APEX 5 New Features - auch die Kleinen verdienen Beachtung!

Dietmar Aust
Opal-Consulting, Köln
www.opal-consulting.de

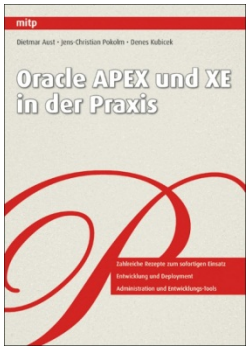
ORACLE ACE

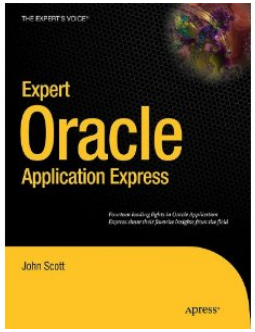
OPAL Consulting

Opal Consulting

- Building Oracle based Web Applications since 1997
  - Portal, Forms, Reports, OWA Toolkit, now APEX!
- Dipl.-Inform. Dietmar Aust, Freelance Consultant
  - Master's Degree in Computer Science (MSCS)
- 1997-2000: Consultant at Oracle Germany
- Since 09/2000: Freelance Consultant, Since 2006 – APEX only!
- Blog: http://daust.blogspot.com/
- Regular presenter at Oracle conferences (ODTUG, DOAG, OOW)
- Author of the JasperReportsIntegration toolkit
  - http://www.opal-consulting.de/tools

ORACLE ACE





Which Features To Talk About?

APEX 5 New Features
Marquee Features of Oracle APEX 5.0

- Marquee Features of Oracle APEX 5.0:
  - Page Designer
  - Multiple Interactive Reports
  - Modal Dialogs
  - Navigation Lists
  - Enhanced Mobile Support
  - Enhanced Calendar Support
  - Universal Theme
  - HTML5 Capabilities
  - Application Builder Security
- Numerous functional improvements.

APEX 5 New Features
List of available new features

- List of new features for Oracle APEX 5.0: https://apex.oracle.com/pls/apex/f?p=65339:1:0

Oracle Application Express 5.0 Early Adopter Feature Descriptions

Logout

Features | Known Issues

Hierarchical View | Flat View

Features

- APEX 5.0 Early Adopter
- Page Designer
- Universal Theme
- Interactive Reports
- Modal Pages
- Navigation Lists
- Mobile
- Calendars
- Packaged Applications
- File Handling
- Websheets

HIERARCHICAL VIEW - Marquee Feature:

Go | Reports | 1. Primary Report | Actions

Feature	Available	Summary	Lvl
(+) APEX 5.0 Early Adopter	EA1		1
... (+) 3rd Party Libraries	EA1	Upgrades, and new third-party components incorporated into the Application Builder	2
... (+) APEX 5.0 Other Features	EA1	Other features introduced in Application Express 5.0	2
... (+) Calendars	EA1	Completely new calendar component that allows displaying date entries in a monthly, weekly, daily and list view. This new calendar supports desktop and mobile applications, editing of date entries in a modal dialog and re-scheduling of events via drag & drop. The calendar look and feel can be customizable via CSS and uses responsive design. Date entries are also available for export in CSV, iCal, PDF and XML format.	2

What is possible => Samples!

APEX 5 New Features
List of available new features

- Packaged Applications => Lots of samples

Sample Database Application
Mobile
Help
dietmar.aust

Home
Customers 8
Products 10
Orders 10
Reports
Administration
Feedback
Reports
Sample Data
States
Theme Style

Sample Database Application
Track and Manage Customers, Orders and Products

Search customers, orders & products

Dashboard

$3,565
Monthly Sales

10
Total Products

Top Customers

Bradley, Eugene $2,760.00
Logan, Edward $2,420.00

Number of Orders 2
Address Schoephoester Road
City Windsor Locks
State CT
Postal Code 06096
Email
Phone Number 860-555-1835
Credit Limit 1000
Tags REPEAT CUSTOMER

$2,700.00
Bag - 16 x $1,25 $2,000.00

APEX 5 New Features
List of available new features

Sample Dynamic Actions

Help | dietmar.aust

- Home
- Simple
- Style
- Server Side
- Complex
- Administration

Sample Dynamic Actions

Add interactivity to your applications with dynamic actions

This application demonstrates many of the native dynamic actions that ship with Application Express and also a number of plug-in dynamic actions. Each page containing the dynamic action will have an 'Information' region on the top, explaining the purpose of the dynamic action and a breakdown of which are used and whether they are native or plug-in.

Debugging dynamic actions in Application Express is slightly different than other debugging, because much of the processing done with the dynamic action framework is done on the client, not on the server. In order to debug dynamic actions, we output debug information when running in 'Debug' mode to the browser's JavaScript console, if your browser supports it (for example Firefox with Firebug installed will show the debug information in its Console pane). The debug information will tell you when a dynamic action fires, the name of the dynamic action and also specifically which action has fired. In this application, debugging has already been enabled at application level. If you're interested to see exactly when a dynamic action is firing, just switch this on and interact with the example pages.

Please click on any of the examples to get started.

Disable/E…	Hide/Show	Add/Rem… Class (Error)	Add/Rem… Class (Focus)
Disable and enable items automatically	Declaratively hide and show items based on	Using the Add and Remove Class actions	Using the Add and Remove Class actions

APEX 5 New Features
List of available new features

Sample Reporting

Help | dietmar.aust

- Sample Reports
- Use Cases
- SQL Examples
- Analytic Functions
- Administration

Sample Reporting

Demonstration of reports and reporting techniques in Oracle APEX

This application highlights how to write Oracle SQL and the primary reporting engines of Oracle Application Express, interactive reports and classic reports. Use this application to better understand the native and declarative reporting functionality of Oracle Application Express and how to write simple and advanced SQL.

Interactive Report

Reports which enable customization by the end user

Classic Report

Tabular data which can be filtered by page item values

Sidebar Filter Report

Classic report with a sidebar "e-commerce" filter

Use Cases

Demonstrations of advanced APEX report techniques

SQL Examples

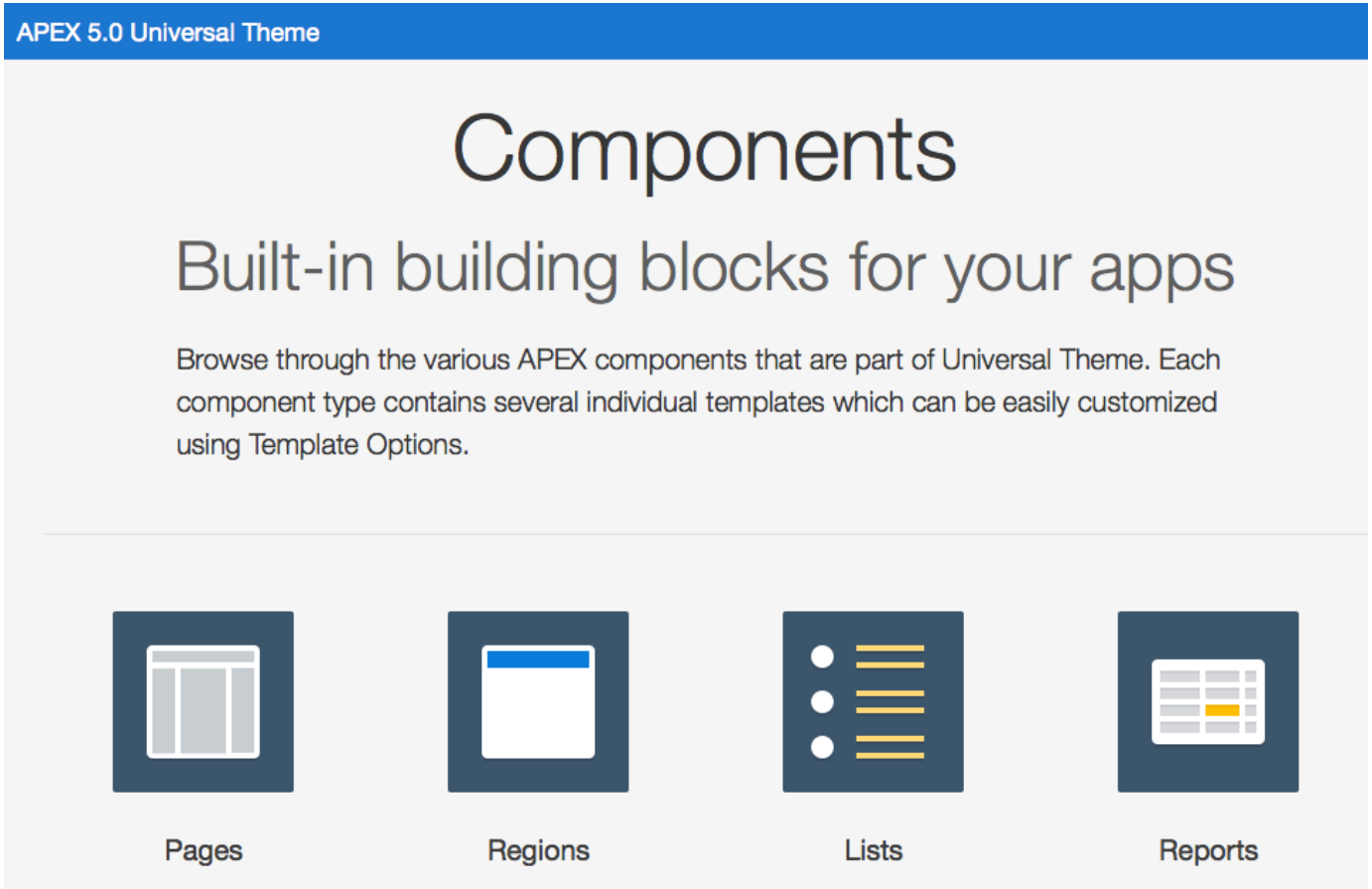
Demonstrations of advanced SQL techniques

Analytic Functions

Demonstrations of various analytic functions available in Oracle SQL.

APEX 5 New Features
List of available new features

- Universal Theme Sample Application



Changes to the APEX IDE

Changes to the APEX IDE
Workspace Login / Flexible Authentication

Use a different authentication scheme to log into the Development Environment.

