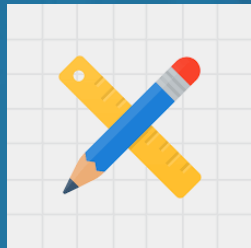


ORACLE®

# Extend Your APEX Application with Oracle JET



**Hilary Farrell**

**Consulting Member of Technical Staff**

Oracle Application Express, Database Division

November 20, 2018

# Safe Harbor Statement

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 remains at the sole discretion of Oracle.

# Hilary Farrell

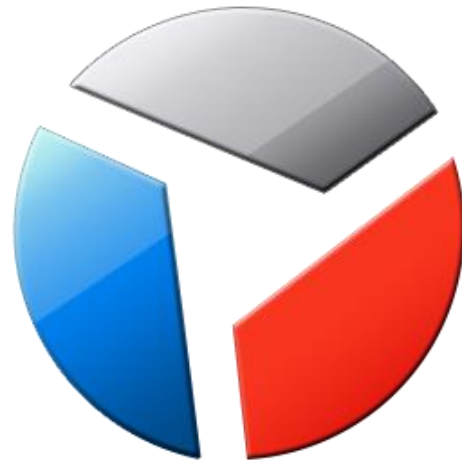
## Oracle Application Express

- Consulting Member of Technical Staff
- Based in Dublin, Ireland
- Joined Oracle in 1998
- Joined APEX Team in 2006
- Work includes Charts Integration, Modal Dialogs, RESTful Services integration with ORDS, Create Wizards, Packaged App Installations, Forms Conversion utilities; Release Management responsibilities; Hudson Build
- [Hilary Farrell-Oracle](#) on OTN Forum

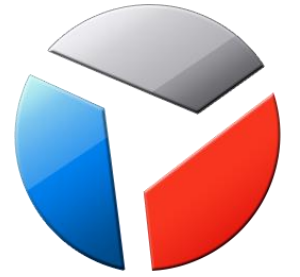


# Agenda

- 1 Oracle JET
- 2 Integration Guidelines
- 3 Demo



Oracle JET



# Oracle JET

## Brief Intro

- Oracle JavaScript Extension Toolkit (JET)
- JavaScript-based solution
- Highly Customizable
- Built-in Accessibility Support
- Built-in Mobile Support
- Wide Range of UI Components
  - **Collections**, Controls, **Forms**, Framework, Layout & Nav, Patterns, **Visualizations**

# Oracle JET in APEX

## JET Versions

- APEX 5.1.x
  - JET 2.0.2
    - jQuery 2.1.3
    - jQuery UI 1.11.4
  - New Charting Solution: based on JET Data Visualizations
    - Bar, Bubble, Combination, Line, Scatter, Stock
    - Dial Gauge, Donut, Pie
    - Area, Line with Area, Funnel, Polar, Radar





# Oracle JET in APEX

## JET Versions



- APEX 18.x
  - JET 4.2.0
    - jQuery 3.1.1
    - jQuery UI 1.12.0
  - New “Text Field with Autocomplete” Item Type:- based on JET Input Search
  - New Chart Types:- Box Plot, Gantt, Pyramid
  - New Common Bundle:- `/images/libraries/apex/minified/jetCommonBundle.min.js`
    - ‘ojs/ojcore’, ‘ojs/ojcomponentcore’, ‘ojs/ojmessaging’, ‘ojs/ojvalidation-base’, ‘ojs/ojvalidation’

# Oracle JET in APEX

## JET Versions

- APEX 19.x
  - JET 6.x.x => currently 6.0.0
    - jQuery 3.3.1
    - jQuery UI 1.12.1
  - Growing declarative support: new chart type & other components planned
- Future Releases of APEX
  - Simplify integration process; aim to make integration of JET components more declarative!



