APEX External FOP tracing plugin

License

The plugin has been developed by the J4L FO Designer team (http://www.apex-reports.com/) and will work with any report. It is a free component to use and modify.

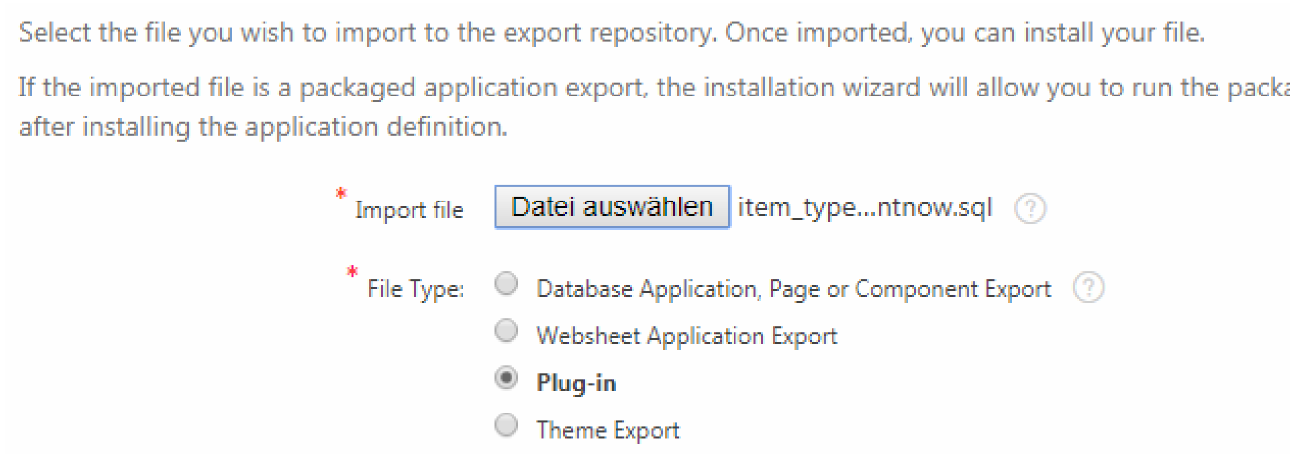
Introduction

This plugin provides the follwing capabilities using the J4L FOP server:

- Tracing of XSL-FO templates execution for debugging purposes
- It can be used to display and download the realtime XML or the generic XSL-FO generated by Oracle APEX
- Generation of PDF (limited for non customers)
- Generation of Excel files (limited for non customers)

Import plugin

Import the file *process_type_plugin_externalpdf.sql* with the option: *application -> shared components -> plug-ins -> import*



Select the file you wish to import to the export repository. Once imported, you can install your file.

If the imported file is a packaged application export, the installation wizard will allow you to run the packa
after installing the application definition.

* Import file: Datei auswählen item_type...ntnow.sql

* File Type:
○ Database Application, Page or Component Export
○ Websheet Application Export
● **Plug-in**
○ Theme Export

Preparation

First you need to configure Oracle APEX to allow connections to our FOP Server (eval.apex-reports.com). Follow the directions in this page:

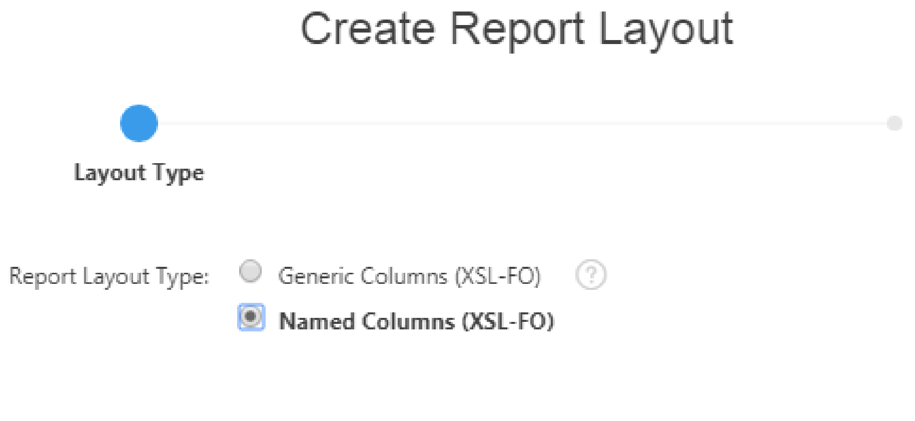
http://www.apex-reports.com/saassetup.html