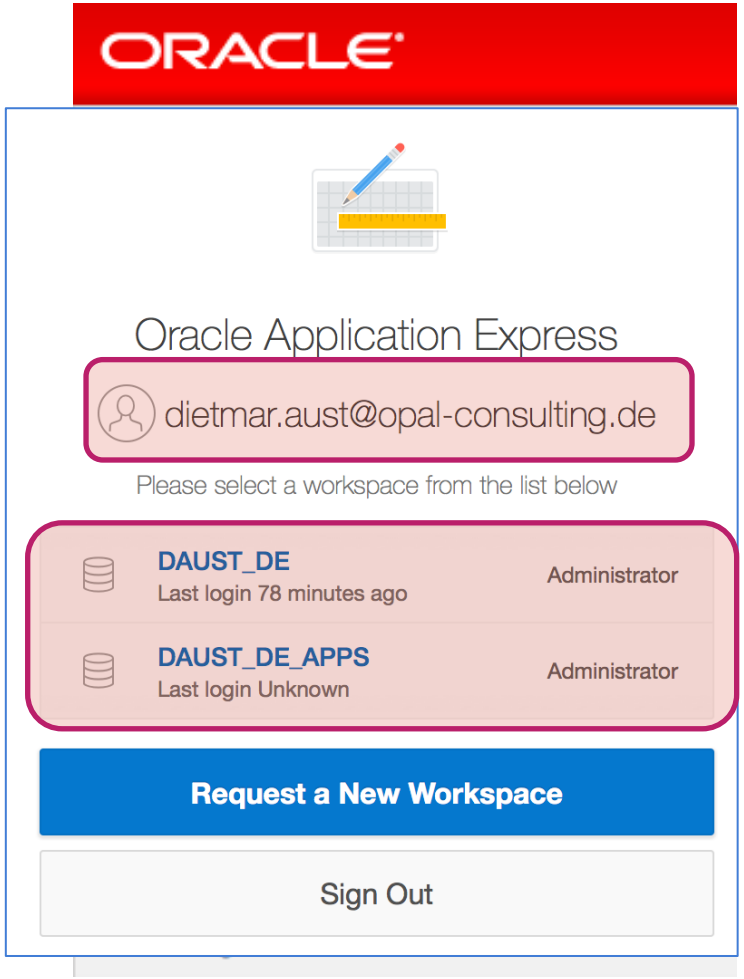
- Why?
  - Customers require to authenticate against the central user repository, typically an external SSO Server or LDAP directory
- How does it work?
  - This can be configured in the INTERNAL Administration: Manage Instance > Security

Development Environment Authentication Schemes

	Name	Description	Status
	Application Express Accounts	This is the default authentication scheme. It authenticates users against the workspace user repository.	Current
	Database Accounts	This scheme utilizes database credentials. The user name and password of the database account is used to authenticate the user.	Not Current
	HTTP Header Variable	This scheme relies on a HTTP header variable to contain the username and on an external login method to log in. Application Express presents the available workspaces for the user.	Not Current
	LDAP Directory	This scheme checks credentials against an LDAP repository.	Not Current
	Oracle Application Server Single Sign-On	This scheme authenticates developers with Oracle Single Sign-On, using SSO's external login page. After authentication Application Express presents the available workspaces for the user.	Not Current

Changes to the APEX IDE
Workspace Login / Flexible Authentication

- Display a list of available workspaces for the following authentication schemes
  - HTTP Header Variable
  - Oracle Single Sign On Server



Changes to the APEX IDE
Workspace Login / Flexible Authentication

- After upgrading to APEX 5.0 the status quo remains, **this feature has to be activated explicitly!**

- When upgrading APEX instances there is work to be done BEFORE we can activate the external authentication

- Translate all user accounts to a corresponding entity in your SSO system:

DIETMAR	=>	DIETMAR.AUST@OPAL-CONSULTING.DE
DIETMAR.AUST	=>	DIETMAR.AUST@OPAL-CONSULTING.DE
ADMIN (WS TEST)	=>	DIETMAR.AUST@OPAL-CONSULTING.DE
ADMIN (WS INTERNAL)	=>	DIETMAR.AUST@OPAL-CONSULTING.DE

- **Option 1**: Change the existing accounts (rename) either manually or via script:

```
BEGIN
    APEX_UTIL.SET_USERNAME(
        p_userid     => APEX_UTIL.GET_USER_ID('DIETMAR'),
        P_username   => 'DIETMAR.AUST@OPAL-CONSULTING.DE');
END;
```

Changes to the APEX IDE
Workspace Login / Flexible Authentication

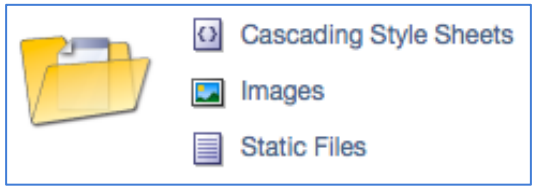
- **Option 2:** Create additional users, e.g. DIETMAR.AUST@OPAL-CONSULTING.DE
  - The users ADMIN, DIETMAR and DIETMAR.AUST will no longer be able to log into the workspace because they don't exist in the SSO server.
  - But they can still be used for your application and for the websheets.

Changes to the APEX IDE
Improved File Storage

The storage of files in the application builder has been completely overhauled.

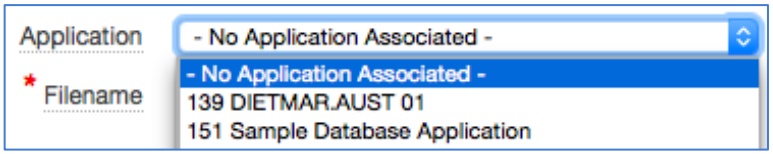
- Why?

- Too many different file types:



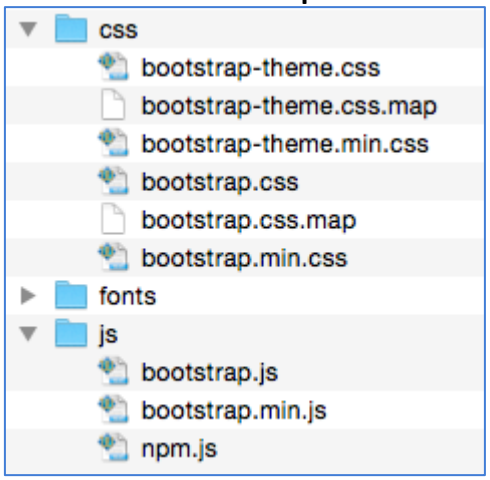
  - Reference via
    - #WORKSPACE_IMAGES#
    - #APP_IMAGES#

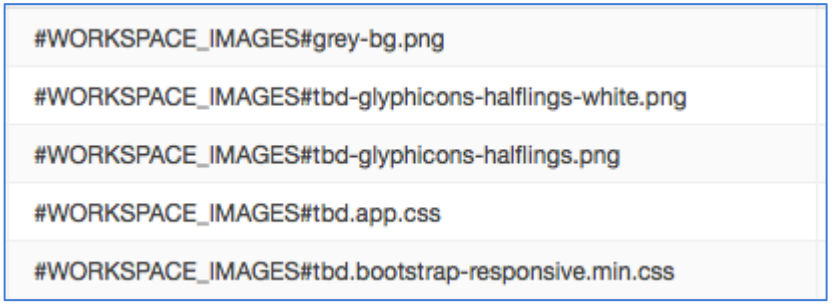
Assigned to either Workspace or Application



Changes to the APEX IDE
Improved File Storage

- No relative file paths available

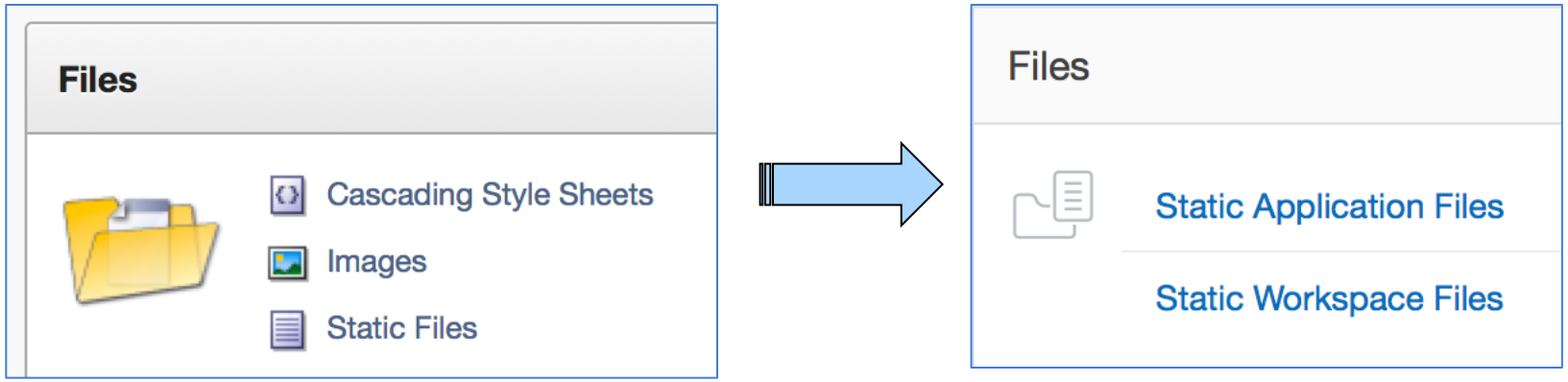


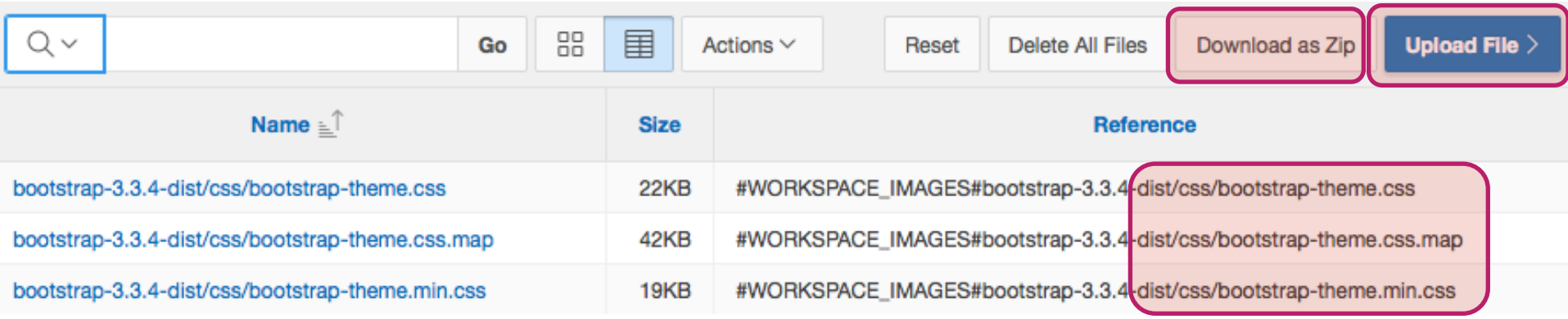


- All files had to be uploaded individually by a manual procedure
- Updating many files was a pain
- Moving the files to a different instance was cumbersome
- Performance!

