**Integration of Oracle Apex and E-Business Suite R12**

This page is a compilation of the blogs I wrote summer 2014 about this subject.  
  
[**1. Need for change**](http://apex4ebs.blogspot.com/p/integration-of-oracle-apex-and-e.html#par01)  
[**2. Installation / Activation of Apex in E-Business Suite R12**](http://apex4ebs.blogspot.com/p/integration-of-oracle-apex-and-e.html#par02)  
[**3. Authentication, Authorization and more Shared Components**](http://apex4ebs.blogspot.com/p/integration-of-oracle-apex-and-e.html#par03)  
[**4. Make an Apex page look like an Oracle EBS R12 page**](http://apex4ebs.blogspot.com/p/integration-of-oracle-apex-and-e.html#par04)  
[**5. Upload a file in Apex and submit concurrent request in EBS R12**](http://apex4ebs.blogspot.com/p/integration-of-oracle-apex-and-e.html#par05)  
[**6. Upload a file in Apex and write it to the UNIX filesystem**](http://apex4ebs.blogspot.com/p/integration-of-oracle-apex-and-e.html#par06)  
[**7. Download a file stored in a blob field from an Apex page**](http://apex4ebs.blogspot.com/p/integration-of-oracle-apex-and-e.html#par07)

**1. Need for change**

When upgrading an E-Business Suite implementation from R11i to R12 we ran into the problem that some pages written in HTML PL/SQL (htp package) and made accessible via functions of function type: 'SSWA plsql function that opens a new window (Kiosk Mode)' were no longer working in R12 (however the system still accepted this funcion type). So, we had to look for an alternative:

* (Re)use the generic Upload / Download functions introduced in R12 (System Administrator > Generic File Manager);
* Build static HTML pages and put this file under $OA\_HTML directory (Function Type : 'JSP Interoperable with OA' and HTML Call: name of the file);
* Build custom pages using OAF;
* Integrate Oracle Application Express (Apex) and develop the same functionality.

It wouldn't be interesting for this blog if we had not opted for the last option :-)

**2. Installation / Activation of Apex in E-Business Suite R12**

See Oracle Support Document 1306563.1 [Extending Oracle E-Business Suite Release 12 using Oracle Application Express (Apex)](https://support.oracle.com/epmos/faces/DocumentDisplay?id=1306563.1) for a white paper and a few scripts to get the integration of Application Express and E-Business Suite R12 working, including a demo application. In short:

* Install Application Express in the Oracle E-Business Suite database. Download the latest version of Application Express from [download.oracle.com](http://www.oracle.com/technetwork/developer-tools/apex/downloads/index.html) (Developer Tools) and follow the [instructions](http://docs.oracle.com/cd/E37097_01/doc/install.42/e35123/otn_install.htm#BABJJAGF).
* Install Oracle REST Data Services (former Apex listener) as Web Listener
  + Preferred by Oracle over Oracle HTTP Server and Embedded PL/SQL Gateway
  + Works standalone; Light use: no application server necessary (Web Logic, Glassfish)
  + Also see the article [Moving to the Apex Listener](https://www.ogh.nl/artikel.aspx?id=244) by Dimitri Gielis
* Log into Oracle Application Express Administration
  + Create Workspace
    - Workspace Name: APEX\_EBS
    - Schema Name: APEX\_EBS
  + Set security setting 'Allow Public Upload' (Manage Instance)
* Apply interoperability patch 12316083

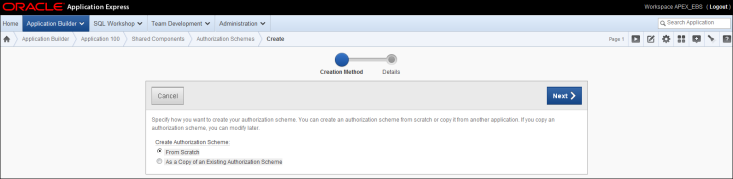
Next, build your application in Apex: it's all about rights !  
  
Our solution has been built using the following releases / versions:  
- Oracle E-Business Suite: release 12.1.3  
- Oracle Application Express: release 4.2.4.00.08  
- Oracle Database: version 11.2.0.4.0  
- Oracle Rest Data Services: version 2.0.7.91.15.01

**3. Authentication, Authorization and more Shared Components**

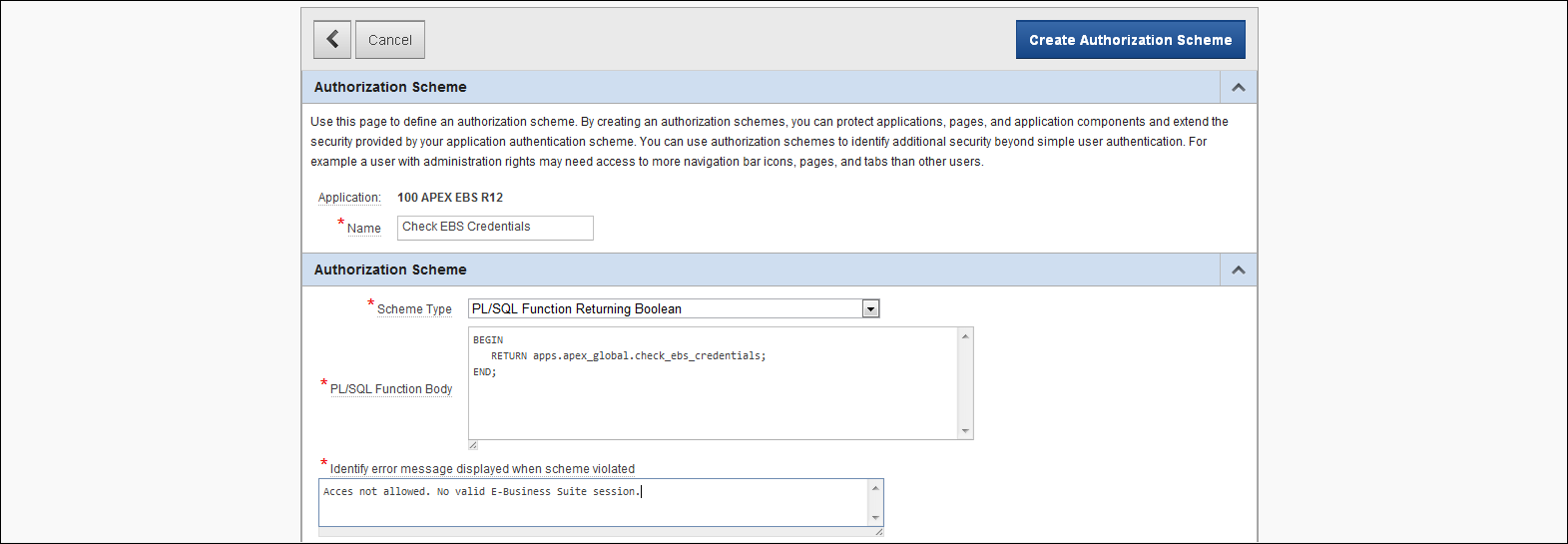
So far I have not seen any customer who uses Oracle Single Sign-on (OSSO) or Oracle Access Manager (OAM) collaborating with the Oracle E-Business Suite. So, I could not take advantage of the pre-configured Authentication Schemes in Oracle Application Express. In Oracle's White Paper a solution is described when using custom authentication for Oracle E-Business Suite. However, that solution is more suitable for a stand-alone application where you can login with your Oracle E-Business Suite credentials. For our problem, we are looking for a fully integrated application and therefor I developed another solution using the cookie settings in the icx\_sessions table of Oracle E-Business Suite. Moreover, when checking the session variables I store additional relevant Oracle E-Business Suite session information in Oracle Application Express Application Items ...  
  
**Application Items** (Shared Components > Logic)  
Scope: Application  
Session State Protection: Restricted - May not be set from browser

|  |  |
| --- | --- |
| Application Item Name | Comments |
| EBS\_USER\_ID | Key to User; To check EBS authorization and to set EBS context (icx\_sessions) |
| EBS\_RESP\_ID | Key to Responsibility; To check EBS authorization and to set EBS context (icx\_sessions) |
| EBS\_RESP\_APPL\_ID | Key to Responsibility Application; To check EBS authorization and to set EBS context (icx\_sessions) |
| EBS\_SEC\_GROUP\_ID | Key to Security Group; To check EBS authorization and to set EBS context (icx\_sessions) |
| EBS\_TIME\_OUT | Session Time Out in Oracle E-Business Suite (icx\_sessions) |
| EBS\_URL | URL to return to EBS Homepage from Apex (icx\_session\_attributes) |
| EBS\_ORG\_ID | EBS ORG\_ID (icx\_sessions) - MO: Operating Unit from Responsibility |
| EBS\_APPLICATION\_NAME | To be displayed at the left tophand corner (application\_name from fnd\_application\_tl using EBS\_RESP\_APPL\_ID) |

The Application Items are used in queries or when setting the 'environment' (apps\_initialize).  
  
**Authentication** (Shared Components > Security)  
The Oracle Application Express pages are directly launched from the E-Business Suite. Additional login is not desirable, so no Authentication Scheme.  
  
**Authorization**(Shared Components > Security)  
I created an Authorization Scheme 'Check EBS Credentials' that will check whether the user has a valid E-Business Suite session. If so, then session attributes are copied into the Application Items. If not, then an error message will be displayed that access is not allowed. The E-Business Suite function icx\_sec.getsessioncookie is used to determine the session\_id. This session\_id is the key to retrieve additional information from the E-Business Suite tables icx\_sessions and icx\_session\_attributes.   
  
