APEX 5 plugins for everyone

Alan Arentsen

Database Consultancy

APEX 5 plugins for everyone

Alan Arentsen

Database Consultancy

Database
Consultancy

ORACLE
ACE

alanarentsen.blogspot.com

twitter @alanarentsen

ā'pěks

(#orclapex)

HTML 5

CSS 3

JS JavaScript

PL/SQL

HTML

CSS

JavaScript

PL/SQL

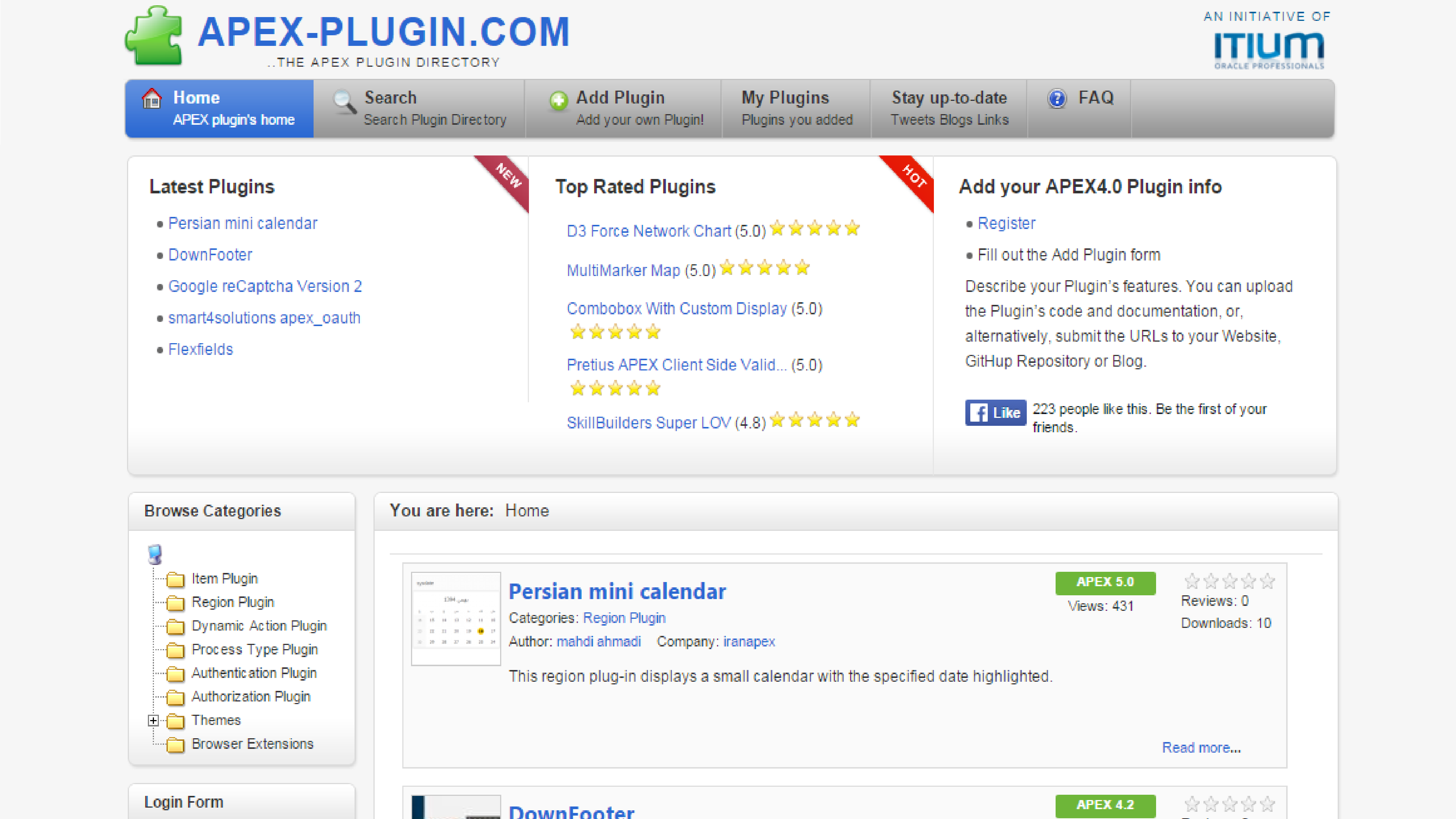
- Authentication Scheme Type
- Authorization Scheme Type
- Dynamic Action
- Item
- Process
- Region