Note you can use HTTP or HTTPS, for testing purposes we recommend HTTP since you do not have to upload the certificates to your Oracle APEX wallet.

How to use the plug-in

Case *Named Columns Layout*

For this use case you need to have a self-developed XSL-FO report layout which you have uploaded to the Shared *components → Reports layouts*



Create Report Layout

Layout Type

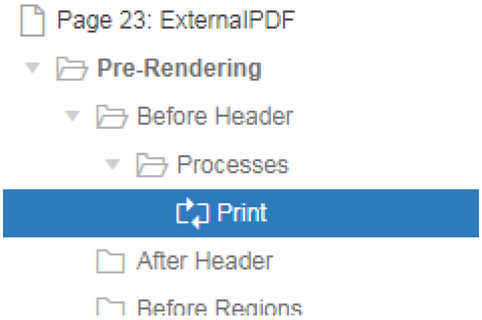
Report Layout Type: ○ Generic Columns (XSL-FO) ● Named Columns (XSL-FO)

Note: you will get a more understandable trace if you add comments to your XSL-FO (as for example the J4L FO designer does). For this you add an <xsl:comment> tag as below right before the output of your report's fields:

```
 <xsl:comment> End Date Field </xsl:comment>
<fo:block   font-size="10pt"  font-family="SansSerif"  color="#000000"   text-
align="left"  margin-left="0.51cm" margin-right="0.51cm"  margin-top="0.0cm"   >
<xsl:value-of select="END_DATE"  ></xsl:value-of>
</fo:block>
```

As second step your create **an empty page** in Oracle Apex. You must:

- Add a “before header” process called “print”

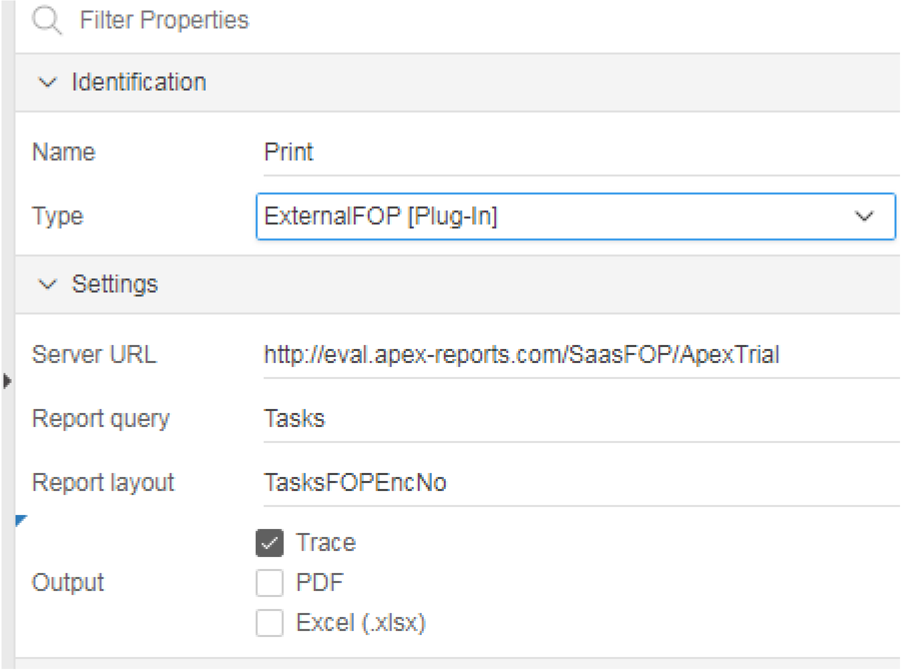


Page 23: ExternalPDF

- Pre-Rendering
  - Before Header
    - Processes
      - Print
  - After Header
  - Before Regions

- Select as Type the “ExternalFOP” plugin.
  - Select “*Trace*” and
  - enter “*Report query”* and “*Report layout”* which you must have previously defined in the share components section of the application.