Authorization Schemes: Create> (B)

[](http://3.bp.blogspot.com/-beGv91u1Zxw/U74ygaEke7I/AAAAAAAAAgw/QYZurAvpyLc/s1600/AuthorizationScheme1.png)

Next> (B)

[](http://1.bp.blogspot.com/-xlxtNOYhsp0/U74yjFZBwkI/AAAAAAAAAg4/CLtmoCTK4TY/s1600/AuthorizationScheme2.png)

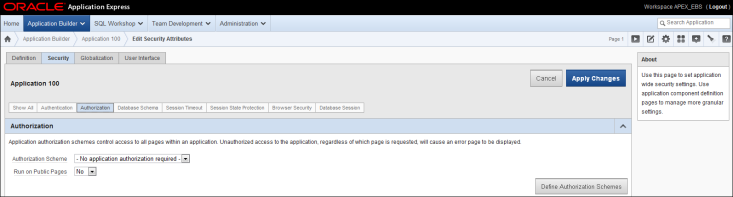
Name: Check EBS Credentials   
Scheme Type: PL/SQL Function Returning Boolean  
PL/SQL Function Body:

BEGIN

RETURN apps.apex\_global.check\_ebs\_credentials;

END;

Error message displayed when scheme violated: "Access not allowed: No valid E-Business Suite session."  
Evaluation point: once per page view  
Create Authorizartion Scheme (B)  
  
See the [code for package apex\_global](http://apex4ebs.blogspot.nl/p/code-apexglobal-package.html) for the code of the function check\_ebs\_credentials  
  
Setting the session timeout seems not to work. After a while the Oracle E-Business Suite session shows a timeout, but if you did not close the Apex page to upload a file, it still lets you upload and submit a file ... Is this a bug?  
  
**Security Attributes** (Shared Components > Security)  
Access to any page in the Apex application is not allowed when no E-Business Suite session is active. This is arranged by setting the Authorization Scheme as a Security Attribute. However, it is also possible to manage authorization per page. In the latter case don't set the authorization scheme as shared component.   
  
Security > Security Attributes: Authorization

[](http://4.bp.blogspot.com/-V7myZGI0pbw/U74yuRTTVuI/AAAAAAAAAhA/XLNTNtpcmQA/s1600/SecurityAttributes.png)

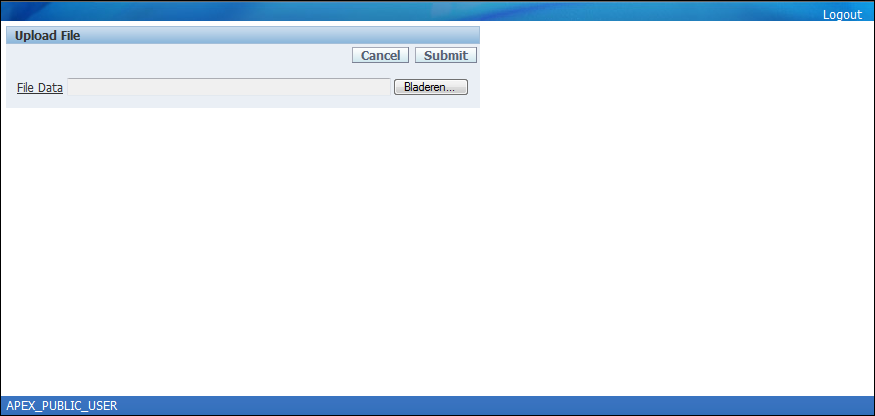
Authorization Scheme: Check EBS Credentials  
Apply Changes (B)  
  
PS: In an attempt to get the session timeout working, I also tried to set the Maximum Session Idle Time In Seconds to 1800 (default value E-Business Suite). This is also a Security Attribute: Session Timeout. I read somewhere that this was a a condition to get the session timeout working. Unfortunately, it didn't help. Besides setting the Maximum Session Idle Time In Seconds here at application level it was initially already set at instance level by the dba (Oracle Application Express Administration).

**4. Make an Apex page look like an Oracle EBS R12 page**

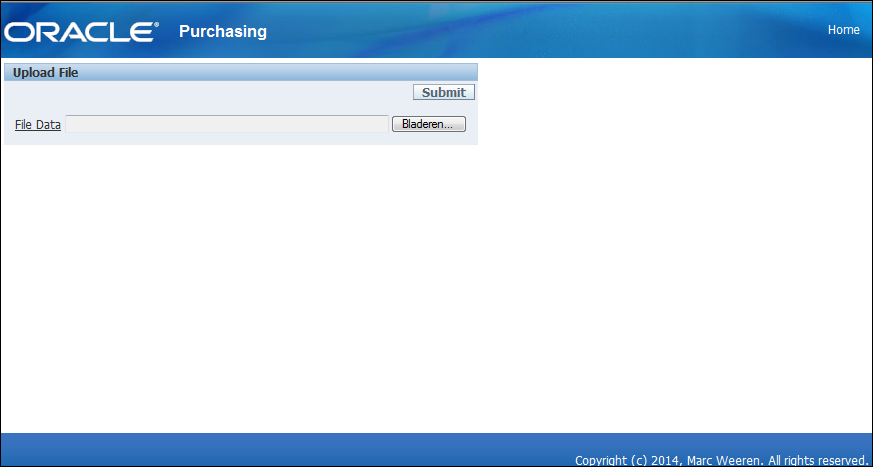
I made the folowing changes to let the Apex page more resemble an E-Business Suite R12 page:

* First of all, I created several **Application Items** (Shared Components > Logic) including EBS\_APPLICATION\_NAME and EBS\_URL. These will be set in a routine check\_ebs\_credentials that will run when a page is launched (part of the Authorization Scheme I explained earlier).
* For the User Interface select Apex **Theme** 20 (Traditional Blue), a standard theme that comes along with Apex (one of the Legacy Themes)
* **Logo** (Shared Components > User Interface > User Interface Attributes)  
  The Oracle Logo displayed in Oracle EBS R12 can also be displayed in the Apex application:
  + First copy the file FNDSSCORP.gif $OA\_MEDIA (EBS) to the directory 'images' in Apex
  + Logo Type: Image
  + Logo: /i/FNDSSCORP.gif
  + Logo attributes: alt="Oracle Logo" title="Oracle Logo" width="155" heigth="20" border="0"
* **Navigation Bar Entries** (Shared Components > Navigation)  
  In the E-Business Suite R12, self-service pages (HTML pages) like the Apex pages can be called from a menu in forms (java program) or a menu in the browser (HTML page). In the first case the HTML page is opened in a new browser window and when done the user has to close that window. In the latter case the HTML page appears in the same browser window and when done the user can navigate back to the menu. This is a bit of a problem for navigation. In the normal E-Business Suite self-service pages (using OAF) this is managed correctly: in the navigation bar you will see 'Close window' or 'Home'. Because it seems not to be possible to determine how the Apex page is launched, only the 'Home' entry has been provided. For closing a window, all browsers have built-in possibilities.
  + Remove Logout
  + Add Home. For 'URL Target' enter &EBS\_URL. (don't forget the dot)  
    EBS\_URL holds the value icx\_session\_attribute '\_USERORSSWAPORTALURL' but can also be composed of the value of profile option APPS\_SERVLET\_AGENT concatenated with /OA.jsp?OAFunc=OAHOMEPAGE#
* **Template** (Shared Components > Templates > Type: Page, Name: No Tabs).   
  In the application I only make use of pages that use the page template named 'No tab'. The application name is added to the page header (however defined in the page body) and the user name has been removed from the footer as well as a copyright text was added to the bottom right hand corner. I couldn't find how to set the built-in application substitution strings, so I made changes to the templates, however setting of the built-in application substitution strings would result in a more generic solution.
  + Definition > Body:
    - The application name is added stored in the Application Item EBS\_APPLICATION\_NAME.
    - The style attribute is necessary to display the name with the correct format. Image blank.gif is used to force some space between the logo and the application name.
    - Also two blank lines are added (one above the logo; one beneath the logo):  
        
      <div id="t20PageHeader">  
      <table border="0" cellpadding="0" cellspacing="0" summary="">  
      <tr><td colspan=3> </td></tr>  
      <tr>  
      <td id="t20Logo" valign="top">#LOGO#<br />#REGION\_POSITION\_06#</td>  
      <td id="t20HeaderMiddle" valign="top" width="100%"><img src="/i/blank.gif" width="20" border="0"><span title="EBS Application Name" style="font-family:Arial; color:#FFFFFF; font-size:16px; white-space:nowrap; font-weight:bold; vertical-align:top;">&EBS\_APPLICATION\_NAME.</span> #REGION\_POSITION\_07#<br /></td>  
      <td id="t20NavBar" valign="top">#NAVIGATION\_BAR#<br />#REGION\_POSITION\_08#</td>  
      </tr>  
      <tr><td colspan=3> </td></tr>  
      </table>  
      </div>  
      <div id="t20BreadCrumbsLeft">#REGION\_POSITION\_01#</div>  
      <table border="0" cellpadding="0" cellspacing="0" summary="" id="t20PageBody" width="100%" height="70%">  
      <td width="100%" valign="top" height="100%" id="t20ContentBody">  
      <div id="t20Messages">#GLOBAL\_NOTIFICATION##SUCCESS\_MESSAGE##NOTIFICATION\_MESSAGE#</div>  
      <div id="t20ContentMiddle">#BOX\_BODY##REGION\_POSITION\_02##REGION\_POSITION\_04#</div>  
      </td>  
      <td valign="top" width="200" id="t20ContentRight">#REGION\_POSITION\_03#<br /></td>  
      </tr>  
      </table>
  + Definition > Footer
    - Added an extra line to look it more like EBS R12.
    - Also a copyright text is added like in EBS R12.
    - The username has been deleted from the footer (&APP\_USER.):  
        
      <table border="0" cellpadding="0" cellspacing="0" summary="" id="t20PageFooter" width="100%">  
      <tr><td colspan=3> </td></tr>  
      <tr>  
      <td id="t20Left" valign="top"><span id="t20UserPrompt"></span><br /></td>  
      <td id="t20Center" valign="top">#REGION\_POSITION\_05#</td>  
      <td id="t20Right" valign="top"><span id="t20Customize">#CUSTOMIZE#Copyright (c) 2014, Marc Weeren. All rights reserved.</span><br /></td>  
      </tr>  
      </table>  
      <br class="t20Break"/>  
      #FORM\_CLOSE#   
      #DEVELOPER\_TOOLBAR#  
      #GENERATED\_CSS#  
      #GENERATED\_JAVASCRIPT#  
      </body>  
      </html>

Before:

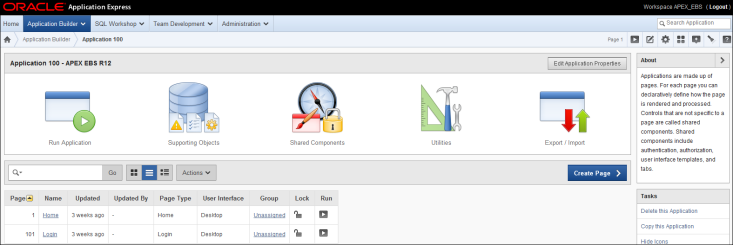
[](http://2.bp.blogspot.com/-Pc1FIAP3I2E/U7v_6hXczzI/AAAAAAAAAfs/jfJoIxaZ6F4/s1600/Upload+Files+before.PNG)

After:

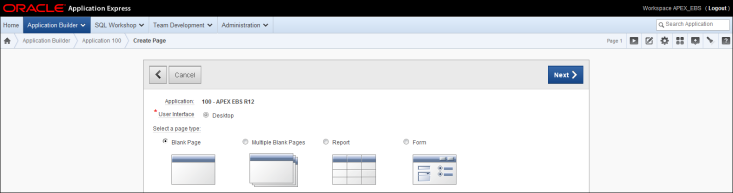
[](http://1.bp.blogspot.com/-vIi8wFCVGX4/U7v_-6oLTiI/AAAAAAAAAf0/6syDA3fyoyY/s1600/Upload+Files+after.PNG)

**5. Upload a file in Apex and submit concurrent request in EBS R12**

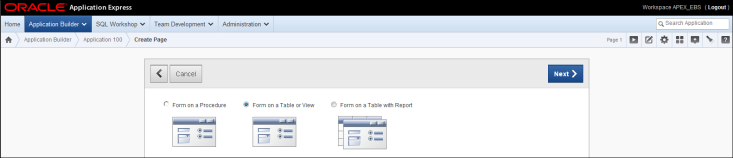
One of the pages I built in Application Express (Apex) as an extension to the Oracle E-Business Suite R12 (EBS R12) is having the possibility to upload an Excel file into a blob field and subsequently submit a concurrent process (batch job) in EBS R12 to process the Excel file. Although interesting too, in this post I won't bother you with how to process an Excel file in PL/SQL / Java. I will show how I managed solving the challenge with Apex, EBS R12 and the underlying database and moreover I will explain how I overcame some problems I encountered. Some special points for attention:   
- Validation of the filename   
- Success message   
- Submit button   
- Submit concurrent request in Oracle EBS   
- Set 'environment' in submitted process  
  
**Validation of the filename**   
I had a look at the sample application 'Sample File Upload and Download' that comes along with Apex (see tab Packaged Applications on the Application Builder page). I noticed that validation of the filename is done via a database trigger (before insert). In case of an error a raise\_application\_error is done, resulting in a not so user friendly error messages. Due to the way Apex handles a field that is displayed as 'File Browse...' it is not possible to catch the filename in an earlier stage. However, it should be possible to solve this problem using javascript, but I am not a fan of such a contrived solution (my opinion).  
  
**Success message**   
In Apex it seems not to be possible to include a variabele in a Process Success Message that can be entered on Process definition page. After the concurrent process has been submitted in EBS R12 the user wants to see the request number, necessary to monitor the progress of the request and to view the request log in Oracle EBS. This is solved by calling the Apex routine g\_print\_success\_message that can handle a composed string containing the request number.   
  
**Submit button**   
When testing the page, the Submit button disappeared after an error in the filename. This error is set by a before insert database trigger (so after the regular validations / exception to the normal flow Apex works). When the page has been created via the create page wizzard (normal way of working) then default the Condition at the Button definition page is set to check whether FILE\_ID is NULL. However, the trigger that raises the error runs after the process that retrieves the primary key / fills the FILE\_ID (Process 'Get PK'). When returning on the page the Submit button has disappeared because now FILE\_ID has a value. The solution is to change the Condition Type at the Button definition page to 'Button Not Conditional'.  
  
**Submit concurrent request in Oracle EBS R12**  
After the Excel file has been inserted in the database, we want to start a batch job inside EBS R12 that processes the file. This custom process was already developed in the Oracle E-Business R11i implementation, but it is interesting what changes we had to make in order to get it working when submitted from Apex. Besides passing the file identification (little change), now we had to add some extra parameters to the concurrent program (also see next paragraph about setting the environment in the submitted process).   
In the Page Processing lane of the Apex page in hand, we added a Process 'Start concurrent request' after the (standard) Process 'Process Row of ....'. In this process an anonymous block (of PL/SQL code) is entered in 'Source'. Submitting the concurrent process could be achieved by calling the standard Oracle EBS routine fnd\_request.submit\_request. It turned out that in order to get the concurrent program actually running we had to run apps\_initialize first.  
However, it is not allowed to call the standard Oracle routines straight. Apex has no permission to do so and for security reasons it is not preferred to solve this by granting execute rights to APEX\_EBS on those routines. As a central point of access we created in the APPS schema (standard Oracle EBS user, that can access all data / all routines) a package called apex\_global (as in Oracle's [white paper](https://support.oracle.com/epmos/faces/DocumentDisplay?id=1306563.1)). It is defined with an authid clause of AUTHID DEFINER, which means that each routine from this package will be run with the owners rights (APPS) as opposed to invokers rights (current user which is APEX\_EBS). Of course APEX\_EBS must be granted execute rights to this package. Also see the [code for package apex\_global](http://apex4ebs.blogspot.nl/p/code-apexglobal-package.html).  
  
So in the anonymous block on the Process definition (source) you will see a call to a routine in the apps.apex\_global package and in the apps.apex\_global package you will see the call to the standard Oracle EBS routine with exactly the same parameters.  
  
**Set 'environment' in submitted process**   
A concurrent process submitted from outside Oracle EBS means that all context has been lost. Normally, when a concurrent request is submitted in Oracle EBS, Oracle knows what the user is allowed to do / what data the user is allowed to see. This is managed by some settings.These settings can also be achieved by running the standard Oracle EBS routines fnd\_global.apps\_initialize and depending on the MOAC setting (multiple organization access control) also mo\_global.init or mo\_global.set\_policy\_context (for the exact syntax see further down this post).  
Consequence is that we have to pass the necessary information to the concurrent process. Fortunattely all this information is retrieved from the session information we captured when the Apex page was launched by Oracle EBS (see my post 'Authentication, Authorization and more Shared Components'). This more or less confidential information is passed to the concurrent program via parameters.  
  
**Creating the page step by step**  
  
[A. Table and sequence definition](http://apex4ebs.blogspot.com/p/integration-of-oracle-apex-and-e.html#par05.1)  
[B. Define APPS package apex\_global](http://apex4ebs.blogspot.com/p/integration-of-oracle-apex-and-e.html#par05.2)  
[C. Page definition](http://apex4ebs.blogspot.com/p/integration-of-oracle-apex-and-e.html#par05.3)  
[D. Definition of concurrent program in Oracle EBS R12](http://apex4ebs.blogspot.com/p/integration-of-oracle-apex-and-e.html#par05.4)  
[E. Oracle EBS routine submitted in Apex](http://apex4ebs.blogspot.com/p/integration-of-oracle-apex-and-e.html#par05.5)  
[F. Add function to EBS R12](http://apex4ebs.blogspot.com/p/integration-of-oracle-apex-and-e.html#par05.6)  
  
**A. Table and sequence definition**  
Notice that the before insert or update trigger on apex\_ebs.xxapx\_files is not only for the validation of the filename, but also fills the audit information (created\_by, creation\_date, etc) and moreover the org\_id (company) that is copied from the Application Items. Click on the link to see the [database objects creation scripts](http://apex4ebs.blogspot.nl/p/database-objects-creation-scripts.html).  
  
**C. Define APPS package apex\_global**   
Create the package apex\_global as database user APPS (not possible in the SQL Workshop in Apex):   
Click on the link to see the [code of package apex\_global](http://apex4ebs.blogspot.nl/p/code-apexglobal-package.html)  
  
**B. Page definition**  
Most important part is the source in the process 'Start concurrent Process', a PL/SQL anonymous block that submits a concurrent process in EBS R12 and shows the request number assigned.  
  
Application Builder

[](http://3.bp.blogspot.com/-cIKCIiu67qM/U8fI8bUsTEI/AAAAAAAAAig/Pvi9KAFMcbY/s1600/upload1.PNG)

Create Page >

[](http://3.bp.blogspot.com/-0ymWmK1lRYQ/U8fI8VsXKbI/AAAAAAAAAio/-g5xWrzWAyI/s1600/upload2.PNG)

Select a Page Type: Form  
Next >

[](http://4.bp.blogspot.com/-Ah8ftfI7_Vg/U8fI8TzOXeI/AAAAAAAAAik/dT_SdOecRXM/s1600/upload3.PNG)

Form on a Table or View  
Next>

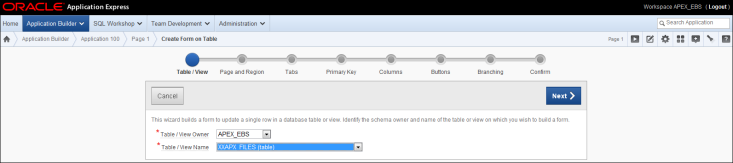
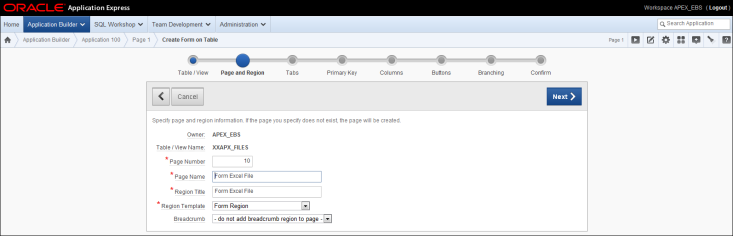
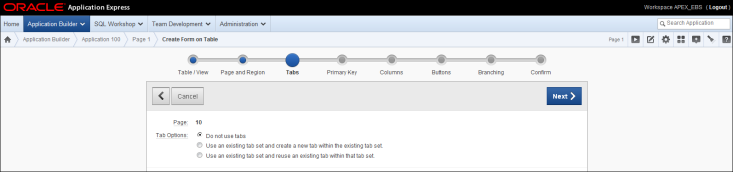
[](http://2.bp.blogspot.com/-AKabuR_fQ98/U8fI9GNpTPI/AAAAAAAAAi0/DCeZn5XafYo/s1600/upload4.PNG)