- **Plug-in Repository**
- **Create from scratch**
- **Copy of existing Plug-in**

Plug-in Repository

Oracle Application Express

Plug-Ins



APEX-PLUGIN.COM

..THE APEX PLUGIN DIRECTORY

AN INITIATIVE OF ITIUM ORACLE PROFESSIONALS

Latest Plugins

- Persian mini calendar
- DownFooter
- Google reCaptcha Version 2
- smart4solutions apex_oauth
- Flexfields

Top Rated Plugins

D3 Force Network Chart (5.0) ★★★★★

MultiMarker Map (5.0) ★★★★★

Combobox With Custom Display (5.0) ★★★★★

Pretius APEX Client Side Valid... (5.0) ★★★★★

SkillBuilders Super LOV (4.8) ★★★★★

Add your APEX4.0 Plugin info

- Register
- Fill out the Add Plugin form

Describe your Plugin's features. You can upload the Plugin's code and documentation, or, alternatively, submit the URLs to your Website, GitHup Repository or Blog.

Like 223 people like this. Be the first of your friends.

Browse Categories

- Item Plugin
- Region Plugin
- Dynamic Action Plugin
- Process Type Plugin
- Authentication Plugin
- Authorization Plugin
- Themes
- Browser Extensions

Login Form

You are here: Home

Persian mini calendar

APEX 5.0

Views: 431

☆☆☆☆☆

Reviews: 0

Downloads: 10

Categories: Region Plugin

Author: mahdi ahmadi Company: iranapex

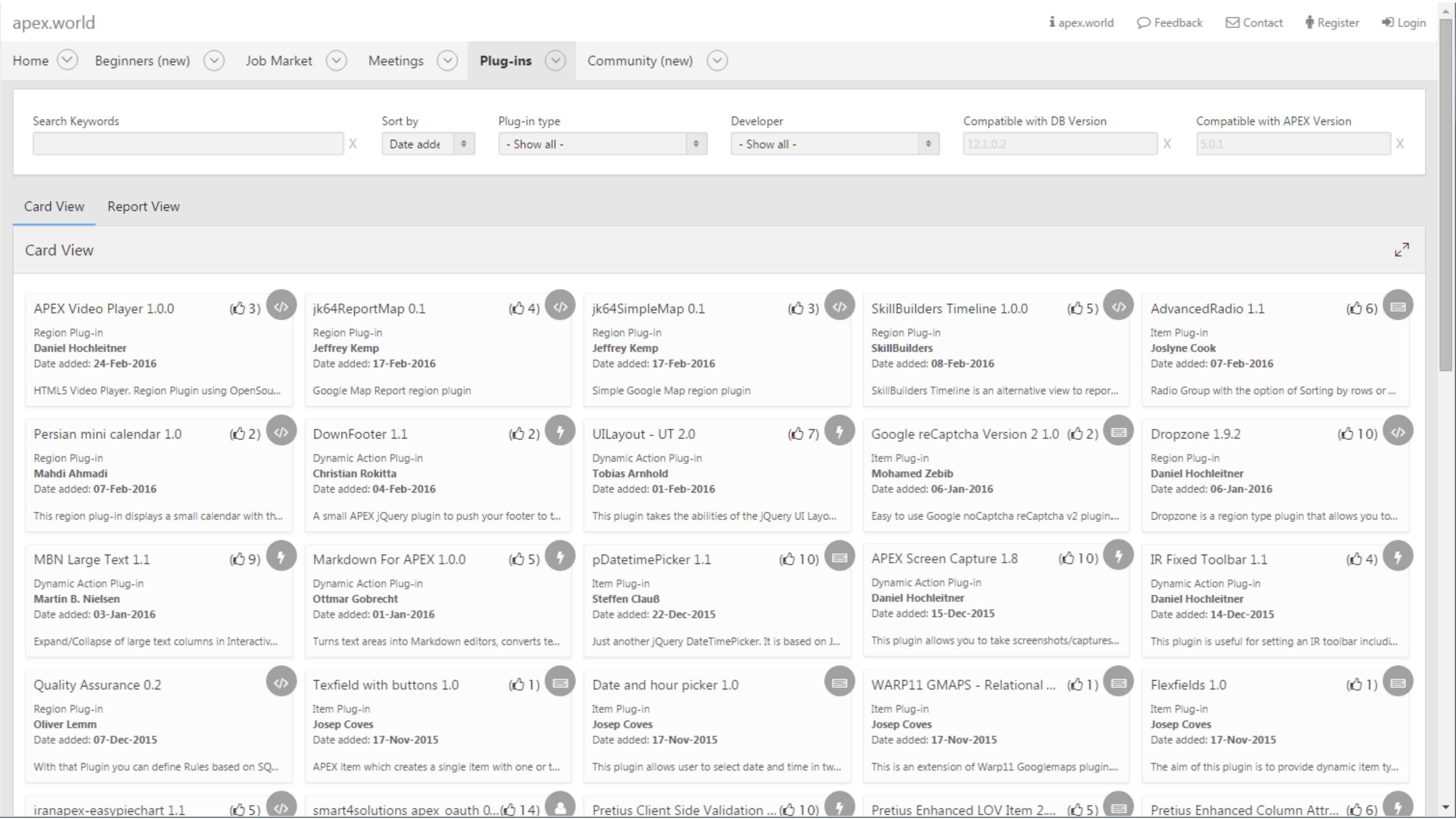
This region plug-in displays a small calendar with the specified date highlighted.

Read more...

