18th Int. Conf. on Acc. and Large Exp. Physics Control Systems ICALEPCS2021, Shanghai, China JACoW Publishing
ISBN: 978-3-95450-221-9 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2021-MOPV047

UPGRADING ORACLE APEX APPLICATIONS AT THE NATIONAL IGNITION FACILITY

A. Bhasker, R. D. Clark, R.N. Fallejo
Lawrence Livermore National Lab, CA, USA

Abstract

As with all experimental physics facilities, NIF has software applications that must persist on a multi-decade timescale. They must be kept up to date for viability and sustainability. We present the steps and challenges involved in a major Oracle APEX application upgrade project from Oracle APEX version 5 to Oracle APEX version 19.2. This upgrade involved jumping over 2 major versions and a total of 5 releases of Oracle APEX. Some applications - that depended on now legacy Oracle APEX constructs required rearchitecting, while others that broke due to custom JavaScript needed to be updated with compatible code. This upgrade project, undertaken by the NIF Shot Data Systems (SDS) team at Lawrence Livermore National Laboratory (LLNL), involved reverse-engineering functional requirements for applications that were then redesigned using the latest APEX out-of-the-box functionality, as well as identifying changes made in the newer Oracle APEX built-in "plumbing" to update custom-built functionality for compatibility with the newer Oracle APEX version. As NIF enters into its second decade of operations, this upgrade allows for these aging Oracle APEX applications to function in a more sustainable way, while enhancing user experience with a more modernized GUI for existing and future Oracle APEX webpages.

INTRODUCTION

The National Ignition Facility (NIF) made history when it first fired all 192 of its laser beams at a single point inside the vast, spherical target chamber in 2009. Now, over 10 years and more than 2,700 shots later, NIF still dominates the world of high-energy lasers by a factor of 10 and continues to help the Laboratory achieve its mission objectives, offering unparalleled laser performance and precision [1]. As NIF enters into its second decade of operations, it must continue to maintain its various software applications in a sustainable and viable way. Two such applications that the Software Data Systems (SDS) team at NIF recently upgraded are the *Production Optics Reporting and Tracking* and the *Shot Planner* applications. At the time of this undertaking, these applications were running on Oracle APEX version 5.0. Oracle APEX is a web-application development tool for the Oracle databases [2], that supports utilization of JavaScript and JQuery components as well. In this paper, we discuss the steps involved in upgrading these NIF applications from Oracle APEX version 5.0 to Oracle APEX version 19.2 – a jump spanning 2 major versions and a total of 5 releases of Oracle APEX.

The Oracle APEX Applications

The original *Production Optics Reporting and Tracking (PORT)* and the *Shot Planner* applications were developed in very early versions of Oracle APEX when NIF was in the process of being built, over a decade ago. The PORT application covers a wide domain of capabilities – ranging from reporting and analysis to task-scheduling and optics installation. The Shot Planner application functions as a scheduling app, providing reports and forms to plan the sequencing of shots on NIF. At the time of the Oracle APEX upgrade, both these applications were running on legacy Oracle APEX themes and utilizing certain Oracle APEX constructs (such as AnyChart/Flash Charts [3]) that are no longer supported by the latest Oracle APEX versions.

METHODOLOGY

The PORT application comprised of ~500 application pages, and the Shot Planner application comprised of ~200 application pages at the time of this upgrade. The following are the steps that the team undertook as a part of the Oracle APEX upgrade for these applications (Figure 1).

1. Identify all Obsolete Application Pages
2. Set up an APEX 19.2 environment for testing
3. Fix Forward Incompatibility
4. Deploy Pre-Upgrade Fixes
5. Line Up Post-Upgrade Fixes
6. Oracle APEX Upgrade Utility Recommendations
7. Upgrade and Deploy Post-Upgrade Fixes

Figure 1: Methodology followed.

Identify Obsolete Webpages

The SDS team worked with the various users to identify any application pages that were no longer used. Given the age of these applications, several application functionalities were no longer required as the program evolved with time. Identifying these application pages gave the team an opportunity to prune out the deprecated functionalities. To be sure that a functionality could be deprecated, all visits of the various application pages on these applications were recorded for a period of one year before beginning the actual upgrade. To achieve this, a new Oracle database table was created to record this information from the APEX_WORKSPACE_ACTIVITY_ LOG view [4].