Changes to the APEX IDE
Improved File Storage

The storage of files in the workspace has been unified:

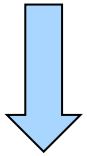




Changes to the APEX IDE
Improved File Storage

- At runtime the references will be translated:

```
#APP_IMAGES#readme.txt
```



```
daust_de/r/467/files/static/v6/readme.txt
```

- Once the files are changed (by uploading them again) a new url (containing a new version number) is generated and invalidates the browser cache.

```
daust_de/r/467/files/static/v7/readme.txt
```

- Caching of 10 years

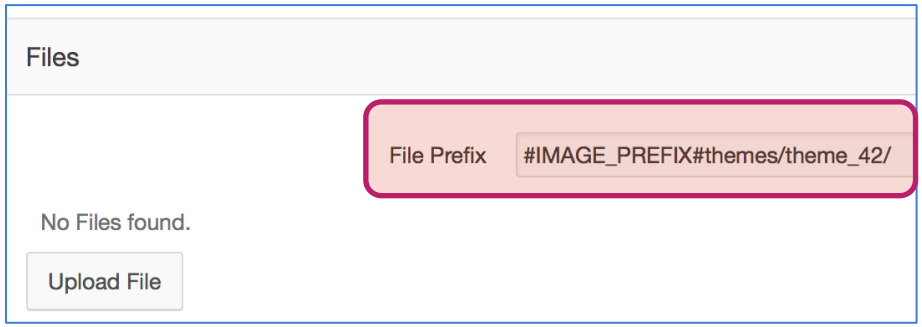
Changes to the APEX IDE
Improved File Storage

- Mechanism works with:
  - Oracle Rest Data Services
  - OHS / mod_plsql
  - Embedded PL/SQL Gateway

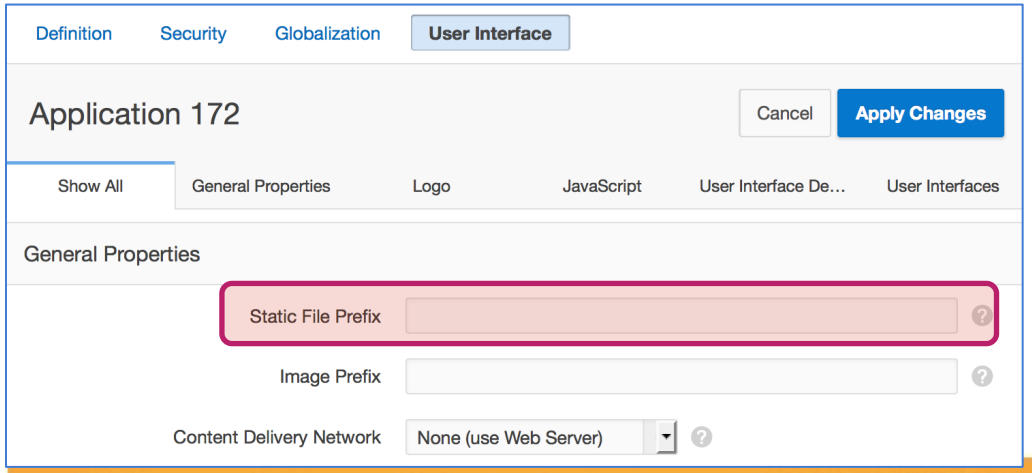
- Mechanism is available for (and their substitution variables)
  - Static Application files #APP_IMAGES#
  - Static Workspace files #WORKSPACE_IMAGES#
  - Theme files #THEME_IMAGES#
  - Plugin files **p_plugin.file_prefix or #PLUGIN_FILES#**

Changes to the APEX IDE
Improved File Storage

- For even increased performance (or a simplified development lifecycle) the files can be moved to the filesystem:
  - Theme static files

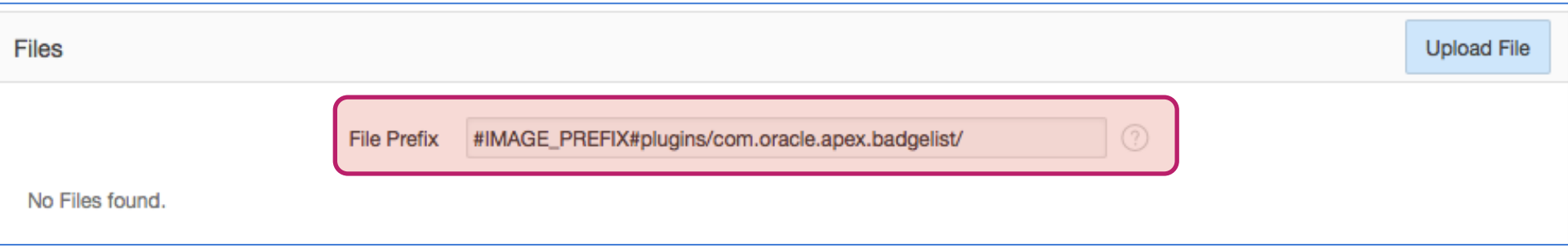


  - Application static files



Changes to the APEX IDE
Improved File Storage

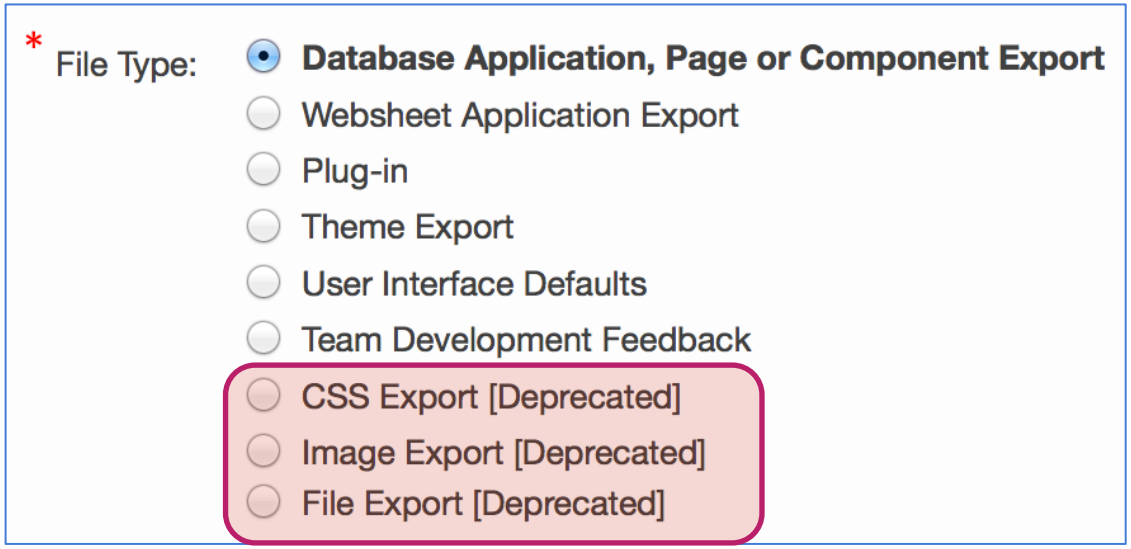
- Plugin static files:



- The reference to the file will NOT change: #THEME_IMAGES#
  - => NO change in your application code is required!

Changes to the APEX IDE
Improved File Storage – Change to the Application Export File

- You cannot export the Application Static Files any more:

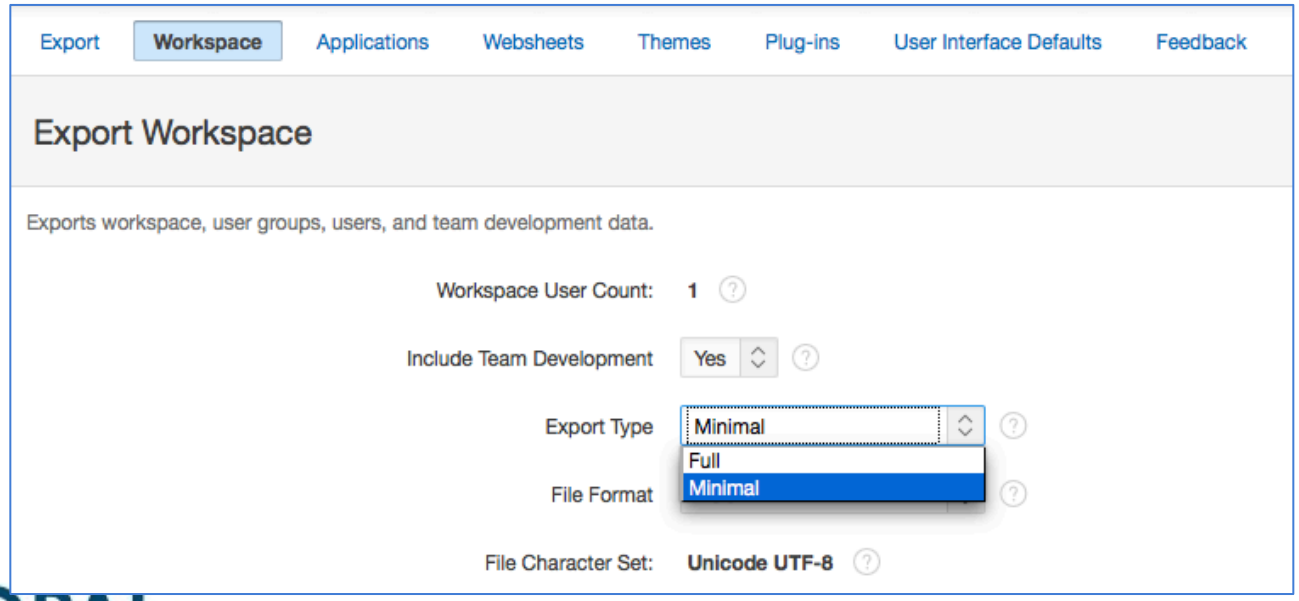


- The Application Static Files are now ALWAYS automatically a part of the application export **!!! This is a huge win!!!**
  - Shipping logos, user documentation (pdfs) or additional CSS files together with the application