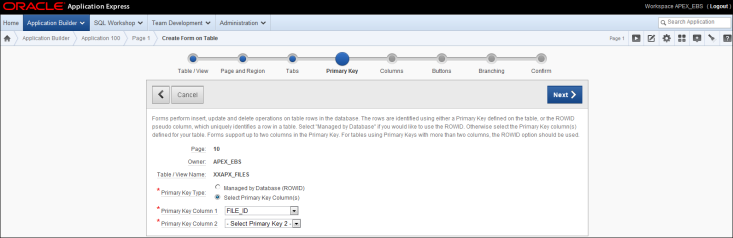
Table / View Owner: APEX\_EBS (default)  
Table / View Name: XXAPX\_FILES  
Next >

[](http://4.bp.blogspot.com/--q7fmHuFT2s/U8fSVKCtK9I/AAAAAAAAAj0/SqTEAC0zv1M/s1600/upload5.PNG)

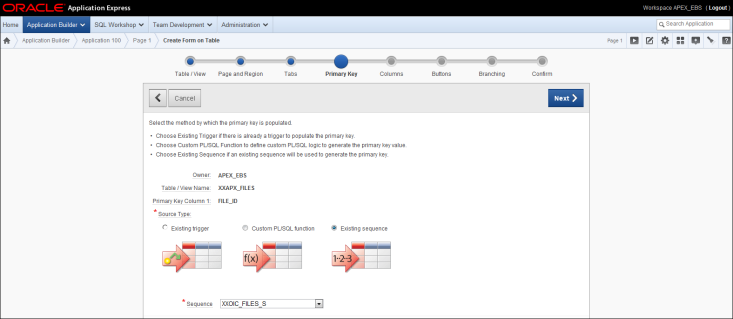
Page Number: 10  
Page Name: Upload Excel File  
Page Title: Upload Excel File  
Region Template: Form Region (default)  
Breadcrumb: do not add breadcrumb region to page (default)  
Next >

[](http://2.bp.blogspot.com/-Tl40Jjpvql4/U8fSVJzjqaI/AAAAAAAAAjs/hhv0fmErKpA/s1600/upload6.PNG)

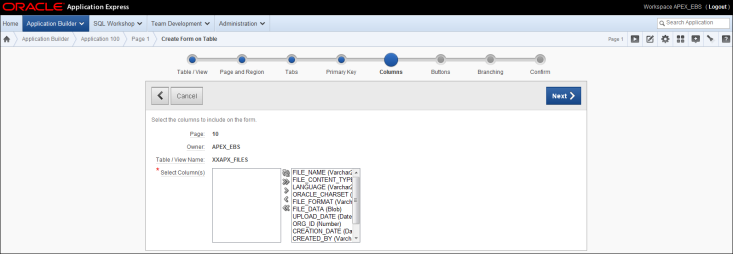
Do not use tabs  
Next >

[](http://1.bp.blogspot.com/-uAzxJ-4hDqg/U8fSVDjKh7I/AAAAAAAAAjw/aFGmZgx4xnE/s1600/upload7.PNG)

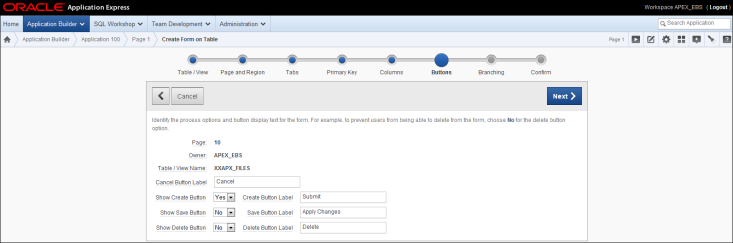
Primary Key Type: Select Primary Key Column(s)  
Primary Key Column 1: FILE\_ID (defaulted)  
Next >

[](http://4.bp.blogspot.com/-64d4T3pPYn8/U8fSVhwq_wI/AAAAAAAAAj4/efbUUopfFVg/s1600/upload8.PNG)

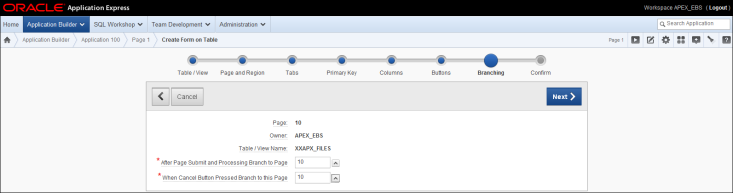
Source Type: Existing sequence  
Sequence: XXOIC\_FILES\_RATE\_TABLE\_S  
Next >

[](http://1.bp.blogspot.com/-47XHo_wpMg0/U8fTT7j8h6I/AAAAAAAAAkU/n8gV6oNU4vQ/s1600/upload9.PNG)

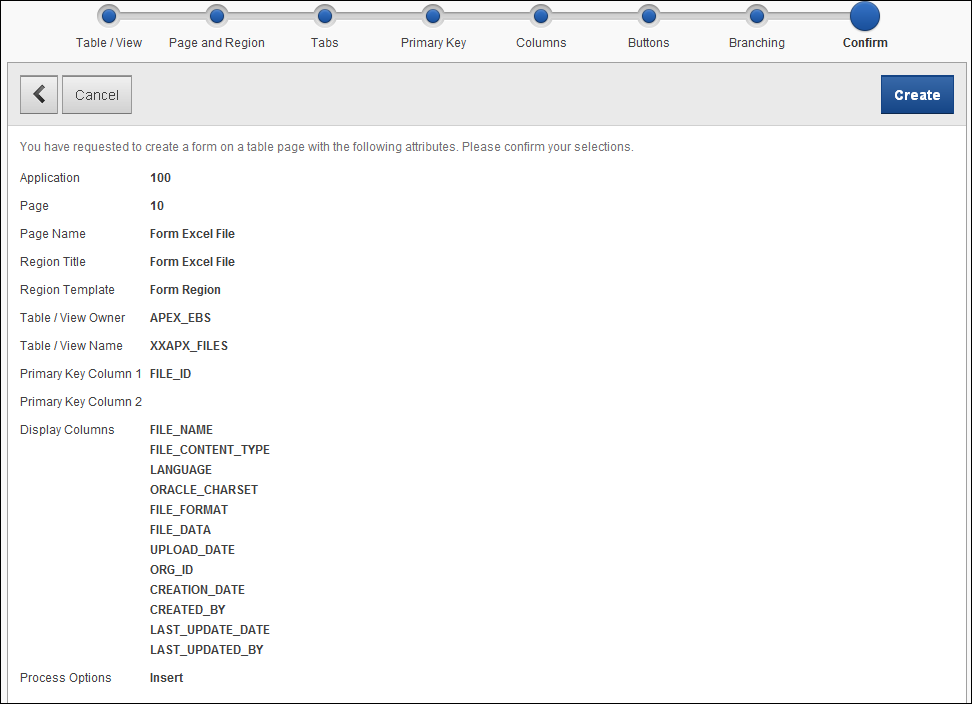
Select Column(s): all (default; Will hide all columns after all, except FILE\_DATA)  
Next >

[](http://2.bp.blogspot.com/-XRESRI3ywqg/U8fTT23yKnI/AAAAAAAAAkQ/clTrvvPKkWI/s1600/upload10.PNG)

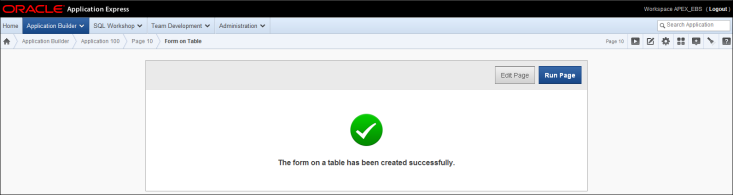
Changed from default:  
Create Button Label: Submit  
Show Save Button: No  
Show Delete Button: No  
Next>

[](http://2.bp.blogspot.com/-JxIHiCJLowU/U8fUwauoOYI/AAAAAAAAAko/A0GI2kFwL74/s1600/upload11.PNG)

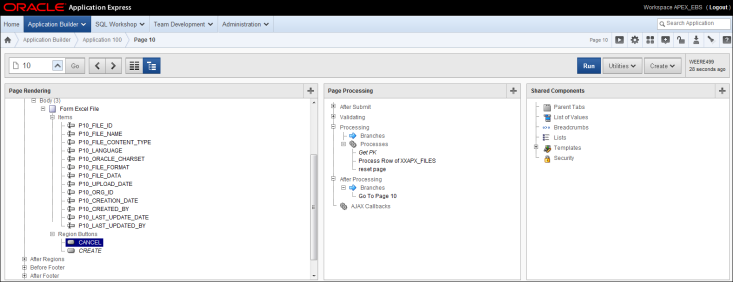
Branching: stay on the page after Page Submit and Cancel (cancel button will de removed later on)  
After Page Submit and Processing Branche to Page: 10  
After Cancel Button Pressed Branche to Page: 10  
Next >

[](http://3.bp.blogspot.com/-lm4JA4fi5Vk/U8fUwd08FXI/AAAAAAAAAkk/RN8V7EfHQ2M/s1600/upload12.PNG)

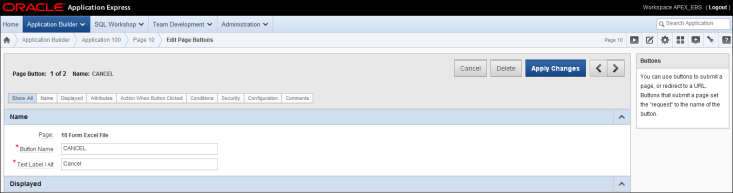
Create >

[](http://2.bp.blogspot.com/-NZwpQVt3_tk/U8fUwSR8r6I/AAAAAAAAAks/agqdBvrHLCE/s1600/upload13.PNG)