# Integration Guidelines

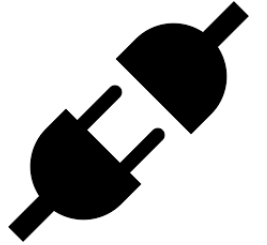
# Integration Guidelines

- 1 Create an APEX Plug-in
- 2 Things to Do
- 3 Things to Avoid



# Create an APEX Plug-in

# Create an APEX Plug-in



- Reusable component
  - Encapsulate the JET component within an APEX plug-in, utilize JET Cookbook & APIs
  - Behaves like native component; available for use/install in any app in your workspace
- Easily extendable & customizable
  - Expose JET attribute as declarative attribute of APEX plug-in
  - Utilize *'Has JavaScript Initialization Code Attribute'* option to allow plug-in user to extend the plug-in capabilities
- Use of APEX APIs
  - Use documented APIs, such as APEX\_PLUGIN, APEX\_JSON, in plug-in rendering logic



# Things to Do

# Things to Do

- Use JET version shipped with APEX
  - Avoid risk of jQuery Library conflicts by introducing newer version of JET libraries
  - Consider supportability & backward compatibility
- Use APEX Substitution Strings
  - **#JET\_BASE\_DIRECTORY#** - i.e /images/libraries/oraclejet/4.2.0/
  - **#JET\_CSS\_DIRECTORY#** - i.e. #JET\_BASE\_DIRECTORY#css/libs/oj/v4.2.0/
  - **#JET\_JS\_DIRECTORY#** - i.e. #JET\_BASE\_DIRECTORY#js/libs/oj/v4.2.0/
- Use RequireJS
  - Many JET modules have dependency on RequireJS so advisable to use it



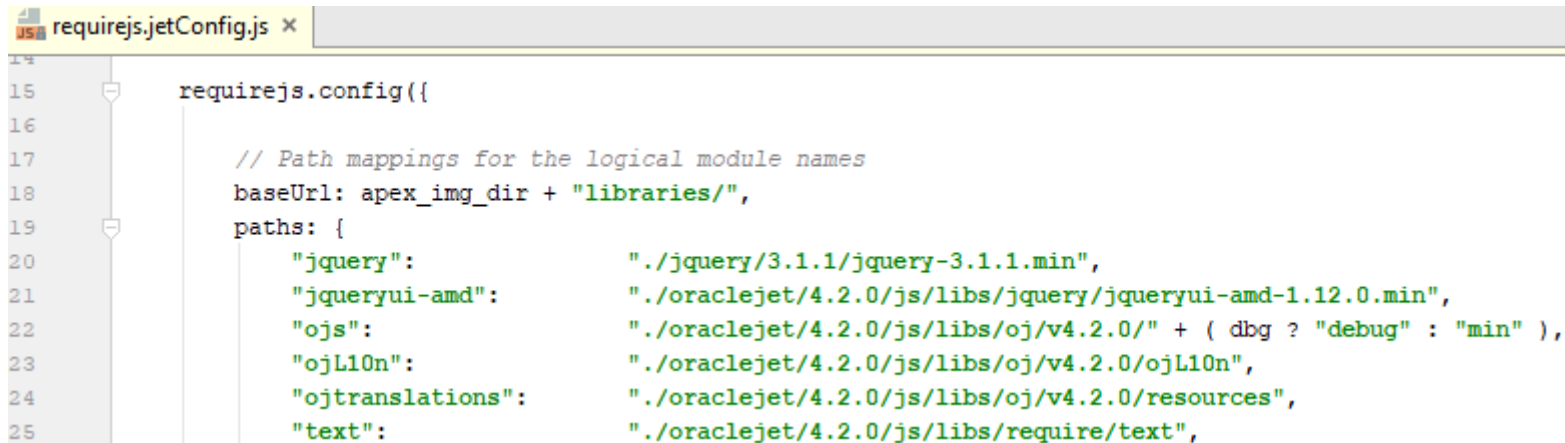
# Things to Do

- Use URL Prefix for Plug-in JS file
  - **[require jet]** e.g. `[require jet]#PLUGIN_FILES#myPluginJSFile.js`

JavaScript ?

```
[require jet]#PLUGIN_FILES#jetLegend.js
```

- Automatically utilizes `/images/libraries/apex/requirejs.jetConfig.js`



```
requirejs.config({  
  
    // Path mappings for the logical module names  
    baseUrl: apex_img_dir + "libraries/",  
    paths: {  
        "jquery":                "./jquery/3.1.1/jquery-3.1.1.min",  
        "jqueryui-amd":          "./oraclejet/4.2.0/js/libs/jquery/jqueryui-amd-1.12.0.min",  
        "ojs":                    "./oraclejet/4.2.0/js/libs/oj/v4.2.0/" + ( dbg ? "debug" : "min" ),  
        "ojL10n":                 "./oraclejet/4.2.0/js/libs/oj/v4.2.0/ojL10n",  
        "ojtranslations":        "./oraclejet/4.2.0/js/libs/oj/v4.2.0/resources",  
        "text":                   "./oraclejet/4.2.0/js/libs/require/text",  
    }  
});
```