Changes to the APEX IDE
Improved File Storage – Change to the Application Export File

- The workspace export has two options now

- You can select the mode in the export dialog:
  - Minimal: The same as before
  - Full: Everything



Changes to the APEX IDE
Improved File Storage – Change to the Application Export File

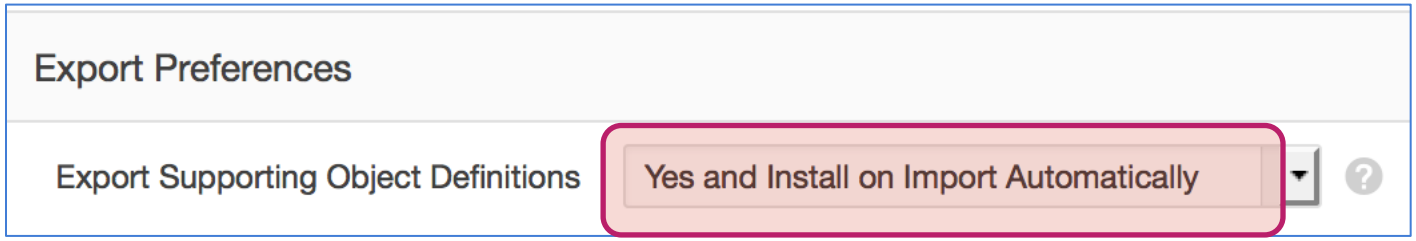
- The FULL workspace export now contains EVERYTHING (with the exception of applications, websheets and workspace static files)
  - -- W O R K S P A C E
  - -- G R O U P S
  - -- U S E R S
  - --Application Builder Preferences
  - --Click Count Logs
  - --csv data loading
  - --mail
  - --mail log
  - --app models
  - --password history
  - --preferences
  - --query builder
  - --sql scripts
  - --sql workshop history
  - Restful service definitions, ...
- Why? Used for the Oracle Cloud to move workspaces do a different instance transparent to the user.
- Export of the workspace static files using APEXExport command line utility in the apex/utilities directory (see the Readme.txt file there):

```
java oracle.apex.APEXExport -db localhost:1521:xe -user test
     -password <pwdTest> -workspaceid 2614526486790575 -expFiles
```

Changes to the APEX IDE
Application Export – Command line install supporting objects

Install supporting objects automatically on the command line.

- Why?
  - Currently it is NOT possible to install supporting objects on the command line (install on Runtime Only environment **impossible**)
- How is it implemented?
  - Added a new option in the export dialog: Export Preferences > Export Supporting Object Definitions:



  - This will generate a changed export application file:

```
wwv_flow_api.import_end(p_auto_install_sup_obj => true);
```

Changes to the APEX IDE
Application Export – Optimized Application Export File Format

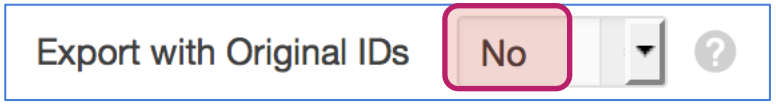
- Can also be used on the command line with apex_application_install:

```
APEX_APPLICATION_INSTALL.SET_AUTO_INSTALL_SUP_OBJ(
    p_auto_install_sup_obj IN BOOLEAN);
```

Changes to the APEX IDE
Improved File Storage – Change to the Application Export File

The export application file has been optimized.

- Smaller size, faster to install
- Option to export using the Original IDs
  - Better support for using file differencing tools to show the actual differences between files.
  - Better suited for version control systems



Changes to the APEX IDE
Improved File Storage – Concatenated Files

Concatenated files in the user interface section, this is a runtime replacement of file urls to consolidate multiple single files into a consolidated one at runtime.

- Why?
  - Performance vs. modular development:
  - During development, you want to use multiple CSS, JS files
  - In production you want to include just a single minified file

- How does it work?
  - We can reference external files in our application (application level, page level)
  - In the **`user interface details`** section we can specify **multiple** concatenated files:

Concatenated Files

Create Concatenated File >

File URL	Only for Page	Build Option
#APP_IMAGES#js/all#MIN#.js?version=#APP_VERSION#	-	

1 - 1

Changes to the APEX IDE
Improved File Storage – Concatenated Files

- Concatenated file definition:

* Concatenated File URL	#APP_IMAGES#js/all#MIN#.js?version=#APP_VERSION#
* Single File URLs	#APP_IMAGES#js/core#MIN_DIRECTORY#.js?version=#APP_VERSION# #APP_IMAGES#js/search.widget#MIN#.js?version=#APP_VERSION# #APP_IMAGES#js/profile.widget#MIN#.js?version=#APP_VERSION#
Only for Page	
Build Option	- No Build Option -

- Each occurrence of a single file url will be replaced with the concatenated file (only once per page rendering)

Changes to the APEX IDE
Improved File Storage – Concatenated Files

- Placeholders for referenced URLs in general:
  - `#MIN#` => `.min`
  - `#MIN_DIRECTORY#` => `minified/`

- Example URL to use in your application

  `#APP_IMAGES#js/#MIN_DIRECTORY#search_widget.js`

- During Development (logged into the workspace)

  1. Concatenated URLs will not be replaced
  2. #MIN# and #MIN_DIRECTORY# => NULL

`Test/r/101/files/static/v1/js/search_widget.js`

- At Runtime (not logged into the workspace)

`Test/r/101/files/static/v1/js/minified/search_widget.js`

Changes to the APEX IDE
Subscription of themes

Themes and templates can be subscribed on a theme level now, even themes from the Theme Repository

- Why?
  - Previously only individual templates could be subscribed to, now whole themes including all templates => ease of use
  - Now templates can be updated safely (bug fixes, other improvements)
- How does it work?
  - When subscribing, all templates are READONLY
  - => Bugs can be fixed in patchsets or later releases because the theme clearly was not modified!

Theme Subscription

Verify | Unsubscribe | Refresh Theme

Subscribed to Standard Theme : **Universal Theme**

No themes subscribe to this theme.

Changes to the APEX IDE
View Subscription Easily

- See subscribed masters of elements:
  - Plugins, LOVs, Themes, Authentication Schemes, Authorization Schemes, …

Themes | Reports | History

Go | Actions

Number	Name	User Interface	Is Current	Subscribed From	Subscribers	Templates
42	Universal Theme - 42 *	Desktop	✓	Theme Repository		52
199	Scarlet - 199	Desktop		20150318		93

Changes to the APEX IDE
Application – Level CSS Files

Add application specific CSS files to the user interface which is NOT part of the theme.

- Why?
  - Clean separation of concerns (themes were never truly generic)
    - a theme should be generic for all applications,
    - application specific styles should NOT be part of the theme
- How?
  - New section in the Application Attributes > User Interface section
- The application specific Javascript section was moved to the User Interface defaults as well
- Used with the readonly subscription of themes this is really clean and themes can now be truly generic

Changes to the APEX IDE

User Interface Details

Cancel | Delete | **Apply Changes**

Show All | Identification | Attributes | Navigation Menu | Navigation Bar | JavaScript | Cascading Styl... | Concatenated ...

JavaScript

Content Delivery Network: None (use Web Server)

File URLs:

```
#APP_IMAGES#js/core#MIN#.js?version=#APP_VERSION#
#APP_IMAGES#js/search.widget#MIN#.js?version=#APP_VERSION#
#APP_IMAGES#js/profile.widget#MIN_DIRECTORY#.js?version=#APP_VERSION#
```

Include Legacy Javascript: No

Include jQuery Migrate: No

Cascading Style Sheets

File URLs:

Concatenated Files

Create Concatenated File >

Changes to the APEX IDE
Session State Protection Enabled by Default

By default Session State Protection is enabled when creating new applications or new pages.

- Why?
  - Improved security, now EVERY NEW application has session state protection enabled
  - Now we cannot forget to add session state to the application or newly created pages

Application Development / Runtime Engine

Application Development / Runtime Engine
Modal Pages

Pages can have a page mode to make them being run as a modal page.

- Why?
  - Common user interface technique well known in native application.
  - Good usuability, because no additional navigation context is required.
  - Provide a rich, “complex” UI composed of simple technical elements.