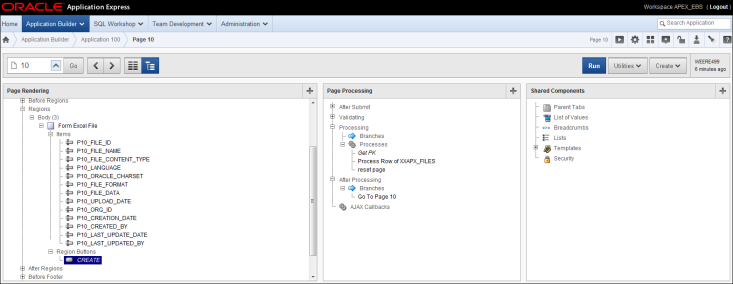
Edit Page >

[](http://2.bp.blogspot.com/-cymYDTUNGPo/U8fUxLyXXjI/AAAAAAAAAk8/gueQGKkdnm4/s1600/upload14.PNG)

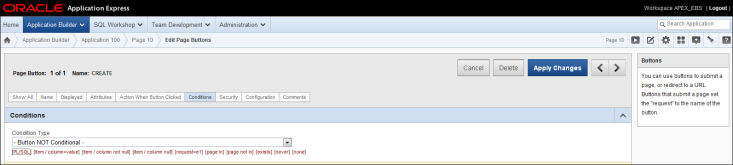
Page Rendering: Upload Rate Table > Regions > Body > Upload Rate Table > Region Buttons  
Double Click Region Button CANCEL (or choose Edit after right mousclick)

[](http://1.bp.blogspot.com/-9bKdrgRBwLM/U8fUxLEnOKI/AAAAAAAAAk0/OHzhuNlSdC8/s1600/upload15.PNG)

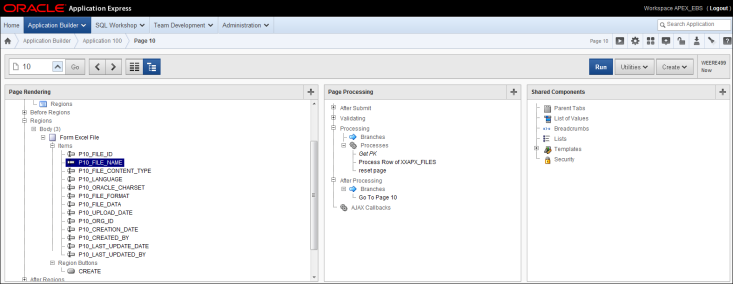
Press Delete and confirm delete action.

[](http://2.bp.blogspot.com/-r_GYgvKkV7g/U8fWxUD5reI/AAAAAAAAAlU/doVfk3bD4Rc/s1600/upload16.PNG)

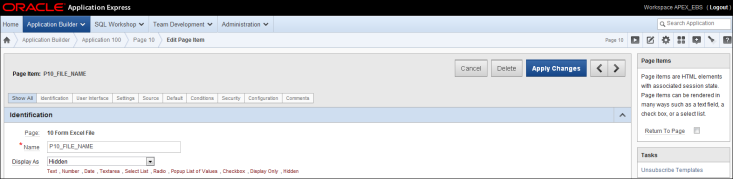
Page Rendering: Upload Rate Table > Regions > Body > Upload Rate Table > Region Buttons  
Double Click Region Button CREATE (or choose Edit after right mouseclick)

[](http://2.bp.blogspot.com/-iT2RO7QZme4/U8fWxbsq0-I/AAAAAAAAAlQ/aaIdvAUXYsU/s1600/upload17.PNG)

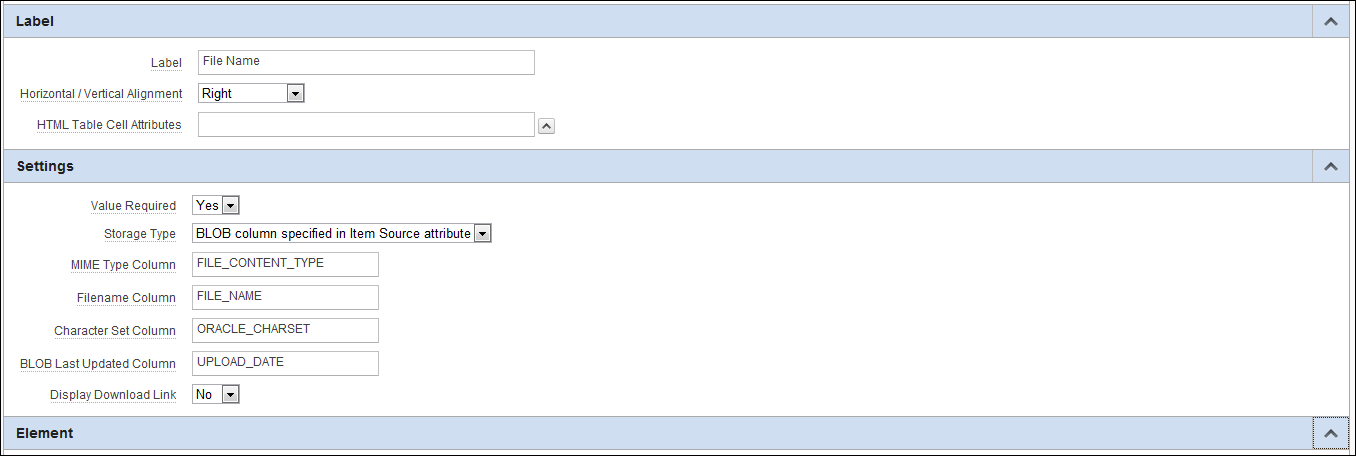
Default there is a check whether FILE\_ID is NULL. This causes the 'Submit' button to disappear in case an error is raised by a database trigger (where validation on FILE\_NAME takes place). In that case FILE\_ID already has a value.  
Condition Type: Button Not Conditional  
Apply Changes

[](http://4.bp.blogspot.com/-KXDeaY9EuqU/U8fWxRxeu5I/AAAAAAAAAlY/GKjxEEeBXBE/s1600/upload18.PNG)

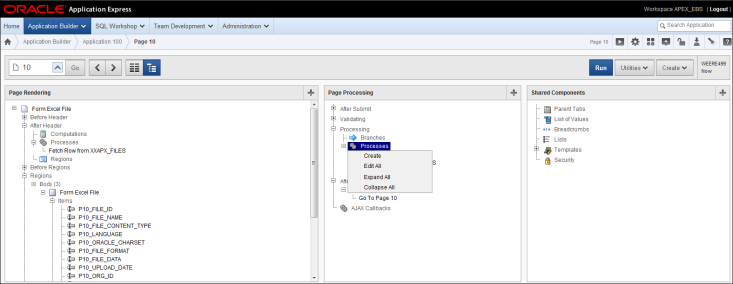
Page Rendering: Upload Rate Table > Regions > Body > Upload Rate Table > Items  
Doubleclick on P10\_FILE\_NAME

[](http://3.bp.blogspot.com/-rLL0X1LNZ-8/U8fWx2Ub9pI/AAAAAAAAAlo/Vuyu6phbIDo/s1600/upload19.PNG)

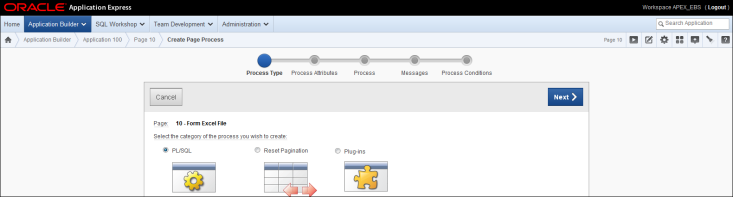
Display As: Hidden  
  
Do the same for the rest of the Page Items, except for Page Item named 'P10\_FILE\_DATA'.  
Browse to the next Page Item by pressing >  
After the last Page Item press Apply Changes

[](http://2.bp.blogspot.com/-nFNdRc6S4tA/U8fZ5hgRb1I/AAAAAAAAAl0/X_KVnyTQoaA/s1600/upload20.PNG)

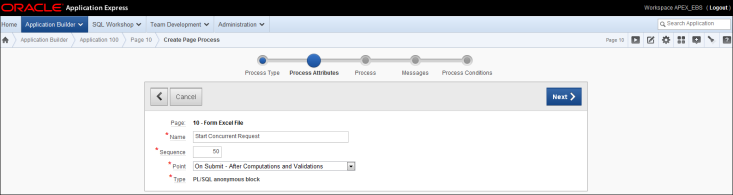
Values for Page Item 'P10\_FILE\_DATA':  
  
Label :  
Label: File Name  
Settings:  
Required: Yes  
MIME Type Column: FILE\_CONTENT\_TYPE  
Filename column: FILE\_NAME  
Character Set Column: ORACLE\_CHARSET  
BLOB last updated Column: UPLOAD\_DATE  
Display Download Link: No

[](http://3.bp.blogspot.com/-I5qWMeN1Tw4/U8fZ5j0MDjI/AAAAAAAAAl4/OYYIWmcjxNY/s1600/upload21.PNG)

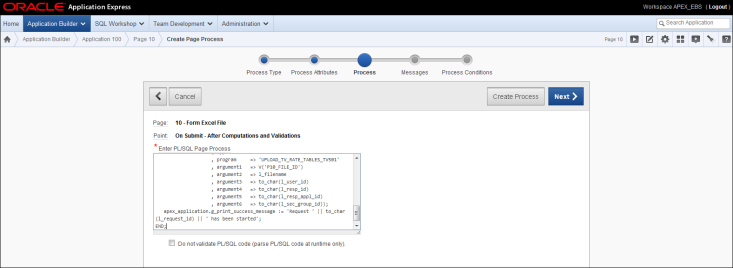
Page Processing: Processing > Processes  
Right mouseclick and click on Create

[](http://4.bp.blogspot.com/-Qz1Cywyeg9E/U8fZ5pr9leI/AAAAAAAAAmk/dueb4tolDVY/s1600/upload22.PNG)

Process Type: PL/SQL

[](http://1.bp.blogspot.com/-GWbdueDkq5w/U8fZ6WDlMrI/AAAAAAAAAl8/gRMpj6YTPRc/s1600/upload23.PNG)

Process Attributes:  
Name: Start Concurrent Request  
Sequence: 50 (defaulted as latest process; Will renumber process 'Reset Page' next)  
Point: On Submit – After Computations and Validations (defaulted)  
Next >

[](http://2.bp.blogspot.com/-OtDyZYFYEtk/U8fZ6uBMrRI/AAAAAAAAAmA/ggTscaShjew/s1600/upload24.PNG)

PL/SQL anonymous block that submits a concurrent process in Oracle E-Business Suite and show the request number assigned. In Apex it seems not to be possible to include a variabele in a Success message. Therefore we make a call to the Apex routine apex\_application.g\_print\_success\_message.