    Warning!, use only Named columns layouts.



Identification

Name	Print
Type	ExternalFOP [Plug-In]

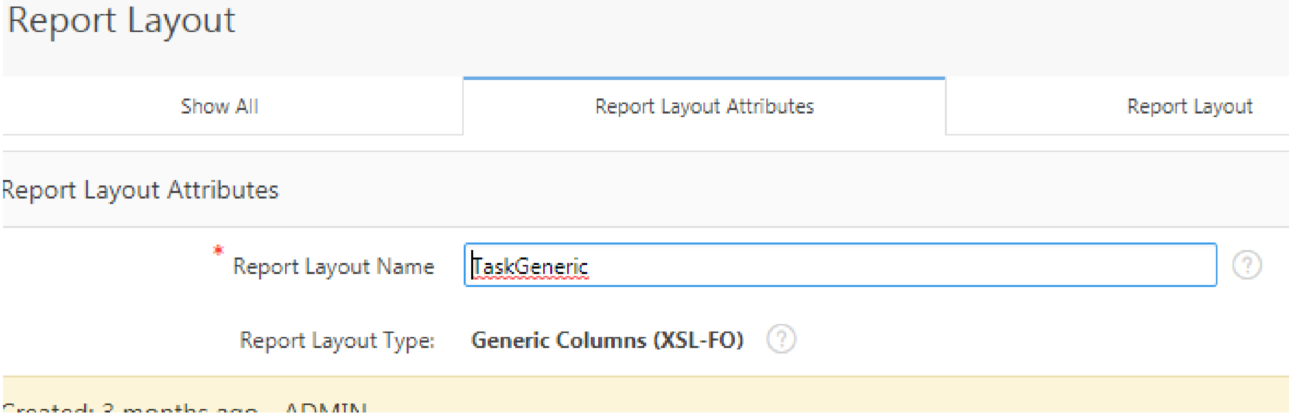
Settings

Server URL	http://eval.apex-reports.com/SaasFOP/ApexTrial
Report query	Tasks
Report layout	TasksFOPEncNo
Output	☑ Trace □ PDF □ Excel (.xlsx)

Case *Generic Columns Layout*

If you want to trace the execution of a generic report or display the generated layout you must proceed as follows:

First create a Shared report layout, type “Generic Columns”

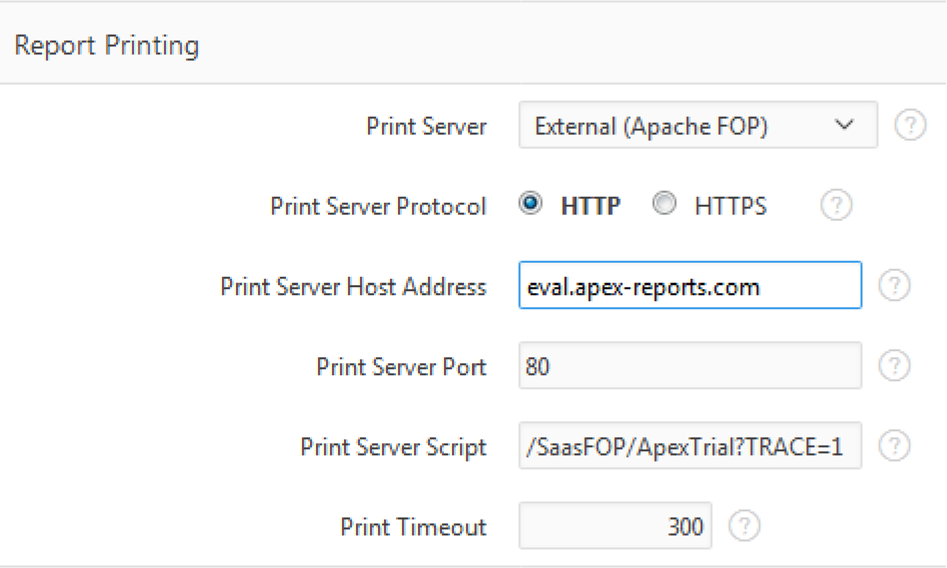


* Report Layout Name	TaskGeneric
Report Layout Type:	Generic Columns (XSL-FO)