- How? Requirements?
  - Theme 42 (Universal Theme), because there is the dialog initialization code

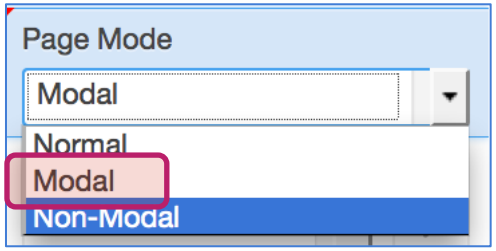
Dialog Initialization Code	apex.navigation.dialog(#PAGE_URL#,{title:#TITLE#,height:#DIALOG_HEIGHT#,width:#DIALOG_WI
Dialog Closure Code	apex.navigation.dialog.close(#IS_MODAL#,#TARGET#);
Dialog Cancel Code	apex.navigation.dialog.cancel(#IS_MODAL#);

  - Note that options *Modal* and *Non-Modal Dialog* will only be available for selection where the current application theme contains a default *Dialog Page Template*. => it will not work otherwise

Application Development / Runtime Engine
Modal Pages

- Use Case #1: Simple Modal Page – the processing does not change
  - The target page is opened as a modal dialog
  - Regular ACCEPT / SHOW processing applies, **no partial page refresh**
  - Simplest integration

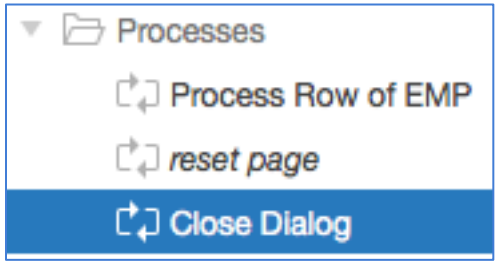
- Steps:
  1. Set Page Mode of the target page to "Modal"

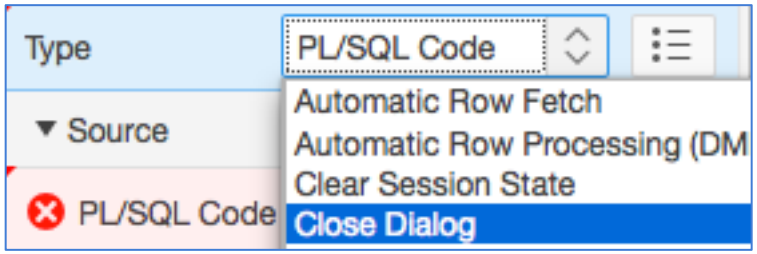


Application Development / Runtime Engine
Modal Pages

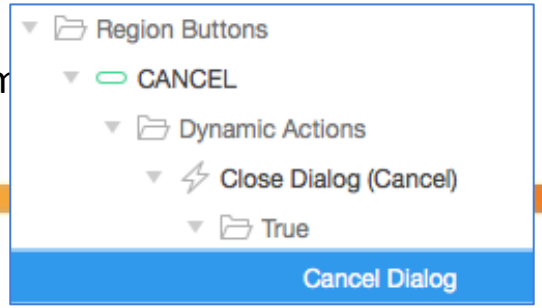
- Use Case #2: Modal Page – **with partial page refresh**
  - The target page is opened as a modal dialog
  - The dialog is closed
  - The calling page processes the close event and refreshes a region

- Steps:
  1. Target Page: Set Page Mode to “Modal”
  2. Target Page: Add process “Close Dialog” without condition => Stops branching





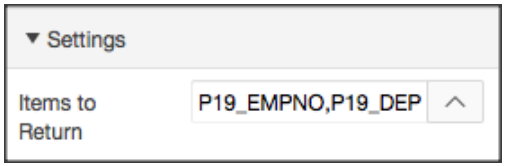
  3. Calling Page: Create Dynamic Action for the “Close Dialog” Event
  4. Target Page: Cancel Button closes the page with a dynam “Cancel Dialog”

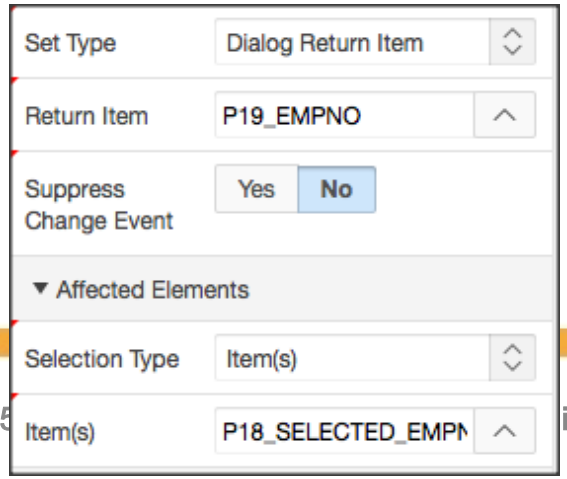


Application Development / Runtime Engine
Modal Pages

- Use Case #3: Modal Page – **Return item values to the calling page**
  - Return item values from the target page to the calling page

- Steps:
  1. Target Page: Modify process "Close Dialog" and set the return items
  2. Calling Page: Create Dynamic Action on "Dialog Closed" event
     - use Javascript to retrieve the value:
       `this.data.P19_EMPNO`
     - Or use a dynamic action of type "Set Value"





Application Development / Runtime Engine
Modal Pages

- Use Case #4: Modal Page – **Distinguish between Dialogs**
  - We want to implement different Dynamic Actions on the calling page depending on the buttons we press.

- Steps:
  1. Calling Page: Attach the Dynamic Action to the REGION "Employees"
     - This will catch all events created by the "Edit Link" and the "Create" button (all events will bubble upwards)
  2. Calling Page: Attach the Dynamic Action to the button "Select with LOV"

Application Development / Runtime Engine
Modal Pages

Los | Aktionen | Create

Employee Id	First Name	Last Name	Email	Phone Number	Hire Date	Job Id
100	Steven	King	SKING	515.123.4567	17.06.1987	AD_PRES
101	Neena	Kochhar	NKOCHHAR	515.123.4568	21.09.1989	AD_VP
102	Lex	De Haan	LDEHAAN	515.123.4569	13.01.1993	AD_VP
103	Alexander	Hunold	AHUNOLD	590.423.4567	03.01.1990	IT_PROG
104	Bruce	Ernst	BERNST	590.423.4568	21.05.1991	IT_PROG
105	David	Austin	DAUSTIN	590.423.4569	25.06.1997	IT_PROG
106	Valli	Pataballa	VPATABAL	590.423.4560	05.02.1998	IT_PROG
107	Diana	Lorentz	DLORENTZ	590.423.5567	07.02.1999	IT_PROG
108	Nancy	Greenberg	NGREENBE	515.124.4569	17.08.1994	FI_MGR
109	Daniel	Faviet	DFAVIET	515.124.4169	16.08.1994	FI_ACCOUN
110	John	Chen	JCHEN	515.124.4269	28.09.1997	FI_ACCOUN
111	Ismael	Sciarra	ISCIARRA	515.124.4369	30.09.1997	FI_ACCOUN
112	Jose Manuel	Urman	JMURMAN	515.124.4469	07.03.1998	FI_ACCOUN

About

This is the demo we use two modal dialogs. When you create or edit an employee, the selected employee_id and job will be returned into the region "Selected Employee".

When you click on "Select with LOV" another modal dialog opens and you can choose an employee. After you choose an employee, the selected employee will be returned into the second region.

Selected Employee

Employee ID

Job

Selected Employee (with LOV)

Employee ID

Select with LOV

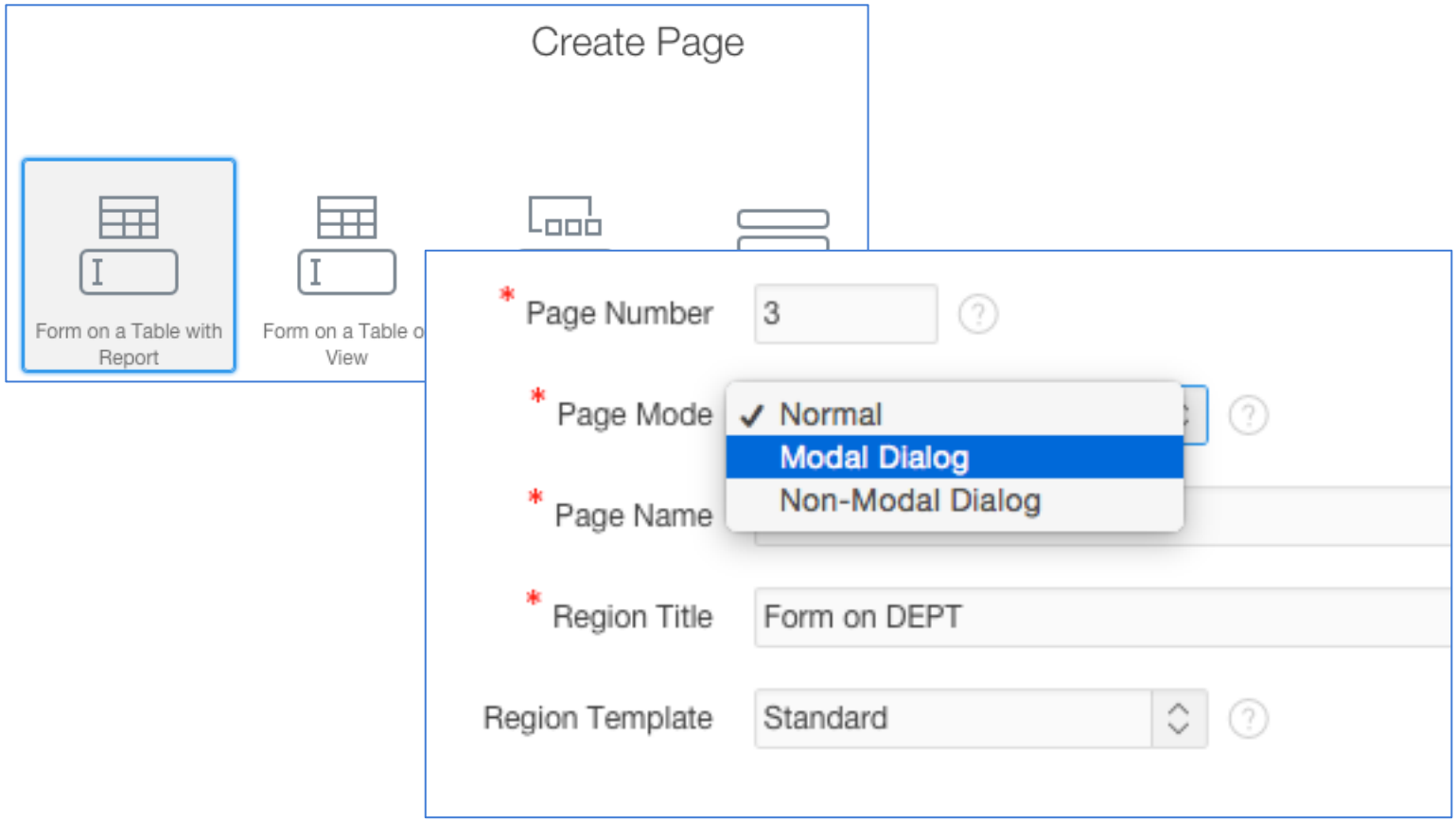
Application Development / Runtime Engine
Modal Pages

- Use Case #5: Create a link manually:

```
apex_util.prepare_url
```

Application Development / Runtime Engine
Modal Pages

- The Form & Report Wizard supports modal dialogs directly



Application Development / Runtime Engine
Session Joining

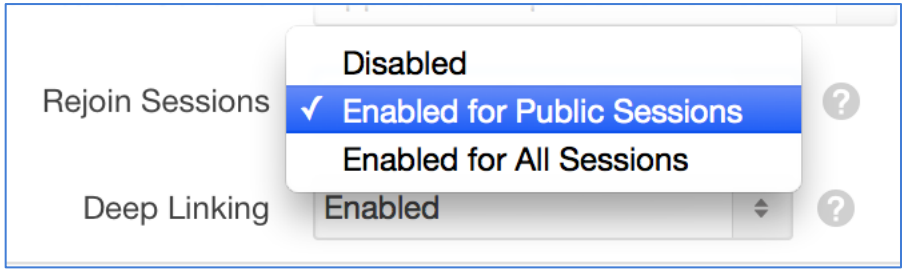
You can now join the current APEX application session in your browser.