DECLARE

CURSOR c\_filename (b\_file\_id NUMBER)

IS

SELECT file\_name

FROM xxapx\_files

WHERE 1=1

AND file\_id = b\_file\_id;

l\_request\_id NUMBER;

l\_filename VARCHAR2(100);

l\_user\_id NUMBER := TO\_NUMBER(V('EBS\_USER\_ID'));

l\_resp\_id NUMBER := TO\_NUMBER(V('EBS\_RESP\_ID'));

l\_resp\_appl\_id NUMBER := TO\_NUMBER(V('EBS\_RESP\_APPL\_ID'));

l\_sec\_group\_id NUMBER := TO\_NUMBER(V('EBS\_SEC\_GROUP\_ID'));

BEGIN

-- initialiseren Oracle Apps

apps.apex\_global.apps\_initialize(l\_user\_id,l\_resp\_id,l\_resp\_appl\_id,l\_sec\_group\_id);

OPEN c\_filename (TO\_NUMBER(V('P10\_FILE\_ID')));

FETCH c\_filename INTO l\_filename;

CLOSE c\_filename;

l\_request\_id := apps.apex\_global.submit\_request

( application => 'XXCUST'

, program => 'XXCUST\_PROCESS\_XLS\_FILE'

, argument1 => V('P10\_FILE\_ID')

, argument2 => l\_filename

, argument3 => to\_char(l\_user\_id)

, argument4 => to\_char(l\_resp\_id)

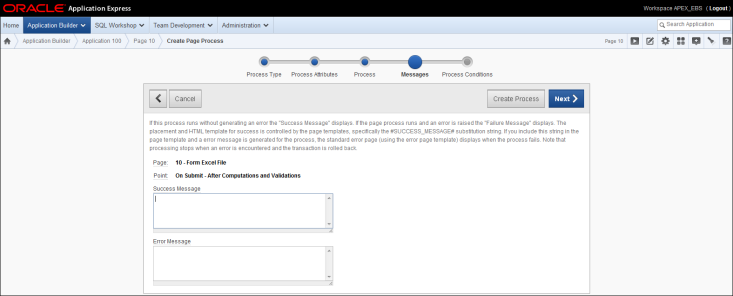
, argument5 => to\_char(l\_resp\_appl\_id)

, argument6 => to\_char(l\_sec\_group\_id));

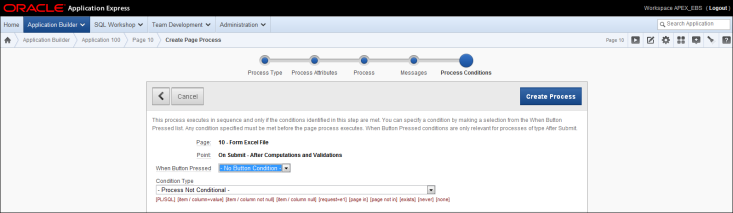
apex\_application.g\_print\_success\_message := 'Request ' || to\_char(l\_request\_id) || ' has been started';

END;

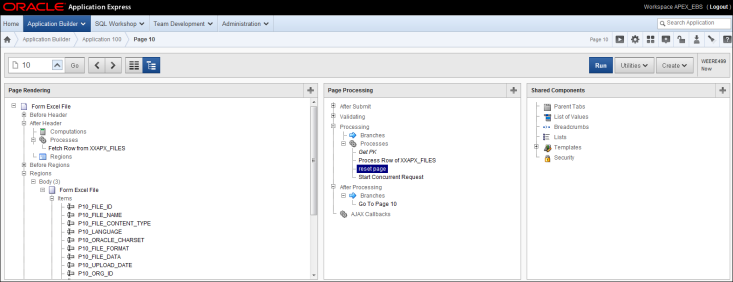
Next >

[](http://3.bp.blogspot.com/-HJ3SXJQQmQ8/U8fZ7G3wTII/AAAAAAAAAmE/8M3MRM3NkHA/s1600/upload25.PNG)

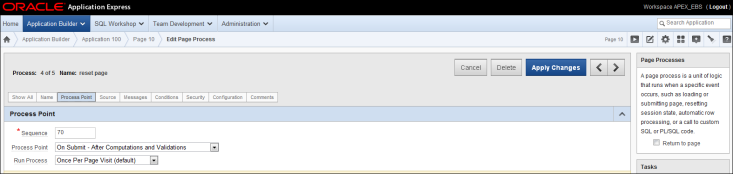
The Succes Message is managed by the PL/SQL code. No Error Message.  
Next >

[](http://4.bp.blogspot.com/-hLlYiFk94bQ/U8fZ7coYuUI/AAAAAAAAAmM/bEexFhyV8BI/s1600/upload26.PNG)

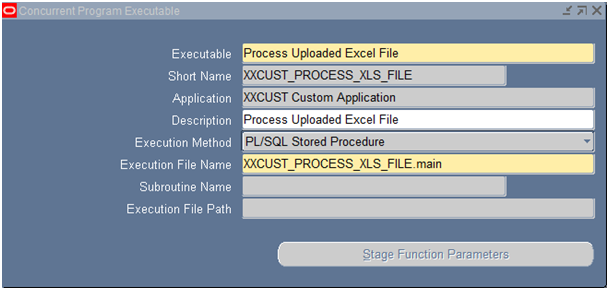
No conditions, just like the by the Apex wizard already created processes.  
Create Process

[](http://4.bp.blogspot.com/-MWsnyHeHP38/U8fcp38TafI/AAAAAAAAAm0/76HX375KUBc/s1600/upload27.PNG)

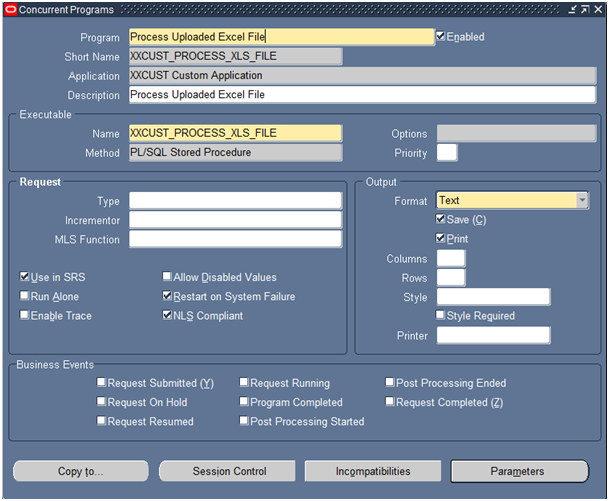
Page Processing: Processing > Processes  
Double click on proces 'reset page'

[](http://2.bp.blogspot.com/-X9p8tr_PJkk/U8fcp-KO7fI/AAAAAAAAAmw/4oQLorTr-Ac/s1600/upload28.PNG)

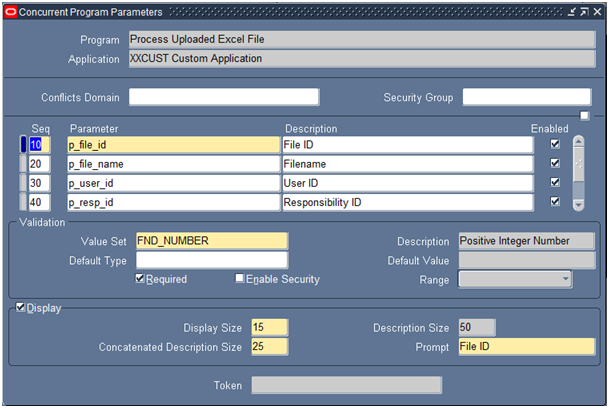
Process Point  
Sequence: 70  
Apply Changes  
  
**D. Definition of concurrent program in Oracle EBS R12**  
This is the definition of the concurrent program in Oracle EBS R12 that is submitted while processing the Apex page. This request wiil be submitted from outside EBS, so we need to pass the necessary parameters to be able to set the right context in the Oracle EBS executable by calling Oracle's procedure apps\_initialize. We will pass back the session information retrieved from icx\_session when the Apex page was launched (also see authorization scheme Check EBS Credentials posted earlier at [Authentication, Authorization and more Shared Components](http://apex4ebs.blogspot.nl/2014/07/authentication-authorization-and-more.html)).  
  
Path in EBS R12: System Administrator: Concurrent > Program > Executable

[](http://4.bp.blogspot.com/-yAElYY47oxs/U8e62cS9DMI/AAAAAAAAAiA/IILACpJpXWI/s1600/ConcurrentProgram3.png)

Executable: Process Uploaded Excel File  
Short Name: XXCUST\_PROCESS\_XLS\_FILE  
Application: XXCUST Custom Application  
Description: Process Uploaded Excel File  
Execution Method: PL/SQL Stored Procedure  
Execution File Name: XXCUST\_PROCESS\_XLS\_FILE.main  
  
Path in EBS R12: System Administrator: Concurrent > Program > Define

[](http://1.bp.blogspot.com/-_xDeCepgPB8/U8e66z41NYI/AAAAAAAAAiQ/PtSYuRvQlyM/s1600/ConcurrentProgram1.png)

Program: Process Uploaded Excel File  
Short Name: XXCUST\_PROCESS\_XLS\_FILE  
Application: XXCUST Custom Application  
Description: Process Uploaded Excel File  
  
Executable Name: Process Uploaded Excel File  
Executable Method: PL/SQL Stored Procedure  
  
Parameters (B)

[](http://4.bp.blogspot.com/-udegX77ZcVw/U8e65Fv8jaI/AAAAAAAAAiI/WryG-YGFtSs/s1600/ConcurrentProgram2.png)

Parameters:

|  |  |  |  |
| --- | --- | --- | --- |
| Seq | Parameter | Description + Prompt | Value Set |
| 10 | p\_file\_id | File ID | FND\_NUMBER |
| 20 | p\_file\_name | Filename | 240 Characters |
| 30 | p\_user\_id | User ID | FND\_NUMBER |
| 40 | p\_resp\_id | Responsibility ID | FND\_NUMBER |
| 50 | p\_resp\_appl\_id | Responsibility Application ID | FND\_NUMBER |
| 60 | p\_sec\_group\_id | Security Group ID | FND\_NUMBER |

**E. Oracle EBS routine submitted in Apex**  
Just an outline of the code.

* In Oracle EBS the output parmeters p\_errbuf and p\_retcode are mandatary for procedures that are called from a concurrent program / executable. When the p-errbuf returns 1 it will show yellow (warning) on the view request screen and when 2 is returned it will show red (error).
* Also notice that the first statements in the main procedure are to set the environment, otherwise it cannot select data from 'MOAC tables'. Running apps\_intiatialize is always necessary. Depending on the MOAC settings (Multi Organization Access Control) in Oracle EBS it is also necessary to run mo\_global.init or in our situation (MOAC is activated, but only single org) run mo\_global.set\_policy\_context with parameter 'S' (single) and the org\_id of the organization applicable (retrieved from the profile option 'ORG\_ID' that usually is set at responsability level which is set by running apps\_initialize; this is the normal way of working in Oracle EBS).