Created: 3 months ago – ADMIN

The second step is changing the global APEX Printing setting (you cannot use the plugin for this case). Use

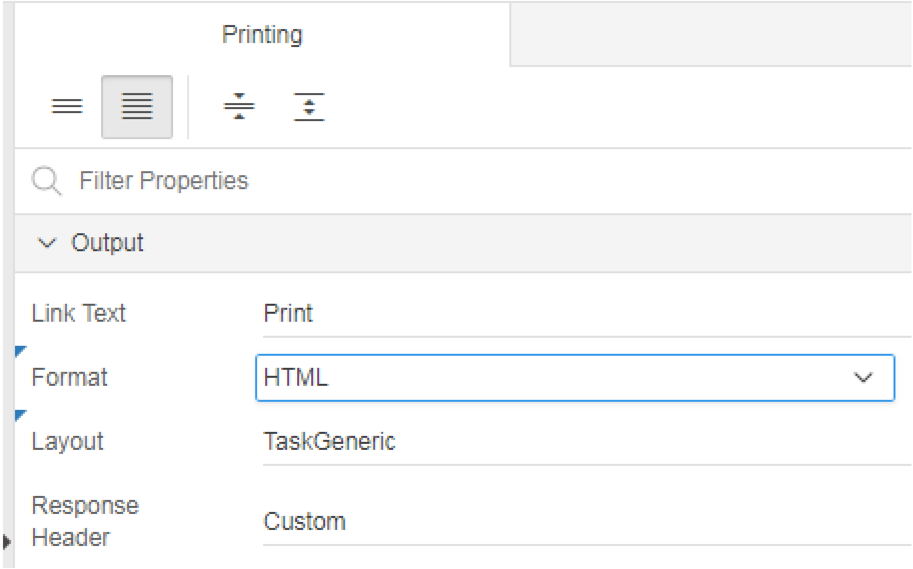
- Address: *eval.apex-reports.com*
- Server Script
  - Trace: */SaasFOP/ApexTrial?TRACE=1*
  - Excel: */SaasFOP/ApexTrial?XLS=1*
  - PDF: */SaasFOP/ApexTrial*



Print Server	External (Apache FOP)
Print Server Protocol	● HTTP ○ HTTPS
Print Server Host Address	eval.apex-reports.com
Print Server Port	80
Print Server Script	/SaasFOP/ApexTrial?TRACE=1
Print Timeout	300

as final step you create a classic report and activate the printing attributes of the region.

- Select your “Generic” report layout
- Select the output format
  - **HTML** if you have used the Trace Script in the global printing settings
  - **Excel** if you have used the Excel Script in the global printing settings
  - PDF for standard output