- Why?
  - When sending emails with links to your application in APEX 4.2 your current session gets killed.
    - You have to login again
    - The active session is invalidated

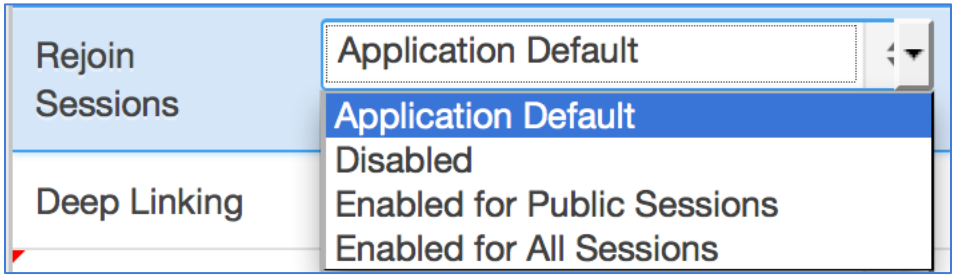
- How does it work?
  - The session id is removed and the session is purely managed through the browser cookie.
  - http://apex.oracle.com:8080/ords/f?p=101:1:1234567890::NO:::
  - =>
  - http://apex.oracle.com:8080/ords/f?p=101:1:::NO:::

Application Development / Runtime Engine
Session Joining

- Can be controlled on application level



- and individual page level



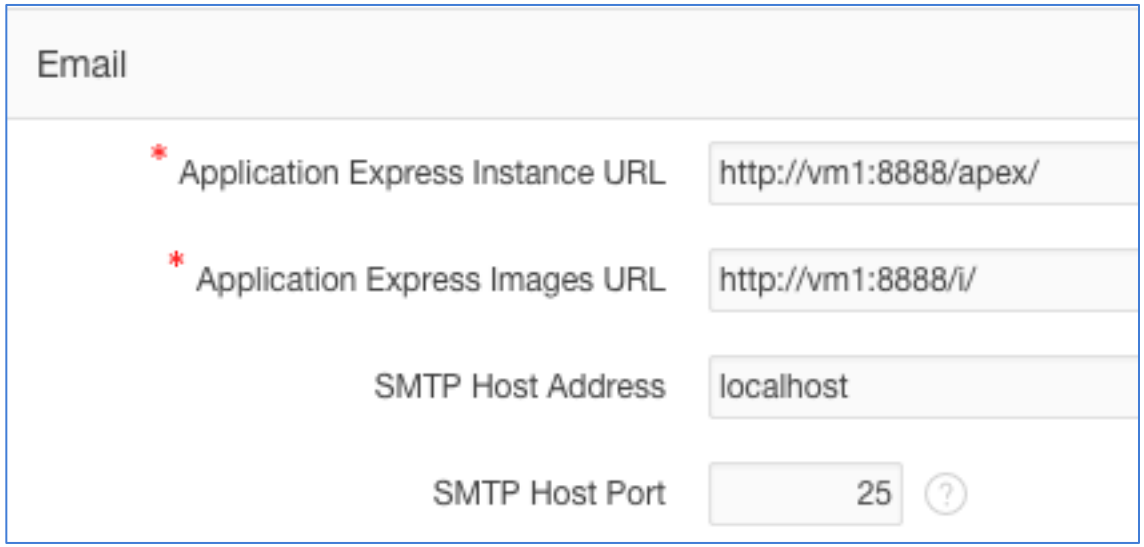
- Can be disabled globally at the instance level. THE SETTING AT THE INSTANCE LEVEL DETERMINES THE MAXIMUM SETTING!

Application Development / Runtime Engine
Session Joining

- Also, there are useful features in APEX 4.2 for sending out emails (configured in the instance settings):

```
APEX_MAIL.GET_IMAGES_URL return VARCHAR2;

APEX_MAIL.GET_INSTANCE_URL return VARCHAR2;
```



Application Development / Runtime Engine
Session context through SYS_CONTEXT

APEX session variables are now available through SYS_CONTEXT, to be used in Views, Packages and Triggers.

- Why?
  - Make it consistent with other database information about the user
  - Performance, more efficient compared to v('APP_USER'), v('APP_SESSION')

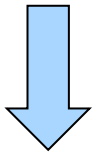
- How does it work?

```
SELECT SYS_CONTEXT ('userenv', 'session_user') session_user,
       SYS_CONTEXT ('userenv', 'current_user') parsing_schema,
       SYS_CONTEXT ('APEX$SESSION', 'workspace_id') workspace_id,
       SYS_CONTEXT ('APEX$SESSION', 'app_session') app_session,
       SYS_CONTEXT ('APEX$SESSION', 'app_user') app_user
  FROM DUAL
```

Application Development / Runtime Engine
Session context through SYS_CONTEXT

- Typically used in Views and Triggers in the database for row level security and auditing of changes, e.g. in triggers to set `:NEW.LAST_UPDATED_BY`

```
NVL(v('APP_USER'), USER)
```



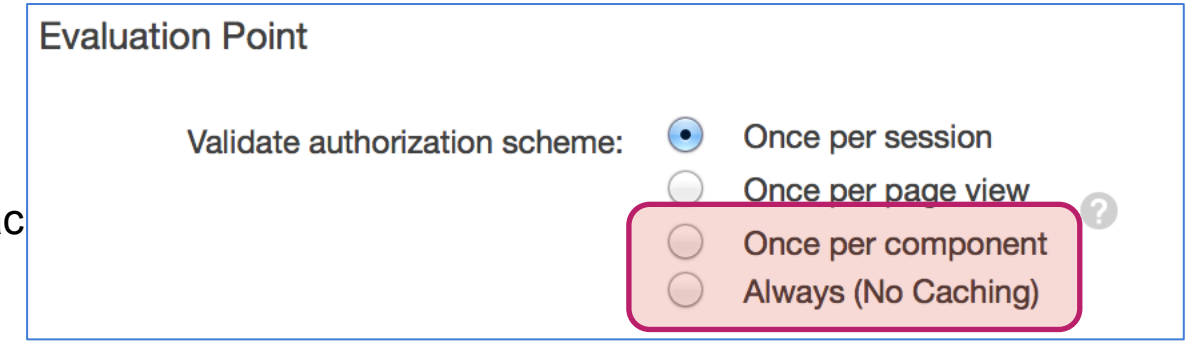
```
COALESCE(SYS_CONTEXT('APEX$SESSION', 'app_user'), USER)
```

- Using `v('APP_USER')` in Views CAN have negative performance impact:
  - http://inside-apex.blogspot.com/2006/11/caution-when-using-plsql-functions-in.html
- NVL always evaluates both arguments, while COALESCE stops evaluation whenever it finds first non-NULL

Application Development / Runtime Engine Improvement to Authorizations

New evaluation points for authorizations.

- Why?
  - Implement a generic and user configurable approac to securing the app components



- New variables to be used in authorizations
  - APP_COMPONENT_TYPE
  - APP_COMPONENT_ID
  - APP_COMPONENT_NAME

```
begin
  xlog('generic auth',
       'APP_PAGE_ID: ' || :app_page_id
         || '<br />APP_COMPONENT_TYPE: '||:APP_COMPONENT_TYPE
         || '<br />APP_COMPONENT_ID: '  ||:APP_COMPONENT_ID
         || '<br />APP_COMPONENT_NAME: '||:APP_COMPONENT_NAME) ;
  return true;
end;
```

Application Development / Runtime Engine
Improved security with appropriate escaping

Extended substitution syntax where you define the escaping mode.

- Why?
  - Protection against cross-site scripting attacks when using `&P1_ITEM` or `#REPORT_COLUMN#` references.

