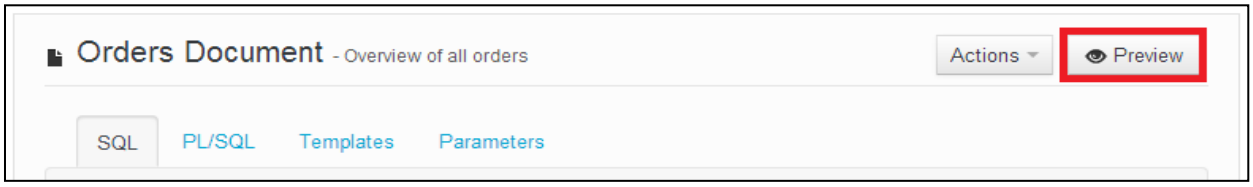
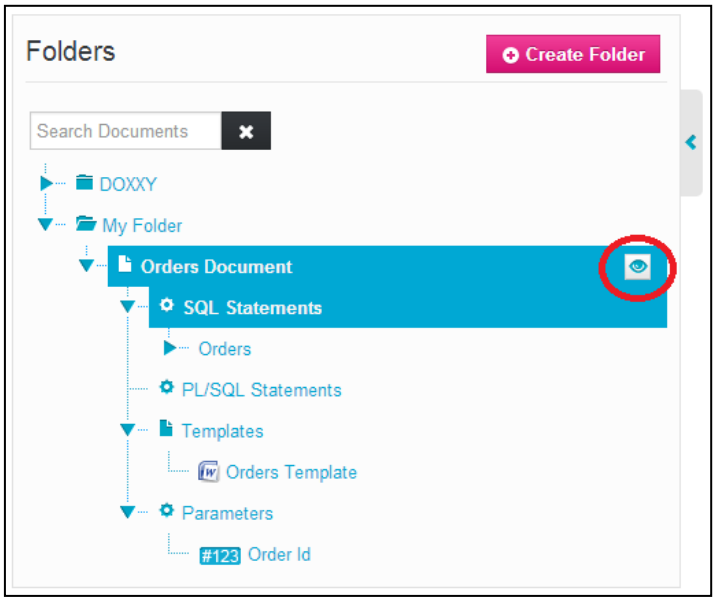
# Generate and download a Doxxy report

Developing reports and documents within the Doxxy tool is very easy thanks to the intuitive and user friendly APEX interface. At any time you can also review your report by requesting a preview. You can do this in two ways in the Doxxy tool:

1) By pressing the "Preview" button in the upper-right corner:




2) By clicking the "eye" icon that appears when you hover over the document in the tree:




But how can you load your desired report from within your own application?
Doxxy provides a PL/SQL API which encapsulates all processing of the report generation. You pass the necessary parameters, and the API returns a BLOB which contains the desired output (DOCX or PDF). You can then do with this BLOB whatever you want: you can store it in the database or on a file system, you can print it, you can mail it, download it, … This document explains how you can implement the call to the API and the download process in Oracle Application Express.

### Step 1: Code to generate your report

The PL/SQL API is the generate_document. This function generates your document (DOCX or PDF) and returns it in a BLOB.

In fact, there are two generate_document functions:

1) a basic form, in which you can pass on a limited number of parameters
2) a generic form, in which the parameters are passed on through a record set, so unlimited possibilities

Both of these functions are located in the BL$DG4O_INTERFACE package, under the database schema where Doxxy is installed.