Link Text: Print
Format: HTML
Layout: TaskGeneric
Response Header: Custom

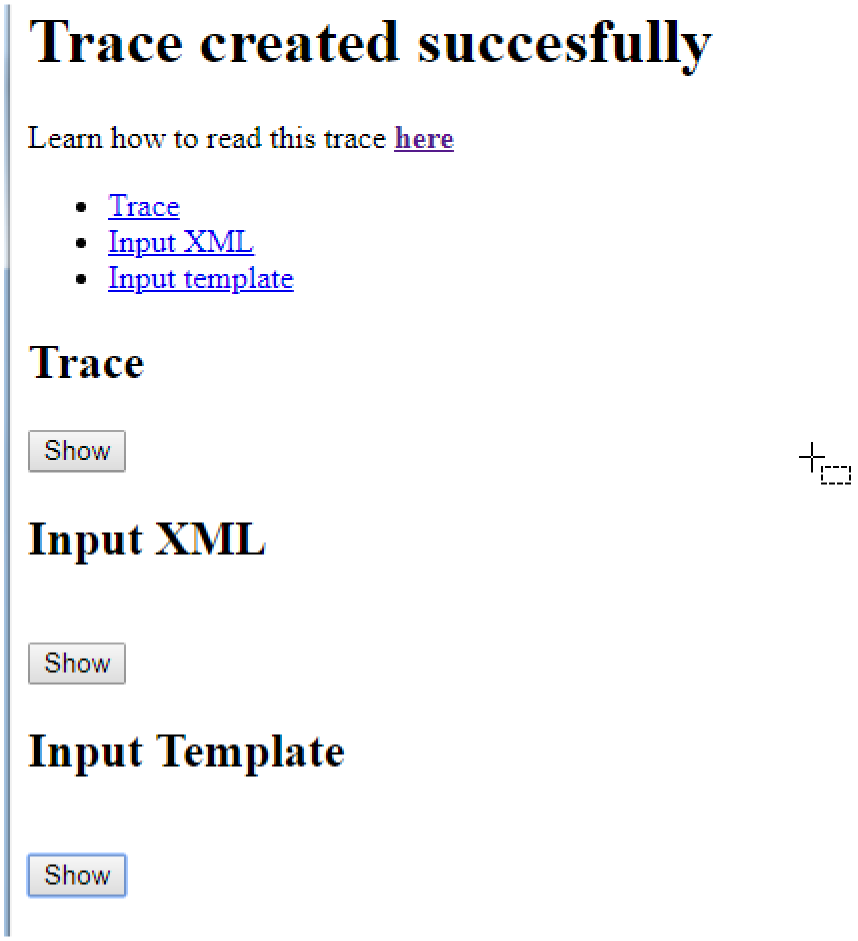
The trace output

The trace output will contain:

- The trace itself
- the XML
- and the XSL-FO template

more information about reading the trace can be found here:

http://www.apex-reports.com/help/structure.html#analyze



Trace created succesfully

Learn how to read this trace here

- Trace
- Input XML
- Input template

Trace

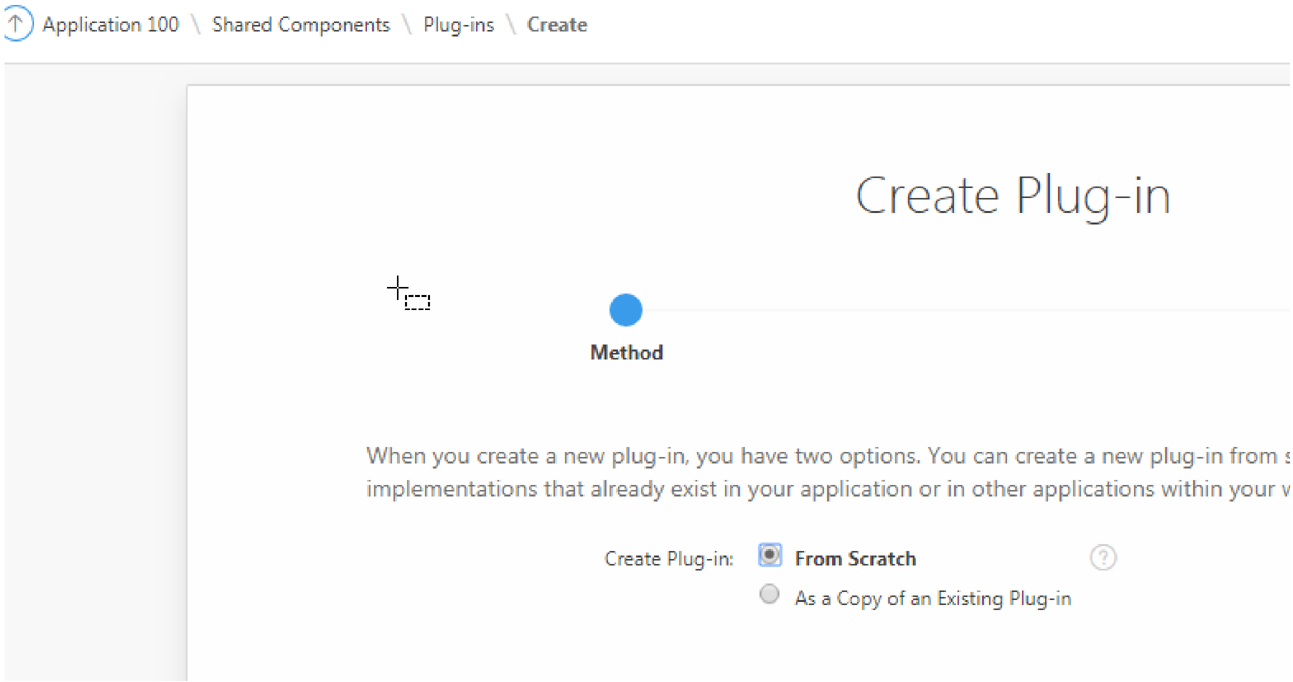
Input XML

Input Template

Creating the plugin manually

If you experiende issues importing the plugin in your APEX version you can create it manually like this.

In the Shared Components select create plugin



Application 100 \ Shared Components \ Plug-ins \ Create

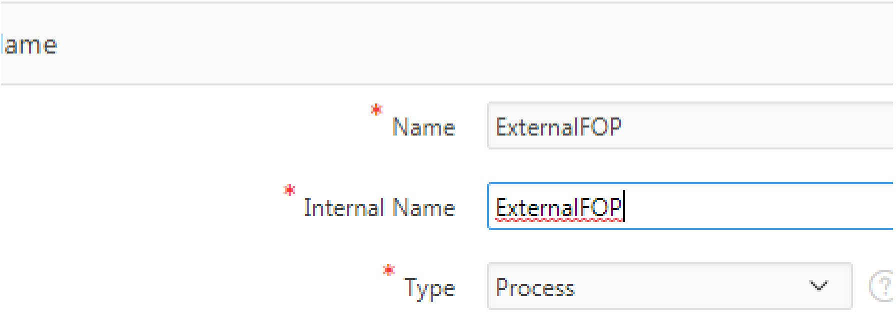
Create Plug-in

Method

When you create a new plug-in, you have two options. You can create a new plug-in from s
implementations that already exist in your application or in other applications within your w