CREATE OR REPLACE PACKAGE apps.xxcust\_process\_xls\_file AUTHID CURRENT\_USER

AS

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*

\* PROGRAM NAME

\* XXCUST\_PROCESS\_XLS\_FILE.pks

\*

\* DESCRIPTION

\* Process Uploaded Excel file

\*

\* CHANGE HISTORY

\* Who When What

\* ---------------------------------------------------------------

\* M. Weeren 01-07-2014 Initial Version

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

PROCEDURE main ( p\_errbuf OUT VARCHAR2

, p\_retcode OUT VARCHAR2

, p\_file\_id IN NUMBER

, p\_file\_name IN VARCHAR2

, p\_user\_id IN NUMBER

, p\_resp\_id IN NUMBER

, p\_resp\_appl\_id IN NUMBER

, p\_sec\_group\_id IN NUMBER

);

END xxcust\_process\_xls\_file;

/

CREATE OR REPLACE PACKAGE BODY apps.xxcust\_process\_xls\_file

AS

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*

\* PROGRAM NAME

\* XXCUST\_PROCESS\_XLS\_FILE.pkb

\*

\* DESCRIPTION

\* Process Uploaded Excel file

\*

\* CHANGE HISTORY

\* Who When What

\* ---------------------------------------------------------------

\* M. Weeren 01-07-2014 Initial Version

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

PROCEDURE main ( p\_errbuf OUT VARCHAR2

, p\_retcode OUT VARCHAR2

, p\_file\_id IN NUMBER

, p\_file\_name IN VARCHAR2

, p\_user\_id IN NUMBER

, p\_resp\_id IN NUMBER

, p\_resp\_appl\_id IN NUMBER

, p\_sec\_group\_id IN NUMBER)

IS

-- initialiaze Oracle Apps

fnd\_global.apps\_initialize(p\_user\_id,p\_resp\_id,p\_resp\_appl\_id,p\_sec\_group\_id);

mo\_global.set\_policy\_context('S',fnd\_profile.value('ORG\_ID'));

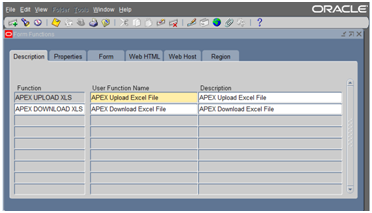
....

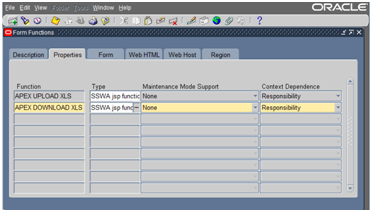
END;

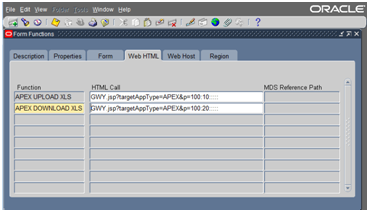
END xxcust\_process\_xls\_file;

/

**F. Add function to EBS R12**  
This paragraph shows how to make the Apex page available in EBS R12. Hereafter the function can be attached to any menu where the user wants to get access to the Apex page (System Administrator: Application > Menu).  
  
Path in EBS R12: System Administrator: Application > Function  
  
Function:Apex UPLOAD XLS  
User Function Name: Apex Upload Exel File  
Description: Apex Upload Exel File  
Type: SSWA jsp function  
HTML Call: GWY.jsp?targetAppType=Apex&p=100:20:::::   
*(In this example 100 is the ID of the application in Apex and 20 is the number of the Page)*

[](http://4.bp.blogspot.com/-mYSlF67A6ik/U8brJMF6--I/AAAAAAAAAhk/-HYwrpwnXJU/s1600/Function0.png)

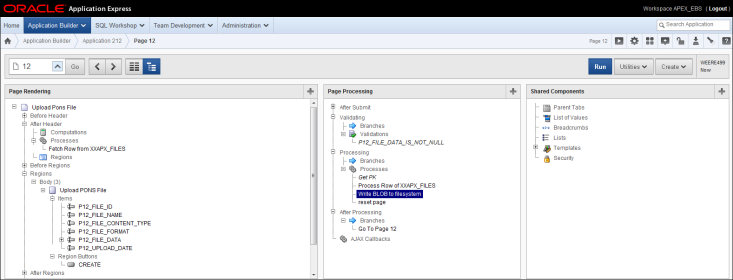
[](http://4.bp.blogspot.com/-YHdwOGk0sfw/U8bqaeiBFfI/AAAAAAAAAhU/SVEZ41Xz4sU/s1600/Function1.png)

[](http://3.bp.blogspot.com/-EREu85VIvHw/U8bqcttQ3HI/AAAAAAAAAhc/zfTN1Nvj15U/s1600/Function2.png)

PS The complete GWY syntax is: GWY.jsp?targetAppType=Apex& p=: :::::. In this way it is possible to pass some session values via the URL to Apex: GWY.jsp?targetAppType=Apex&p=100:10:::::EBS\_RESP\_ID, EBS\_APP\_ID,EBS\_SEC\_GROUP:[RESPONSIBILITY\_ID],[RESP\_APPL\_ID], [SECURITY\_GROUP\_ID].  
But no USER\_ID and no ORG\_ID. Therefore in our application session parameters are retrieved from table icx\_session using the function icx\_sec.getsessioncookie to determine the actual session\_id. Moreover it is safer not to pass parameters via the URL, despite the ability of Apex to manage whether manupalation of parameters via the URL must be ignored or not. 

**6. Upload a file in Apex and write it to the UNIX filesystem**

A csv file is uploaded from Windows into a blob via Apex. Subsequently this blob is written to a file on Unix. There is a lot of code on the internet that do the job, but at the end you will see ctrl-M at the end of each line when opening the file in the vi editor. This is due to the difference between Windows and and Unix. Windows puts a carriage return and linefeed (CR/LF) at the end of each line whereas Unix is only using a linefeed. The extra symbol show up as ^M in vi. To get rid of this character we have to cast the raw (chunck of) data to char and then replace CHR(13) (the CR) by nothing before writing it to the filesystem. The rest of te code in this procedure write\_dos\_blob\_to\_unix\_file is pretty much the same as other code you will find on the internet.  
  
The page in Application Express (Apex) is similar to upload page I created earlier (see post [Upload a file in Apex and submit concurrent request in EBS R12](http://apex4ebs.blogspot.nl/2014/07/upload-file-in-apex-and-submit.html)). Instead of submitting a request in the Oracle E-Business Suite R12 (EBS R12), the additional custom process ('Write BLOB to filesystem') calls a routine that writes the blob content to a unix file. This routine is added to the apex\_global package, where it will run as APPS user (because of the authid clause). This was necessary because APPS already has the rights to write to the unix filesystem via directory 'XXCUST\_TEMP\_FILES'. (Here directory is an Oracle database object that contains the actual path on the filesystem and where rights can be assigned to).

[](http://2.bp.blogspot.com/-sz3KUNxP97U/U8dvkmWuPQI/AAAAAAAAAhw/_YsgxW-ZZrM/s1600/Upload_and_write_to_unix_Apex_page.PNG)

Source in the process definition of the additional process 'Write BLOB to filesystem':

BEGIN

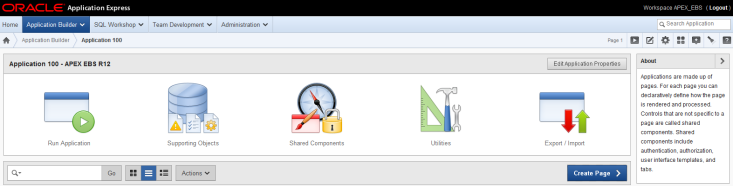
apps.apex\_global.write\_dos\_blob\_to\_unix\_file('XXCUST\_TEMP\_FILES',TO\_NUMBER(V('P12\_FILE\_ID')));

END;

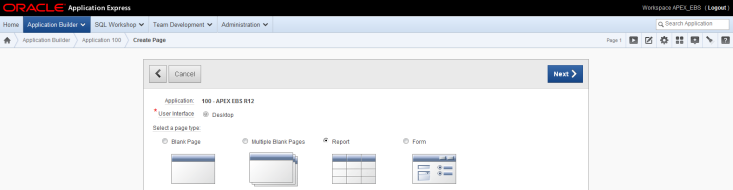
Click on the link to see the [code of package apex\_global](http://apex4ebs.blogspot.nl/p/code-apexglobal-package.html)

**7. Download a file stored in a blob field from an Apex page**

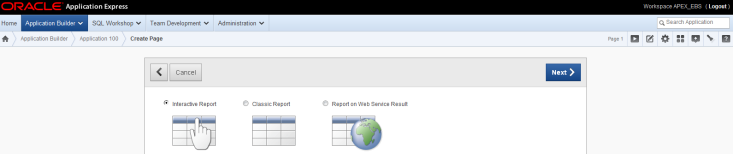
I already described an Apex page where it was possible to upload a file to a database field of type blob. Here I will create a page on which you can download the file again from that same blob field. The trick is creating a report page where a additional field is shown as a download link. This additional field is a 'getlength' of that blob field, and the properties of that field contain the characteristics of the file to download (name, character set, etc).   
  