***Basic form***

```
FUNCTION generate_document( p_document IN VARCHAR2
                           ,p_document_template_name IN VARCHAR2
                           ,p_bind_variable_name_1 IN VARCHAR2 DEFAULT NULL
                           ,p_bind_variable_text_1 IN VARCHAR2 DEFAULT NULL
                           ,p_bind_variable_name_2 IN VARCHAR2 DEFAULT NULL
                           ,p_bind_variable_text_2 IN VARCHAR2 DEFAULT NULL
                           ,p_bind_variable_name_3 IN VARCHAR2 DEFAULT NULL
                           ,p_bind_variable_number_3 IN NUMBER DEFAULT NULL
                           ,p_bind_variable_name_4 IN VARCHAR2 DEFAULT NULL
                           ,p_bind_variable_number_4 IN NUMBER DEFAULT NULL
                           ,p_bind_variable_name_5 IN VARCHAR2 DEFAULT NULL
                           ,p_bind_variable_date_5 IN DATE DEFAULT NULL
                           ,p_bind_variable_name_6 IN VARCHAR2 DEFAULT NULL
                           ,p_bind_variable_date_6 IN DATE DEFAULT NULL
                           ,p_application_key IN VARCHAR2 DEFAULT NULL
                           ,p_user_id IN dg4o_users.username%TYPE DEFAULT NULL
                           ,p_format IN VARCHAR2 DEFAULT 'DOCX')
RETURN BLOB
```

In this function:

- Use the “p_document” parameter to select the document that you require. The name of the document and *the full path* should be specified, please note that these are *case sensitive*.
- Use the “p_document_template_name” parameter to select the template that you require.
- You have two parameter options in which you can enter the VARCHAR2 data type, two parameters for the NUMBER data type and two parameters for the DATE data type. This can cause a restriction if your report requires multiple parameters. In that case, you will use the more generic way of the retrieve function (see below).
- Use the “p_application_key” parameter to specify the application key, if it is required for the document.
- Use the parameter “p_user_id” to pass the username. This parameter is provided for future development.
- Use the parameter “p_format” to specify the required output format. In the TRIAL-version, by default, only DOCX is possible. If you have a license for the PDF option, you can also specify PDF here.

***Generic form***

```
FUNCTION generate_document( p_document IN VARCHAR2
                           ,p_document_template_name IN VARCHAR2
                           ,p_bind_variables IN bind_variables_t
                           ,p_application_key IN VARCHAR2 DEFAULT NULL
                           ,p_user_id IN dg4o_users.username%TYPE DEFAULT NULL
                           ,p_format IN VARCHAR2 DEFAULT 'DOCX')
RETURN BLOB
```

In this function:

- The parameters “p_document”, “p_document_template_name”, “p_application_key”, “p_user_id” and “p_format” have the same meaning in the generic form, as in the limited form.
- You will now only need one record type parameter, instead of a set of parameters, to submit the input data. This record set provides unlimited number of parameters to pass on.

***Example***

In order to use these functions in your own application, you will need a GRANT on the package BL$DG4O_INTERFACE when running this from another database scheme (let's say 'DEMO').
A useful tip is to also create a synonym (private or public), e.g. BL$DG4O_INTERFACE.

Within the DOXXY database scheme:

```
GRANT EXECUTE on BL$DG4O_INTERFACE to DEMO;
```

Within your DEMO database scheme:

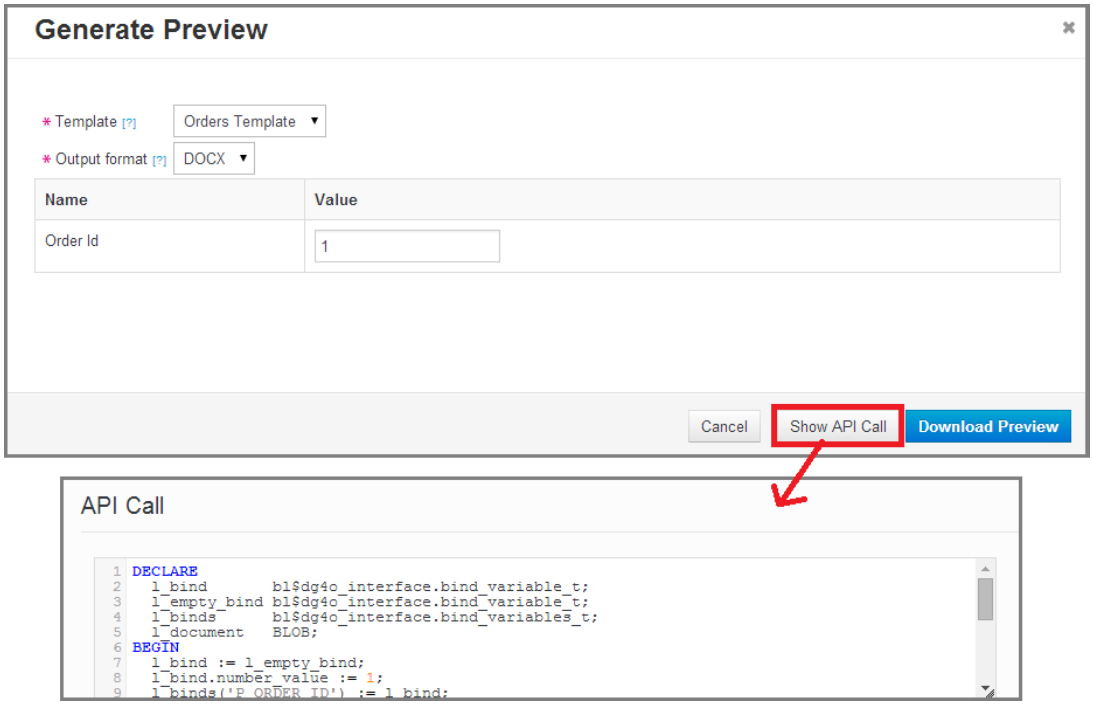
```
CREATE SYNONYM BL$DG4O_INTERFACE for DOXXY.BL$DG4O_INTERFACE;
```