A Sandbox Environment with Oracle APEX 19.2

The next step was setting up a test environment with the target Oracle APEX version 19.2 installed. Copies of the application were imported into this *sandbox* environment. The team meticulously tested all the functionalities on each Oracle APEX application page and identified features that were no longer working in the new upgraded environment. This was done by comparing each application page in the *sandbox* against the actual *production* version of the application page and documenting the results.

Categorize/Fix Forward Incompatibility

The comparison test between the application pages on the sandbox v/s production environments helped identify whether a page could be upgraded to Oracle APEX version 19.2 as-is, or if some additional work needed to be done to bring it to a functioning state. Broadly speaking, the comparison test resulted in pages falling in one of the following three categories:

- The application page could be upgraded as is, i.e., no additional work required.
- The application page rendered initially but some smaller functionalities no longer worked. Example: List of Values (LoV) now require both an explicit return and display value while they did offer backward compatibility for only defining one value to be used for both display and return.
- The application page did not render at all due to some browser incompatibility. This typically meant a JavaScript or JQuery error.

Fortunately, the majority of the application pages belonged to the first category, where the pages could be upgraded as is. However, there were still many high impact application pages on both the applications that needed to be fixed before the production environment could be upgraded to Oracle APEX 19.2. At this point, *two* important determinations need to be made. First, determining the cause behind the application pages not behaving exactly the same as they did in the old environment. We found that the cause was one of the two (Figure 2):

- The application page had an Oracle APEX feature implemented that was deprecated and no longer compatible with version 19.2. These were typically minor fixes that mostly required small changes in code.
- Or, there was custom JavaScript/JQuery either embedded in the application page itself or in a custom plug-in used by that application page. The fixes for these ran the full gamut, from being small code fixes like updating the *html/css* class names being referred to in JavaScript code, to reverse engineering and redesigning an entire application functionality using the latest Oracle APEX 19.2 constructs like Interactive Grids.

Second, determining whether the fix can be applied in the current production environment version *before* upgrading to the new version (Figure 3). For instance, fixes involving certain unsupported APEX features could be made in the current version. An example would be rewriting the List of Values (LoV) definitions to include both return and display values. An example where the fix could not be applied before upgrading would be an application page using complex custom Javascript that is not compatible with the Javascript versions native to the latest Oracle APEX version and was redesigned using the latest Oracle APEX 19.2 out-of-the-box constructs.

By the end of this step, the sandbox environment had been used to develop mitigations for all types of broken application pages, and the sandbox environment was a functional equivalent of the production environment.

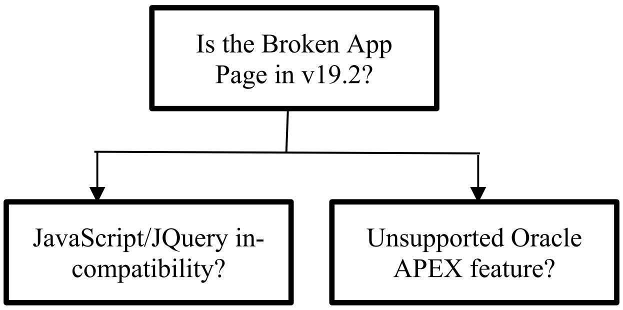

Is the Broken App Page in v19.2?
JavaScript/JQuery incompatibility?
Unsupported Oracle APEX feature?

Figure 2: Page forward incompatibility causes.