Item Syntax	Report Column Syntax	Internal function call
&P1_ITEM!HTML.	#REPORT_COLUMN!HTML#	apex_escape.html
&P1_ITEM!ATTR.	#REPORT_COLUMN!ATTR#	apex_escape.html_attribute
&P1_ITEM!JS.	#REPORT_COLUMN!JS#	apex_escape.js_literal(,null)
&P1_ITEM!RAW.	#REPORT_COLUMN!RAW#	unescaped
&P1_ITEM!STRIPHTML.	#REPORT_COLUMN!STRIPHTML#	strip tags, then apex_escape.html

Application Development / Runtime Engine
Improved security with appropriate escaping

- Demo
- Dynamic action using vulnerable Code: `alert("&P16_TEXT.");`
- Show regular use using the literal: `test`
- Show exploit using: `");alert("you got hacked!`
- Fixed Code: `alert("&P16_TEXT!JS.");`

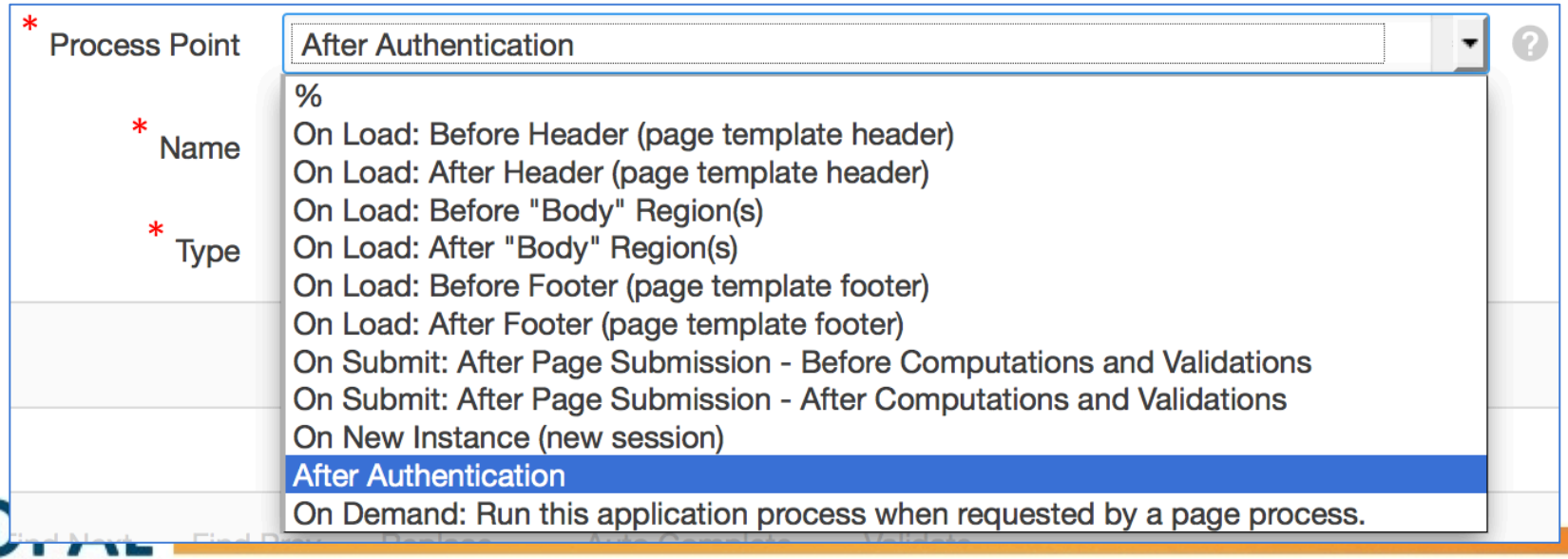
```
apex.da.gEventList = [
{"bindEventType":"ready",actionList:
[{"eventResult":true,"executeOnPageInit":true,"stopExecutionOnError":true,"af
(){      // escape JS?
    if (apex.item("P6_ESCAPE_JS").getValue()=="1"){
        alert("Regular Text");alert("you got hacked!");
    } else {
        // secure mode
        alert("Regular\u0020Text\u0022\u0029\u003Balert\u0028\u0022you\u0020g
    }

},"action":"NATIVE_JAVASCRIPT_CODE"}]}];
}
```

New Application Process Execution Point - After Authentication

- Why?
  - This is „On New Instance“ (new session) => done right ;)
  - We need a process execution right after the user has been authenticated to initialize some variables for the application

- How does it work?

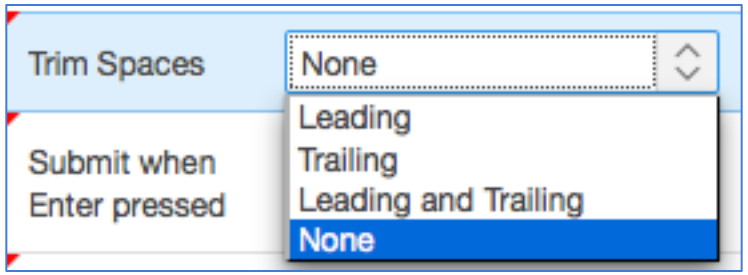


Application Development / Runtime Engine
New Application Process Execution Point - After Login

"Trim Spaces" attribute to Textfield and Textarea Item Type

- Why?
  - This feature is designed to avoid the need to create computation processes to remove unnecessary spaces from a text entry, as generally it is not desirable to store these extra spaces, tabs and new lines.
  - In order to implement this we needed an additional computation or process (per page or on application level) to do that
    - => implementation overhead
    - => performance overhead

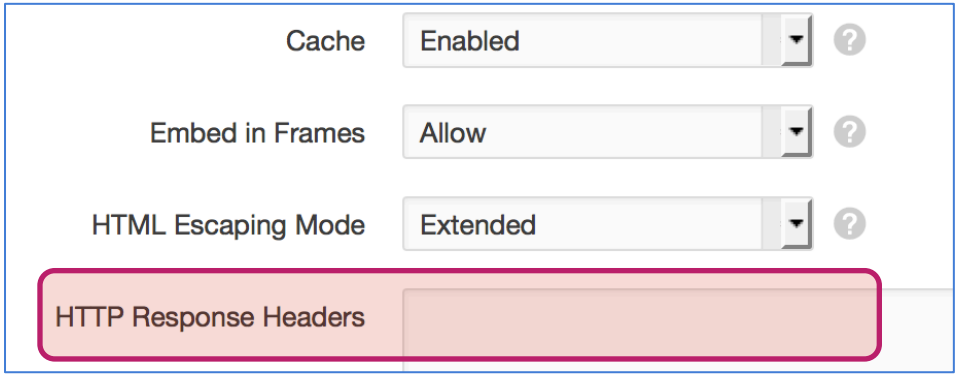
- How does it work?



Additional HTTP Response Headers

Support for sending additional HTTP Response Headers.

- Why?
  - Being able to use specific browser security features which are not declaratively available through APEX
- How?
  - Configuration setting the Application Properties > Security > Browser Security:



  - Sample:

```
X-XSS-Protection: 1; mode=block

X-Content-Type-Options: nosniff
```

API - Packages

API Packages
APEX_JSON

- APEX_JSON
- Why?
  - Generate and parse JSON programmatically
  - apex_util.json_from_sql outputs using htp.p directly, doesn't return string
  - Cannot handle more complex structures and custom tags
- Functions:
  - CLOSE_ALL
  - CLOSE_ARRAY
  - CLOSE_OBJECT
  - DOES_EXIST
  - FIND_PATHS_LIKE
  - FLUSH
- Functions:
  - GET_BOOLEAN
  - GET_COUNT
  - GET_DATE
  - GET_MEMBERS
  - GET_NUMBER
  - GET_VALUE
  - GET_VARCHAR2
  - INITIALIZE_OUTPUT
  - OPEN_ARRAY
  - OPEN_OBJECT
  - PARSE
  - STRINGIFY
  - TO_XMLTYPE
  - WRITE

API Packages
APEX_ZIP

New Package APEX_ZIP to provide ZIP archive support. Based on the original package AS_ZIP by Anton Scheffer:
http://technology.amis.nl/wp-content/uploads/2010/06/as_zip8.txt

- APEX_ZIP
  - ADD_FILE
  - FINISH
  - GET_FILE_CONTENT
  - GET_FILES

API Packages
APEX_ZIP - Upload

```
DECLARE
   l_zip_file        BLOB;
   l_unzipped_file   BLOB;
   l_files           apex_zip.t_files;
BEGIN
   SELECT file_content
     INTO l_zip_file
     from my_zip_files
    WHERE file_name = :P13_file_name;

   l_files := apex_zip.get_files (p_zipped_blob => l_zip_file);

   FOR i IN 1 .. l_files.COUNT
   LOOP
      l_unzipped_file := apex_zip.get_file_content (p_zipped_blob   => l_zip_file,
                                                    p_file_name     => l_files (i));

      INSERT INTO my_files (file_name, file_content)
           VALUES (l_files (i), l_unzipped_file);
   END LOOP;
END;
```

API Packages
APEX_ZIP - Download

```
DECLARE
   l_zip_file   BLOB;
BEGIN
   FOR l_file IN (SELECT file_name, file_content FROM my_files)
   LOOP
      apex_zip.
       add_file (p_zipped_blob   => l_zip_file,
                 p_file_name     => l_file.file_name,
                 p_content       => l_file.file_content);
   END LOOP;

   apex_zip.finish (p_zipped_blob => l_zip_file);
END;
```

API Packages
APEX_PAGE

- APEX_PAGE
  - Function is_read_only() determines whether the page is run in read_only_mode
  - Function Get_url
    - clean way to generate a link to another apex page
    - Also generates checksums for session state protection and the javascript for modal dialogs (just like apex_util.prepare_url)

```
FUNCTION GET_URL (
    p_application        IN VARCHAR2 DEFAULT NULL,
    p_page               IN VARCHAR2 DEFAULT NULL,
    p_session            IN NUMBER   DEFAULT WWV_FLOW.G_INSTANCE,
    p_request            IN VARCHAR2 DEFAULT NULL,
    p_debug              IN VARCHAR2 DEFAULT NULL,
    p_clear_cache        IN VARCHAR2 DEFAULT NULL,
    p_items              IN VARCHAR2 DEFAULT NULL,
    p_values             IN VARCHAR2 DEFAULT NULL,
    p_printer_friendly   IN VARCHAR2 DEFAULT NULL,
    p_trace              IN VARCHAR2 DEFAULT NULL )
    RETURN VARCHAR2;
```

Q&A

Dietmar Aust
Opal-Consulting, Köln

www.opal-consulting.de
daust.blogspot.com
dietmar.aust@opal-consulting.de

Changes to the APEX IDE
New Substitution Variable APP_PAGE_ALIAS

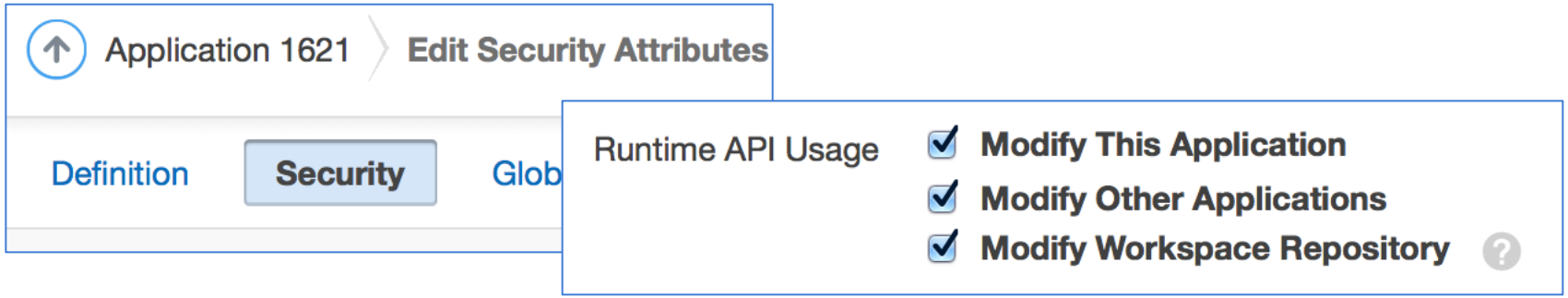
New Substitution Variable APP_PAGE_ALIAS.