Create Plug-in: ● From Scratch
○ As a Copy of an Existing Plug-in

Enter name and type



ame

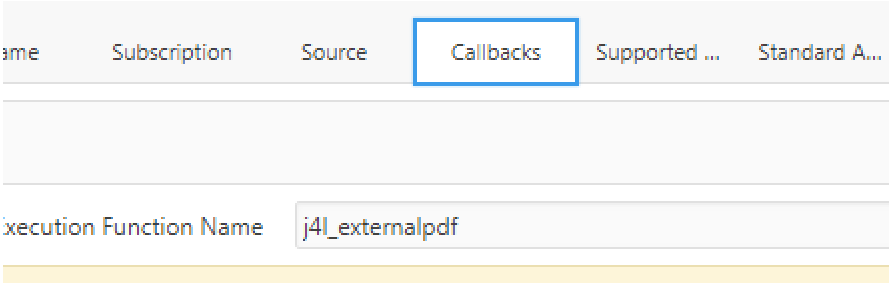
* Name: ExternalFOP

* Internal Name: ExternalFOP

* Type: Process

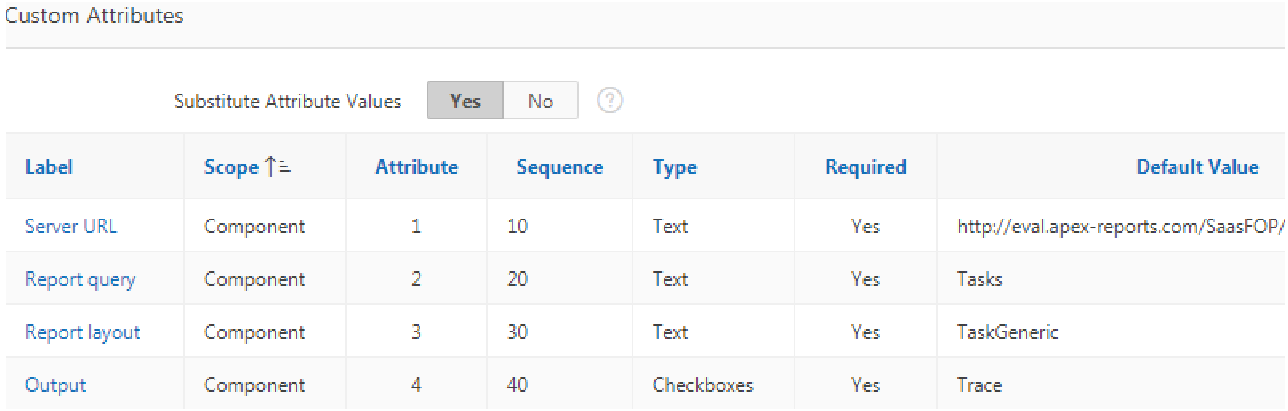
In the code field enter the content of the file “code.sql”

As callback enter *j4l_externalpdf*



:xecution Function Name: j4l_externalpdf

Create 4 attributes like these

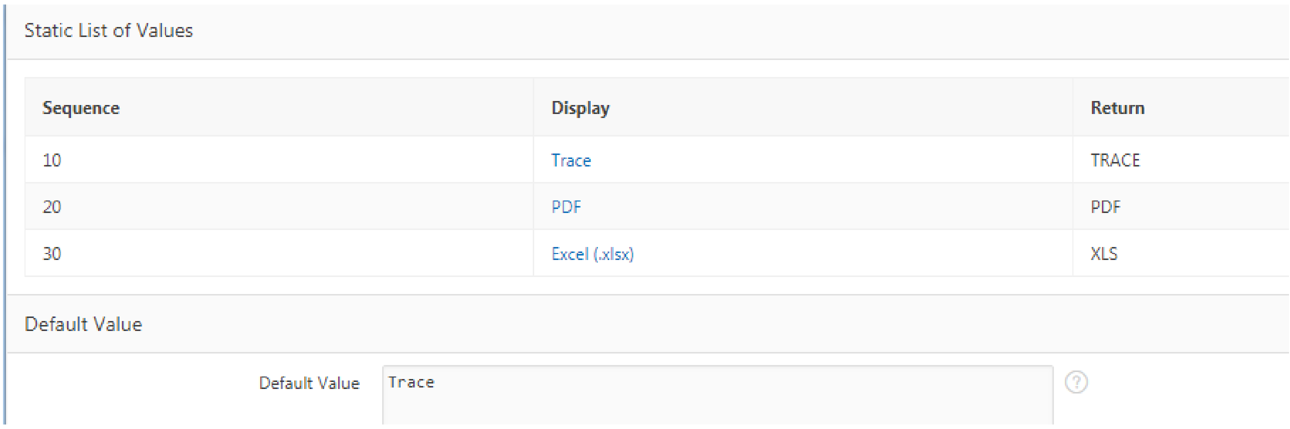


Custom Attributes

Substitute Attribute Values: [✓ Yes] No

Label	Scope ↑	Attribute	Sequence	Type	Required	Default Value
Server URL	Component	1	10	Text	Yes	http://eval.apex-reports.com/SaasFOP/
Report query	Component	2	20	Text	Yes	Tasks
Report layout	Component	3	30	Text	Yes	TaskGeneric
Output	Component	4	40	Checkboxes	Yes	Trace

The first 3 are text attributes, but the last one "Output" is a Checkbox type and contains 3 static values as below



Static List of Values

Sequence	Display	Return
10	Trace	TRACE
20	PDF	PDF
30	Excel (.xlsx)	XLS

Default Value

Default Value: Trace