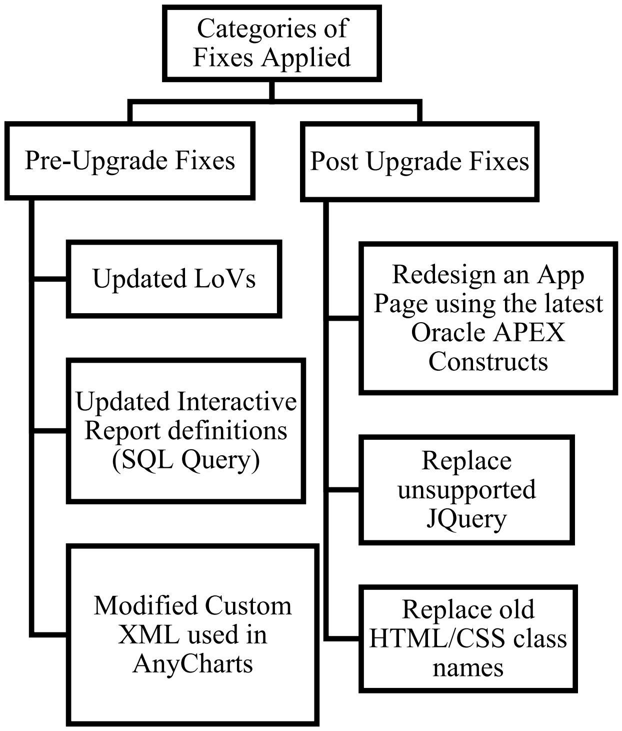

Categories of Fixes Applied
Pre-Upgrade Fixes
Post Upgrade Fixes
Updated LoVs
Updated Interactive Report definitions (SQL Query)
Modified Custom XML used in AnyCharts
Redesign an App Page using the latest Oracle APEX Constructs
Replace unsupported JQuery
Replace old HTML/CSS class names

Figure 3: Examples of fixes for a smooth upgrade.

Apply Pre-Upgrade Fixes

Any identified fix that could be applied to the APEX 5.0 version environment was deployed, and the applications were once again imported to the sandbox environment. A second round of testing confirmed that the specific pages with the pre-applied fixes now worked correctly in the Oracle APEX 19.2 environment.

Export the Post-Upgrade Fixes

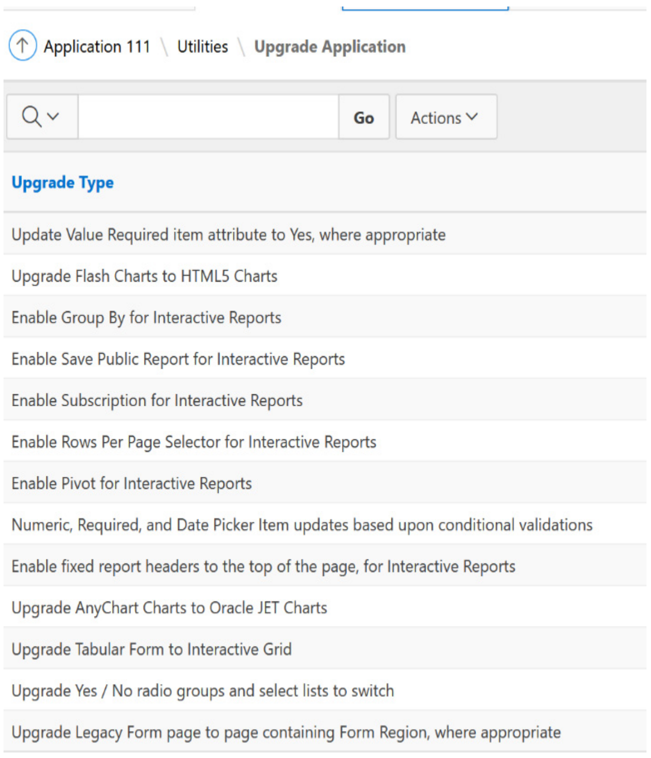
Any re-architected pages using new Oracle APEX constructs like Interactive Grids or Oracle JET charts were exported from the *sandbox* environment and applied as a software release immediately after upgrading the production environment to Oracle APEX 19.2. Another example of a post-upgrade fix includes several pieces of custom JavaScript code and plugins that use the *html/css* class names to add custom features to regular Oracle APEX constructs like Interactive Reports and Classic Reports. For instance, the APEX Interactive Report class is now called 'a-IRR-table' in APEX 19.2; previously this was called 'apexir_WORKSHEET_DATA'.

Upgrade Utility

Oracle APEX has an in-built 'Upgrade Application' Utility which makes recommendations for individual features in pages that can be upgraded. Figure 4 below shows an example of some upgrade categories it auto-suggests. The team also tested the specific application pages listed under these upgrade categories after accepting these upgrades in the sandbox. Some of these upgrades do not result in functional equivalents of the original page. For example, accepting the upgrades to upgrade Tabular Forms (which are still compatible with Oracle APEX 19.2) to Interactive Grids did not result in functioning pages, and required further re-design of the page. Such upgrades were selectively applied during this phase of the upgrade. Other upgrades, such as 'Enable Group By for Interactive Reports' or 'Enable Save Public Reports' were accepted using this utility, as they only enhanced the features on the page.

