Case Restriction” for Page Items/Tabular Form Columns

Adds case restriction (uppercase/lowercase/capitalize) to a text field as done in Oracle Forms with the “Case Restriction” property.

» <http://inside-apex.blogspot.com/2007/02/setting-case-restriction-for-page.html>

Resizable Textarea

Do you have a lot of text in a text area, but don’t want to waste too much space on the page for it? Then this feature is exactly what your users are looking for. It allows them to resize the text area item in the browser to view all the text at once.

» <http://inside-apex.blogspot.com/2007/02/resizeable-textarea.html>

Non navigable Date Picker-/Lov/Image Button Icons

One drawback of web applications is that most of the time just using the keyboard during data entry is not always user friendly, as there are a lot of objects on a web page which can get focus when you tab through a page. Like the date picker-, LOV-icons or some button images. This feature prevents these icons getting cursor focus so it’s easier to tab through a data entry form.

» <http://inside-apex.blogspot.com/2007/02/non-navigable-date-picker-lovimage.html>

Tabular Form record navigation with Up/Down Cursor Keys

From Client/Server or rich web UI like Oracle Forms users are used to use cursor up and down to scroll through the records in a tabular form. This enhanced keyboard usability can be added with this feature.

» <http://inside-apex.blogspot.com/2007/02/tabular-form-record-navigation-with.html>

Hide “Select All” checkbox of row selector

An easy way to hide the “Select All” checkbox of updateable Tabular Forms.

	Employee Id	First
<input type="checkbox"/>	207	kkkkkkkk
<input type="checkbox"/>	210	Helmut

» <http://inside-apex.blogspot.com/2007/02/hide-select-all-checkbox-of-row.html>

Server improvements

Features which make it easier for a developer to access Oracle APEX functionality on the server.

Automatic security checks for invalid data

As already described for the “Out of the box client side validation”, the ApexLib Framework does the same checks on the server when the page is submitted, to make sure that only valid values are processed by the application. Adding these “technical” checks is normally a lot of work for a developer, because he probably has to create several “Validation Processes” for each page item/tabular form. It also clutters the “Validation Processes” with basic technical checks instead just showing business relevant validations. This feature helps to make the application more readable and saves a lot of time for the developer.

» <http://inside-apex.blogspot.com/2007/01/apexlib-automatic-required-and-format.html>

Plug & Play Tabular Form Handling

Tabular Form handling in Oracle APEX isn’t one of his strong features. It’s not been possible to define declarative validation processes for Tabular Form Columns until now. Instead the developer has to use some hard to use PL/SQL arrays to access the data and to do the validation. The code is hard to read and to maintain. With the Plug & Play Tabular Form APIs, the framework adds PL/SQL API wrappers around the Oracle APEX functionality to

make the handling much easier and developer friendly. This helps to leverage the possibilities of the Tabular Forms which are often not used because of their complexity.

» <http://inside-apex.blogspot.com/2006/12/plugin-play-tabular-form-handling.html>

Display Null = “Yes” and the %null% problem

If a LOV allows a NULL selection, Oracle APEX stores %null% as value which can cause some problems in validation- or submit processes. This feature avoids this problems by setting the value to NULL in such cases.

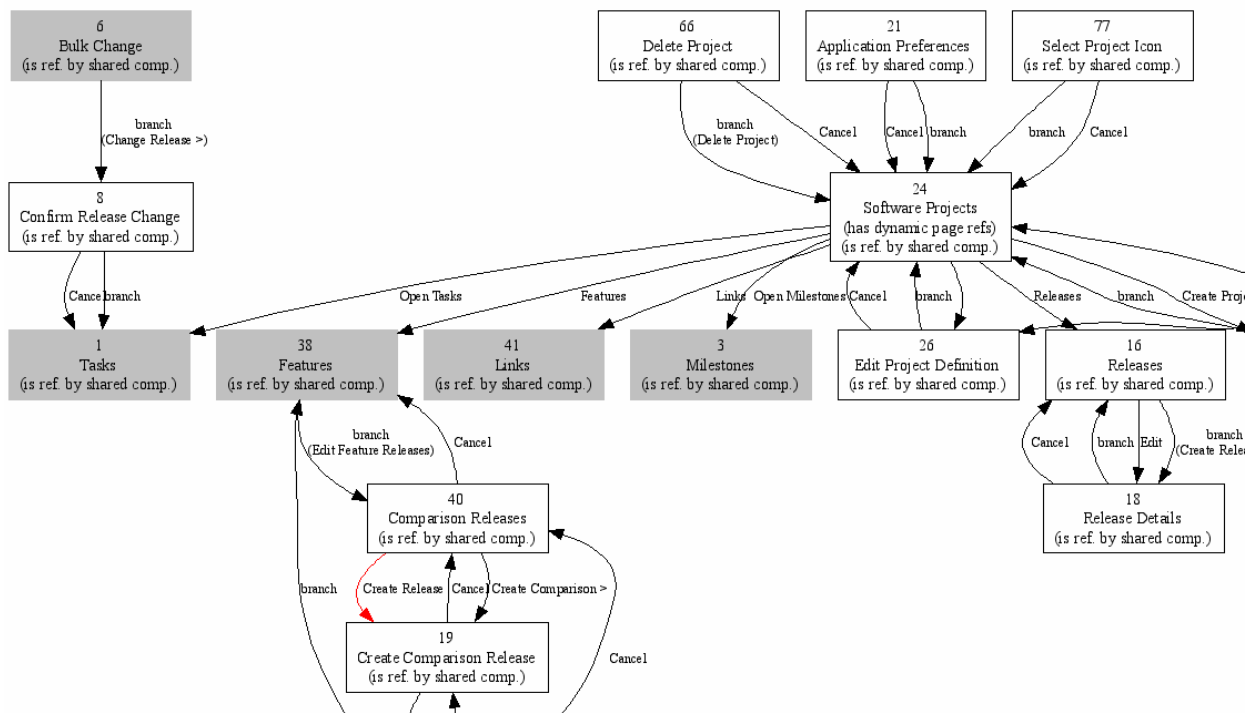
» <http://inside-apex.blogspot.com/2006/12/display-null-yes-and-null-problem.html>

Documentation

With the help of the “Page Flow Diagram Generator” which uses the open source Graphviz tool, the ApexLib framework is able to generate an up-to-date version of the page flow of an application for free and without additional effort.

These diagrams help to better understand how the pages are linked together and which page calls which page. Sometimes a picture says more than a 1000 words. Especially for developers which are new to an application or if you have to change the application a long time after the initial development, the diagrams gives you a good overview of the dependencies within the application.

» <http://inside-apex.blogspot.com/2007/01/apexlib-get-page-flow-diagram-of-your.html>



CONCLUSION

The ApexLib Framework helps to further increase the productivity of the Oracle APEX application development by helping developers to concentrate on the business and not technical problems.

For the application end user it adds a lot of usability enhancements which help to increase their satisfaction with the application.