You can now generate your document by performing the following procedure in your application (this uses the generic form of the retrieve function).

```
DECLARE
     l_bind_variables BL$DG4O_INTERFACE.bind_variables_t;
     l_doc BLOB;
BEGIN
     ------------------------------------------------------------
     -- Generate report
     ------------------------------------------------------------
     l_bind_variables('P_ORDER_ID').number_value := 1;
     l_doc := BL$DG4O_INTERFACE.generate_document
          (p_document => '/My Folder/Orders Document'
          ,p_document_template_name => 'Orders Template'
          ,p_bind_variables => l_bind_variables
          ,p_application_key => ''
          ,p_user_id => ''
          ,p_format => 'DOCX');
END;
```

***Tip***

Use the "Show API Call" button in the "Generate Preview" screen to generate this PL/SQL code. You can copy and paste this code in your own application.



### Step 2: Code to download your report

The generate_document function will generate a BLOB, which contains your DOCX or PDF document. You will need to execute this extra piece of code to download your BLOB file from your web application:

```
DECLARE
    l_bind_variables BL$DG4O_INTERFACE.bind_variables_t;
    l_doc BLOB;
BEGIN
    ------------------------------------------------------------
    -- Generate report
    ------------------------------------------------------------
    l_bind_variables('P_ORDER_ID').number_value := 1;
    l_doc := BL$DG4O_INTERFACE.generate_document
         (p_document => '/My Folder/Orders Document'
         ,p_document_template_name => 'Orders Template'
         ,p_bind_variables => l_bind_variables
         ,p_application_key => ''
         ,p_user_id => ''
         ,p_format => 'DOCX');
    ------------------------------------------------------------
    -- Download report
    ------------------------------------------------------------
    -- Initialize
    htp.flush;
    htp.init;
    -- Set up HTTP header
    owa_util.mime_header('application/octet-stream',false);
    -- Set the size so the browser knows how much to download
    htp.p('Content-length:'||dbms_lob.getlength(l_doc));
    -- The filename will be used by the browser if the users does a 'Save as'
    htp.p('Content-Disposition:attachment;filename="Orders.docx"');
    -- Close headers
    owa_util.http_header_close;
    -- Download the blob
    wpg_docload.download_file(l_doc);
    apex_application.stop_apex_engine;
END;
```

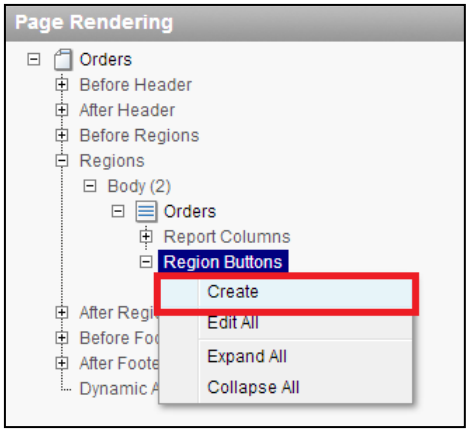
***Additional remark***
Use *apex_application.stop_apex_engine* (from version APEX 4.1) when downloading your report within a *Before Header* process. The engine will just download your BLOB and will then stop the page immediately from rendering, so your application will not go to this download page and the user will never see it.
From version APEX version 4.2 it is recommended to use the *apex_application.stop_apex_engine* function!