Application Builder

[](http://2.bp.blogspot.com/-zgNrEPAkIZs/VDf1TuDyfvI/AAAAAAAAAnw/izrH1H8YGI4/s1600/DownloadPage_1.png)

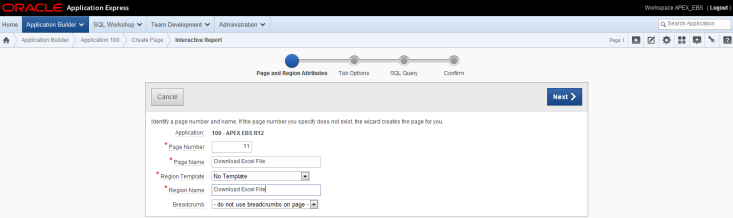
Create Page >

[](http://1.bp.blogspot.com/-R2ZMJPZDz_w/VDf1b7OFRGI/AAAAAAAAAn4/QoZxkD8c8Gk/s1600/DownloadPage_2.png)

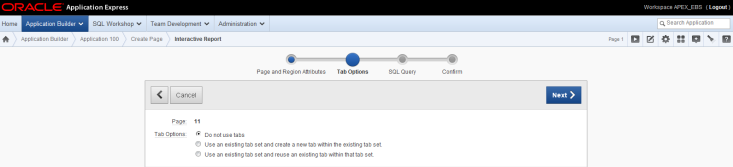
Select 'Report'  
Next >

[](http://2.bp.blogspot.com/-O6EAcZzpMgg/VDf1fOGBF0I/AAAAAAAAAoA/8Da_zjjclxg/s1600/DownloadPage_3.png)

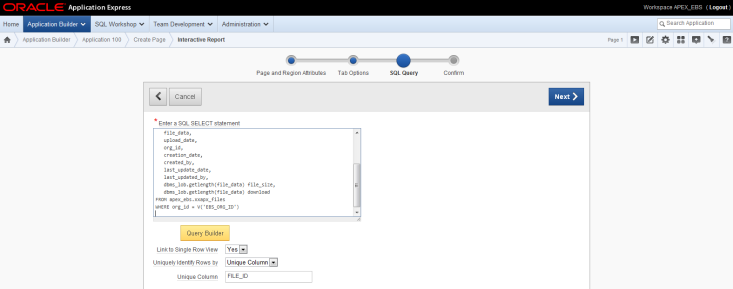
Select 'Interactive Report'  
Next >

[](http://1.bp.blogspot.com/-ythP9fVmKME/VDf1iITiMZI/AAAAAAAAAoI/7NOndLR1LtY/s1600/DownloadPage_4.png)

Page Number: 11  
Page Name: Download Excel File  
Region Template: No Template (default)  
Region Name: Download Excel File  
Breadcrumb: - do not use breadcrumbs on this page  
Next >

[](http://2.bp.blogspot.com/-kfUO2aYCQfg/VDf1kH00MfI/AAAAAAAAAoQ/ioS1ak7TGv8/s1600/DownloadPage_5.png)

'Do not use tabs'  
Next >

[](http://2.bp.blogspot.com/-oRMHpLGWz_Q/VDf1nO-5r5I/AAAAAAAAAoY/OxDrUrEBhdI/s1600/DownloadPage_6.png)

About SQL Query  
- Added is the WHERE clause checking the operating unit (org\_id) of the user. EBS\_ORG\_ID is an Application Item that contains the org\_id linked to the the Oracle E-Business Suite responsibility from where the Apex page was launched.  
- The column 'download' (has to be dbms\_lob.getlength(file\_data)) is added to make a link that will allow the user to download the file stored in the column FILE\_DATA (blob).  
  
SQL Query

SELECT

file\_id,

file\_name,

file\_content\_type,

language,

oracle\_charset,

file\_format,

file\_data,

upload\_date,

org\_id,

creation\_date,

created\_by,

last\_update\_date,

last\_updated\_by,

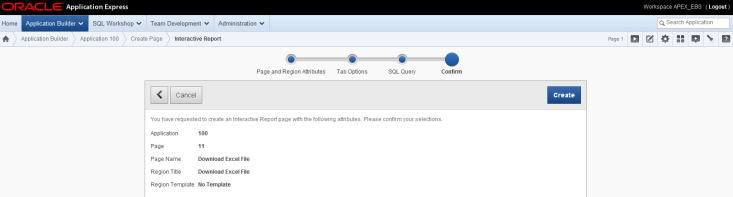
dbms\_lob.getlength(file\_data) file\_size,

dbms\_lob.getlength(file\_data) download

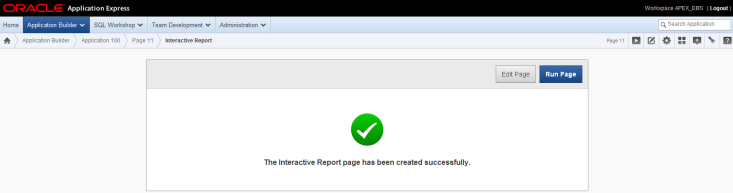
FROM apex\_ebs.xxapx\_files

WHERE org\_id = V('EBS\_ORG\_ID')

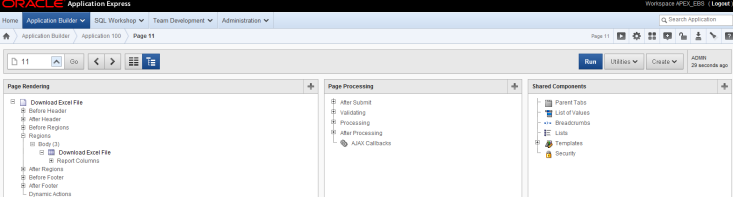
Link to single row view: Yes  
Uniquely Identify Rows by: Unique Column  
Unique Column: FILE\_ID  
Next >

[](http://2.bp.blogspot.com/-nnAjfej390o/VDf1o4666xI/AAAAAAAAAog/vXFMPSZQYfE/s1600/DownloadPage_7.png)

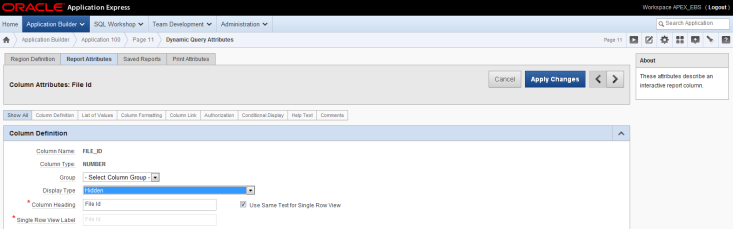
B: Create

[](http://2.bp.blogspot.com/-dV_rSbUZVDs/VDf1rNLYaSI/AAAAAAAAAoo/xRkvng0u894/s1600/DownloadPage_8.png)

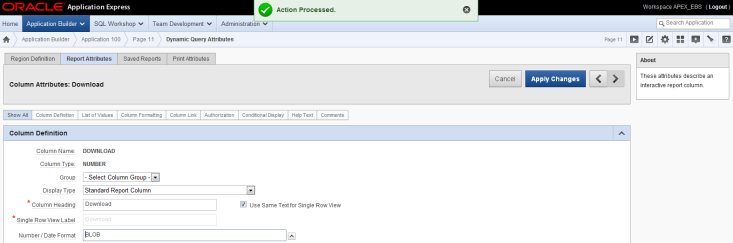
B : Edit Page

[](http://3.bp.blogspot.com/-HRLQ_wHxoF0/VDf1sSUXpFI/AAAAAAAAAow/iKerf5mykds/s1600/DownloadPage_9.png)

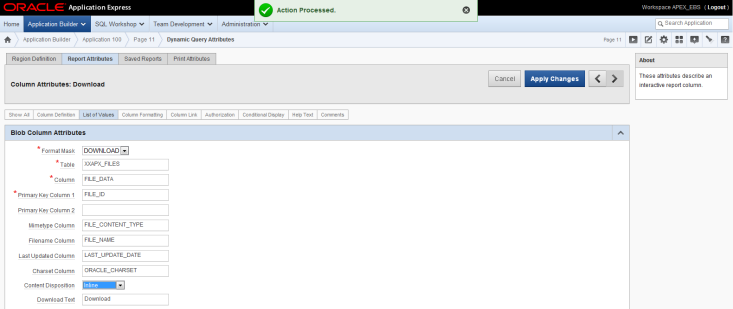
Page Rendering: Download Uploaded Rate Tables > Regions > Body > Download Uploaded Rate Tables > Report Columns Expand Report Columns  
DoubleClick on FILE\_ID

[](http://1.bp.blogspot.com/-CVM5gPzRhIA/VDf1twswIRI/AAAAAAAAAo4/fm8AW3VaX9g/s1600/DownloadPage_10.png)

Display Type: Hidden  
  
Do the same for the rest of the Columns except for:  
- FILE\_NAME  
- UPLOAD\_DATE  
- CREATED\_BY (change Column Heading to 'Uploaded by')  
- FILE\_SIZE  
- DOWNLOAD  
Browse to the next Column by pressing >  
  
For column DOWNLOAD change the following attributes:  
Display Type: Standard Report Column  
Number / Date Format: BLOB Format (select from List of Values)

[](http://2.bp.blogspot.com/-KGzb-6cpWV0/VDf1uyhQgVI/AAAAAAAAApA/2FvSDwZBmls/s1600/DownloadPage_11.png)

The Blob Column Attributes become visible:

[](http://2.bp.blogspot.com/-hwa5-v5BMCk/VDf1wVzv6gI/AAAAAAAAApI/HHHHGFrdtck/s1600/DownloadPage_12.png)

Format Mask: DOWNLOAD (default)  
Table: XXAPX\_FILES  
Column: FILE\_DATA  
Primery Key Column 1: FILE\_ID  
Mimetype Column: FILE\_CONTENT\_TYPE  
Filename Column: FILE\_NAME  
Last Updated Column: LAST\_UPDATE\_DATE  
Charset Column: ORACLE\_CHARSET  
Content Dispostion: Inline  
Download Text: Download (default)  
B: Apply Changes  
  
To integrate this page in Orace EBS see my post Upload a file in Apex and submit concurrent request in EBS R12, paragraph 6 [Add function to EBS R12](http://apex4ebs.blogspot.com/p/integration-of-oracle-apex-and-e.html#par05.6)