Upgrade and Deployment

With all the fixes identified, applied and user-tested over the course of several weeks in the upgraded sandbox environment, the upgrade was successfully performed in the production environment, followed by the deployment of the post-release fixes, and then accepting the pre-tested page-wise upgrades from the APEX Upgrade Utility. The entire effort took around 4 months, which including meticulous testing, identifying the causes for various incompatibilities and any redesign efforts.



Upgrade Type
Update Value Required item attribute to Yes, where appropriate
Upgrade Flash Charts to HTML5 Charts
Enable Group By for Interactive Reports
Enable Save Public Report for Interactive Reports
Enable Subscription for Interactive Reports
Enable Rows Per Page Selector for Interactive Reports
Enable Pivot for Interactive Reports
Numeric, Required, and Date Picker Item updates based upon conditional validations
Enable fixed report headers to the top of the page, for Interactive Reports
Upgrade AnyChart Charts to Oracle JET Charts
Upgrade Tabular Form to Interactive Grid
Upgrade Yes / No radio groups and select lists to switch
Upgrade Legacy Form page to page containing Form Region, where appropriate

Figure 4: APEX upgrade utility.

NOTABLE LESSONS LEARNED

Some issues encountered in the upgrade that are worth mentioning:

- There were several cases of JQuery incompatibility when running the same application page on the same browser, but on the two different versions of APEX. Example: replacing "$.browser.msie" with "navigator.appname" got several pages working again.
- Oracle APEX has changed the CSS class names of some of its more popular constructs like Interactive Reports. Legacy custom code as well as plug-ins often utilize these.
- Some functionalities were redesigned using Interactive Grid instead of applying complex custom JavaScript to Interactive Reports to implement some *Form* functionalities in older versions of APEX. It was discovered that while the Interactive Grids in Oracle APEX allowed users to save Private and Public reports (same as Interactive Reports do), these saved reports are not exported as a part of an application page export. This is significant because this may impact the Oracle APEX administrator's ability to maintain individual application backups containing all saved reports. Fortunately, the full application export does in fact contain the saved reports.
- The APEX_PLUGIN.GET_AJAX_IDENTIFIER() function can sometimes include hyphens in the unique AJAX-ID generated in the upgraded Oracle APEX version. Some browsers may have compatibility issues with this and throw console errors.
- Some LoVs had invalid values that could only be viewed in the *Page Component* View in Oracle.
- APEX 5.0. The LoV's values could not be viewed in Oracle APEX 19.2's *Page Designer* view at all. These LoV definitions were updated as part of the pre-upgrade fixes as the *Page Component* view is not available in Oracle APEX 19.2.
- Interactive Reports with SQL query definitions of the format *select * from <object>* needed redefinition with explicit column names in the select clause. This was for niche cases where the backend definitions had been updated since the application page was created, and the page accommodated for this in APEX 5.0. However, when ported to the APEX 19.2 environment, it generated errors because it was looking for fields that did not exist in the source object anymore.

CURRENT STATUS AND FUTURE WORK

The upgrade project of the Production Optics Reporting and Tracking and the Shot Planner applications from Oracle APEX version 5.0 to Oracle APEX 19.2 was the first phase of an overarching project to keep it in an easily maintainable state, following recommendations from Oracle on best practices with Oracle APEX. The next phase of the project includes upgrading the Oracle APEX Theme for the two applications to the Oracle recommended Universal Theme. This upgrade to Oracle APEX 19.2 allows for these aging Oracle APEX applications to function in a more sustainable and secure way, while enhancing user experience with a more modernized GUI for existing and future Oracle APEX webpages.

ACKNOWLEDGMENT

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. LLNL-CONF-827563.

REFERENCES

[1] National Ignition Facility & Photon Science, https://lasers.llnl.gov

[2] APEX Application Development, www.oracle.com/database/technologies/appdev/apex-what-is.html

[3] Switching from AnyChart to JET Chart, https://docs.oracle.com/en/database/oracle/application-express/19.2/htmdb/switching-AnyChart-to-JET-Chart.html

[4] Creating Custom Activity Reports Using APEX_ACTIVITY_LOG, https://docs.oracle.com/cd/E59726_01/doc.50/e39147/advnc_act_log.htm#HTMDB130