### Step 3: APEX download button + branch

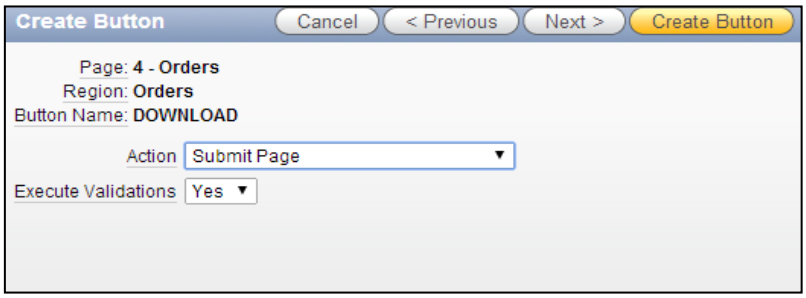
There are multiple ways to add download functionality on your web application. In this section we will explain how you can create a download button with a branch containing the PL/SQL procedure to download your BLOB file.

***Create a button***

First create a button (for example “Download”) in the region of your choice (*Page Rendering* part).

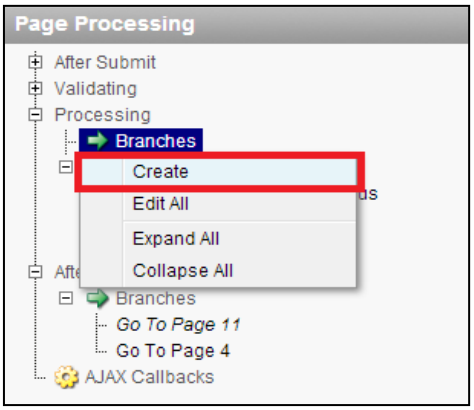



Fill in the fields. For “Action” choose “Submit Page”.