# Things to Do

- Choose JET Plug-in Style
  - Widget
    - `$( "selector" ).widget( "option", "property", value );`
    - Non declarative options set with JavaScript initialization Code
    - See Legend Plug-in example in Sample Charts app
  - Web Component / Custom Element Style
    - Custom HTML Element e.g. `<oj-combobox />`
    - `myComponent.property = value;`
    - Add attribute `data-oj-binding-provider="none"`
    - Non declarative attributes set with Custom Attributes

# Things to Do

- Utilise APEX APIs & Plug-in Attributes
  - Items: create an item interface using JS API [apex.item.create](#)
  - Regions: create a region interface using JS API [apex.region.create](#)
  - Make use of standard plug-in attribute *Has “Initialization JavaScript Code” Attribute*
- Test Plug-in Behaviour After Every APEX Upgrade
  - Ensure Plug-in behaves as expected with each APEX/JET upgrade
  - JET API changes may require plug-in code changes
  - Review APEX Release Notes
    - A new declarative support may negate the need for plug-in
    - New features or APIs may be available, to help improve plug-in

# Things to Do

- Consider using our CDN – for speedier file handling performance
  - Not specific to plug-in development in APEX
  - See Joel’s announcement blog:

<https://blogs.oracle.com/apex/announcing-oracle-apex-static-resources-on-oracle-content-delivery-network>

**APEX 18.2** <https://static.oracle.com/cdn/apex/18.2.0.00.12/>

**APEX 18.1** <https://static.oracle.com/cdn/apex/18.1.0.00.45/>

# Things to Avoid



# Things to Avoid

- Use of Knockout
  - Over-complicates plug-in JS code
  - More overhead for future maintenance
- Use of newer version of JET
  - Not tested & may result in 3<sup>rd</sup> party library conflicts, specifically with jQuery
- Integration of components already declaratively available in APEX
  - Why introduce this additional overhead to your app?
  - Leverage our declarative support – extend only where necessary

# Things to Avoid

- Hard-Coded References to JET version
  - Issues will arise with upgrades to future versions of APEX ( & JET )
  - Consider supportability & backward compatibility
- Hard-Coded Attributes in Plug-in Code
  - Should be reusable code; avoid hard-coding source or attributes in plug-in
  - Make as flexible as a native component

# Demo



# Additional Resources

## APEX



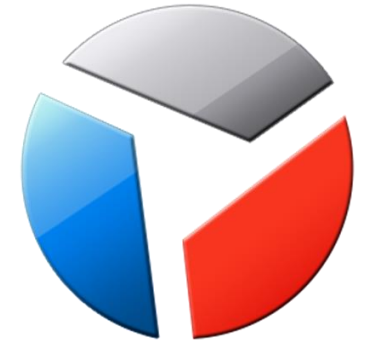
**APEX 18.2 Sample Charts App** – demo ojLegend plug-in  
<https://apex.oracle.com/charts/>

**APEX User's Guide - Chapter 19.2 Implementing Plug-ins** -  
<https://docs.oracle.com/database/apex-18.2/HTMDB/implementing-plugins.htm#HTMDB27001>

**APEX World Plug-In Repository** -  
<https://apex.world/ords/f?p=100:700:::NO:700::&cs=3ONN3042ZA4uWRWfMuhDZBcZCriA>

# Additional Resources

## JET



Oracle JET -

<http://oraclejet.org/>

JET 4.2.0 ojChart API -

<https://docs.oracle.com/en/middleware/jet/4.2.0/reference-api/oj.ojChart.html>



JET 4.2.0 Cookbook -

<https://www.oracle.com/webfolder/technetwork/jet-420/jetCookbook.html>



JET OTN Forum -

[https://community.oracle.com/community/groundbreakers/development\\_tools/oracle-jet](https://community.oracle.com/community/groundbreakers/development_tools/oracle-jet)



# Integrated Cloud

## Applications & Platform Services

ORACLE®