- Why?
  - We can use the alias in our manually created links.
- How?
  - :APP_PAGE_ALIAS
  - &APP_PAGE_ALIAS.
- Complete list can be found here:
  - Open Documentation Homepage: https://docs.oracle.com/cd/E59726_01/index.htm
- E.G.:
  - APEX$ROW_NUM
  - APEX$ROW_SELECTOR
  - APEX$ROW_STATUS

Application Development / Runtime Engine
Lock Down Public APIs

Lock down public APIs that change the APEX repository data.

- Why?
  - Protection against sql injection attacks which can modify your application, e.g. the login page to record all passwords.
- Regular use cases for modifying the APEX metadata
  - Create / change workspace users => Modify Workspace Repository
  - Data Reporter Packaged Application creates new pages on the fly
- How does it work?



Data Dictionary

Data Dictionary
New Views

- New Views for all new features:
  - APEX_APPLICATION_PAGE_IR_PIVOT
  - APEX_APPLICATION_PAGE_IR_PVAGG
  - APEX_APPLICATION_PAGE_IR_PVSRT
  - APEX_APPLICATION_PAGE_REG_COLS
  - APEX_APPLICATION_STATIC_FILES
  - APEX_APPLICATION_THEME_FILES
  - APEX_APPL_PAGE_CALENDARS
  - APEX_APPL_TEMPLATE_OPT_GROUPS
  - APEX_APPL_TEMPLATE_OPTIONS
  - APEX_WORKSPACE_GROUP_GROUPS
  - APEX_WORKSPACE_SESSION_GROUPS
  - APEX_WORKSPACE_STATIC_FILES

Data Dictionary
Added Columns

- Extended the existing views for the functions (small subset)

APEX_VIEW_NAME	COLUMN_ID	COLUMN_NAME	COMMENTS
APEX_APPLICATIONS	66	FILE_PREFIX	File prefix which is used by the application to load additional files like CSS, Javascript and images.
APEX_APPLICATIONS	67	FILES_VERSION	Version counter of the attached application files.
APEX_APPLICATIONS	68	RUNTIME_API_USAGE	A colon separated list that identifies what metadata the application is allowed to change: (T)his application, (O)ther applications, (W)orkspace repository
APEX_APPLICATIONS	69	REJOIN_EXISTING_SESSIONS	Determines whether sessions for this application can be joined without session id in the URL
APEX_APPLICATIONS	70	HTTP_RESPONSE_HEADERS	This attribute contains http response headers that APEX should send each time for this application
APEX_APPLICATIONS	71	BOOKMARK_CHECKSUM_FUNCTION	The algorithm which is used to produce URL checksums

- APEX_WORKSPACE_SESSIONS
  - SESSION_IDLE_TIMEOUT_ON
    - Date when the APEX session times out because of inactivity
  - SESSION_LIFE_TIMEOUT_ON
    - Date when the APEX session times out because of the maximum session length has been reached:

Data Dictionary
Added Columns

- APEX_WORKSPACE_ACTIVITY_LOG
  - PAGE_VIEW_TYPE
    - The type of page view (Rendering, Processing, AJAX, Logout, Authentication callback, Other)
  - REQUEST_VALUE
    - The value of the request parameter for this page view
  - DEBUG_PAGE_VIEW_ID
    - The ID for debug messages stored for this page view

File Upload

- Dictionary views

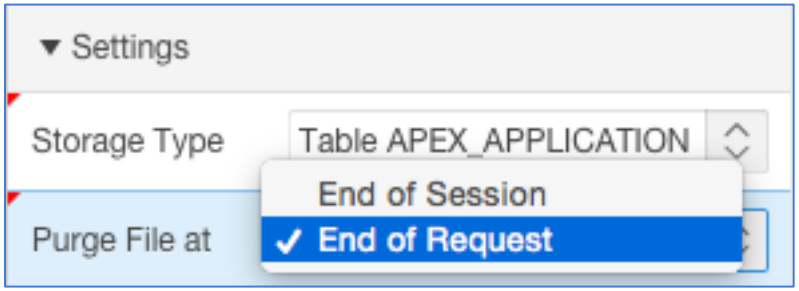
Arbeitsblatt | Query Builder

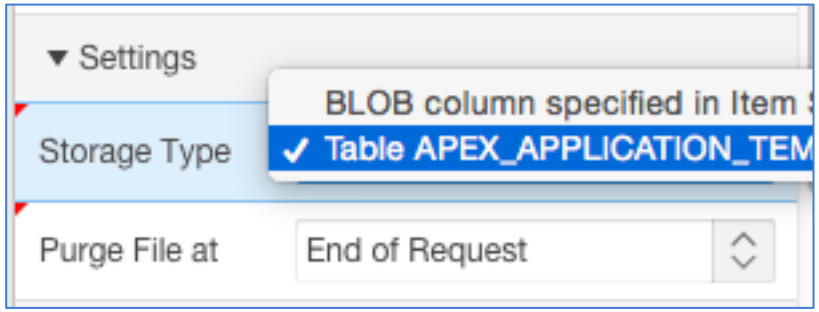
```
1 select apex_view_name,comments from apex_dictionary where apex_view_name like '%FILES%' and column_id=0;
```

Abfrageergebnis

SQL | Alle Zeilen abgerufen:6 in 1,934 Sekunden

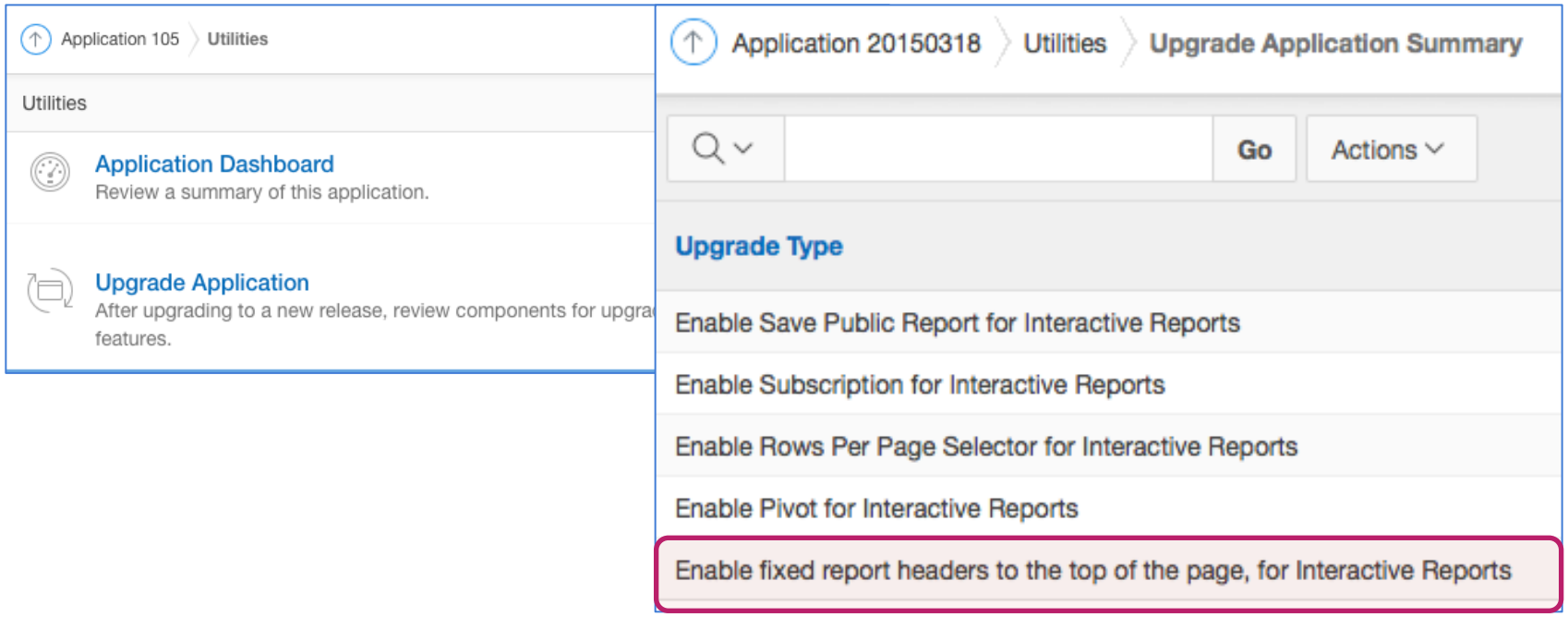
	APEX_VIEW_NAME	COMMENTS
1	APEX_APPL_CONCATENATED_FILES	Concatenated files of a user interface
2	APEX_APPLICATION_STATIC_FILES	Stores the files like CSS, images, javascript files, ... of an application.
3	APEX_APPLICATION_THEME_FILES	Stores the files like CSS, images, javascript files, ... of a theme.
4	APEX_APPL_PLUGIN_FILES	Stores the files like CSS, images, javascript files, ... of a plug-in.
5	APEX_WORKSPACE_FILES	Identifies uploaded files belonging to the workspace in the modplsql or EPG documents table
6	APEX_WORKSPACE_STATIC_FILES	Stores the files like CSS, images, javascript files, ... of a workspace.





Upgrade Assistant

- Using the upgrade assistant
  - Application > Utilities > Upgrade Application



  - Change the tree implementation