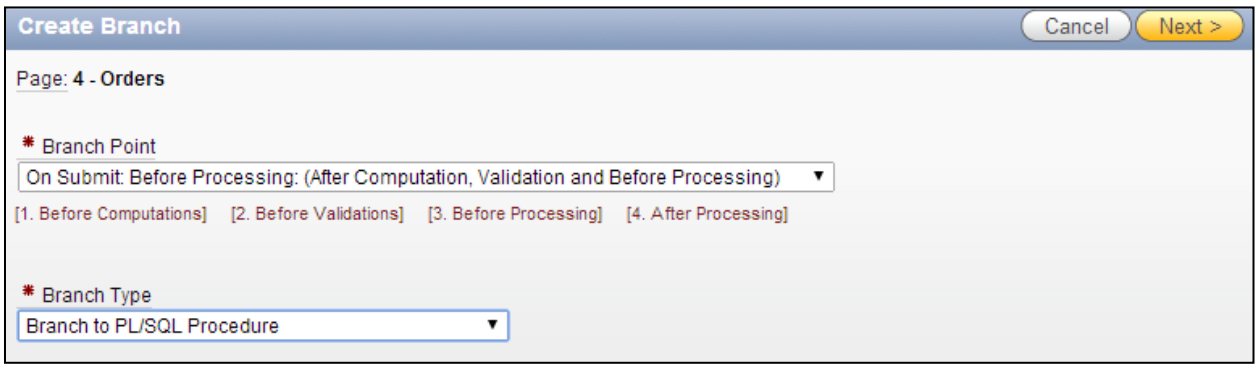



***Create a branch***

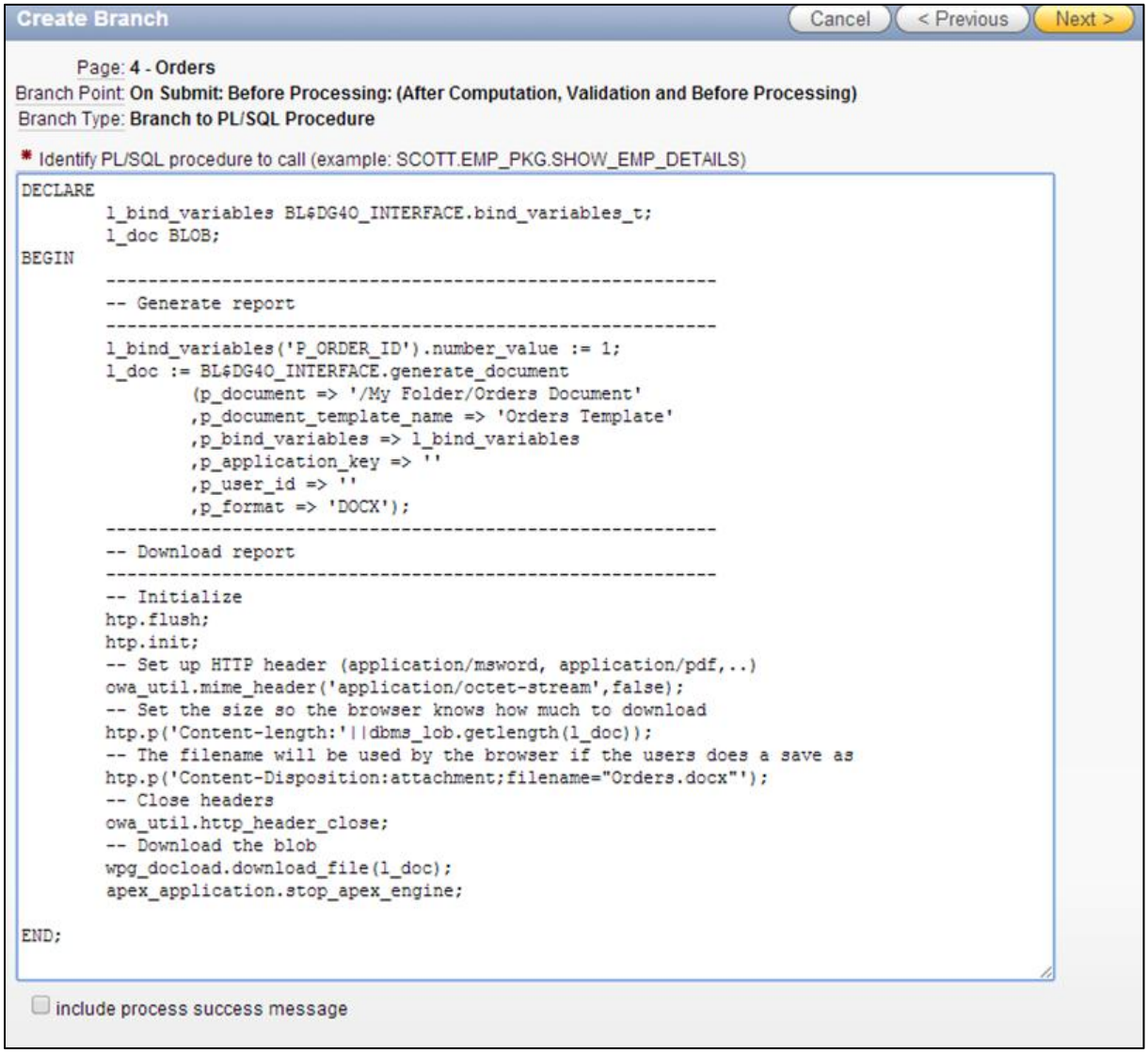
Next create a branch (*Page Processing* part).



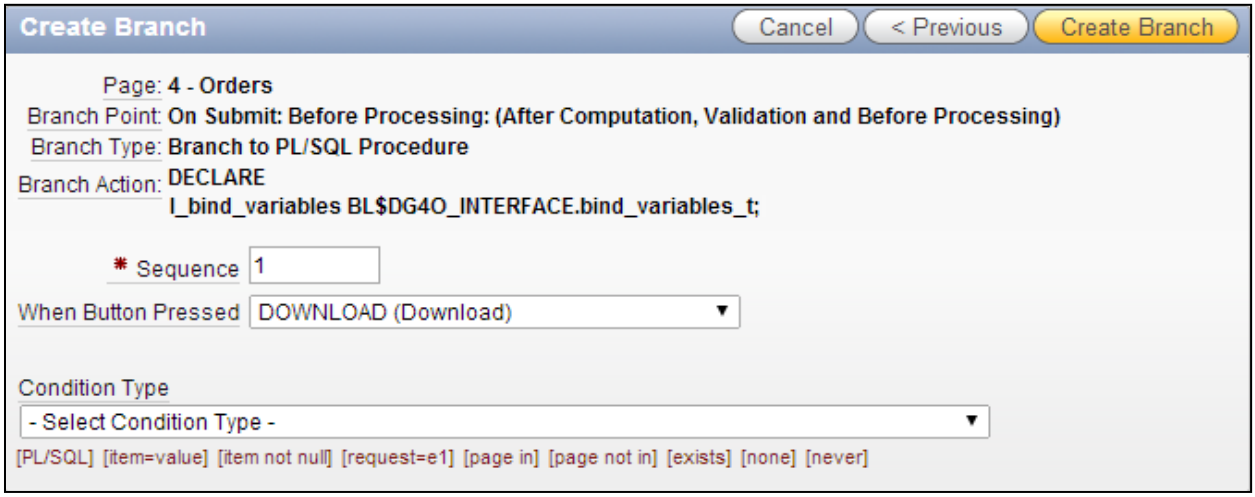

Fill in the next fields. For “Branch Type” choose “Branch to PL/SQL Procedure”.



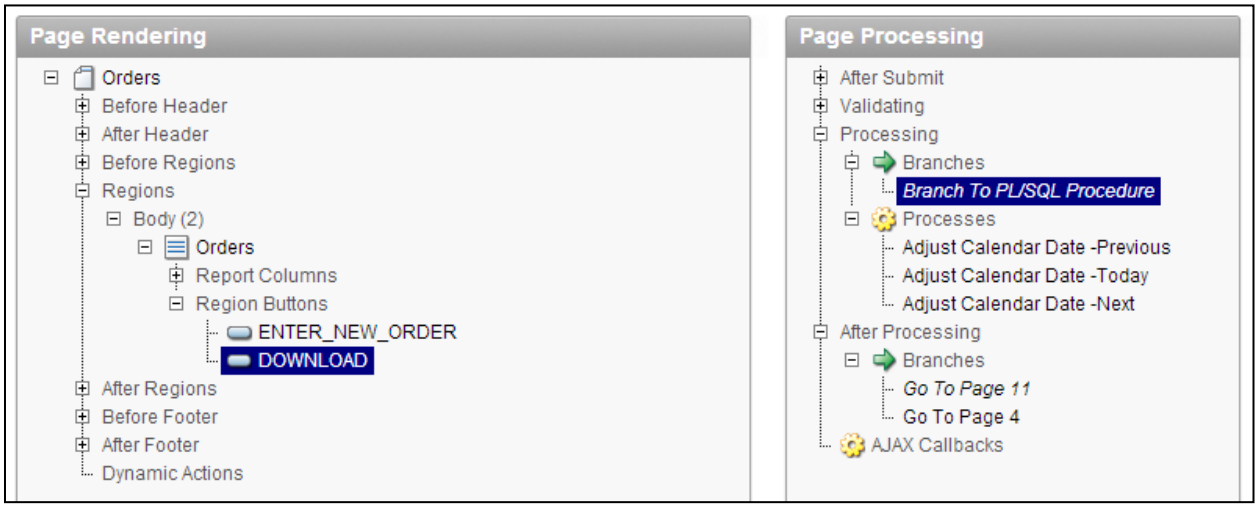
Enter your PL/SQL procedure.
*You can also create a stored procedure in the database and enter a call to this procedure here.*




Continue the wizard. Select your button in the “When Button Pressed” select list.




You are ready now!




Run your page and press the Download-button to test your download process. Your Orders.docx report will be downloaded.