DownFooter

APEX 4.2

☆☆☆☆☆



Search Keywords X

Sort by: Date adde

Plug-in type: - Show all -

Developer: - Show all -

Compatible with DB Version: 12.1.0.2 X

Compatible with APEX Version: 5.0.1 X

Card View

APEX Video Player 1.0.0 (👍 3)
Region Plug-in
Daniel Hochleitner
Date added: **24-Feb-2016**
HTML5 Video Player. Region Plugin using OpenSou...

jk64ReportMap 0.1 (👍 4)
Region Plug-in
Jeffrey Kemp
Date added: **17-Feb-2016**
Google Map Report region plugin

jk64SimpleMap 0.1 (👍 3)
Region Plug-in
Jeffrey Kemp
Date added: **17-Feb-2016**
Simple Google Map region plugin

SkillBuilders Timeline 1.0.0 (👍 5)
Region Plug-in
SkillBuilders
Date added: **08-Feb-2016**
SkillBuilders Timeline is an alternative view to repor...

AdvancedRadio 1.1 (👍 6)
Item Plug-in
Joslyne Cook
Date added: **07-Feb-2016**
Radio Group with the option of Sorting by rows or ...

Persian mini calendar 1.0 (👍 2)
Region Plug-in
Mahdi Ahmadi
Date added: **07-Feb-2016**
This region plug-in displays a small calendar with th...

DownFooter 1.1 (👍 2)
Dynamic Action Plug-in
Christian Rokitta
Date added: **04-Feb-2016**
A small APEX jQuery plugin to push your footer to t...

UILayout - UT 2.0 (👍 7)
Dynamic Action Plug-in
Tobias Arnhold
Date added: **01-Feb-2016**
This plugin takes the abilities of the jQuery UI Layo...

Google reCaptcha Version 2 1.0 (👍 2)
Item Plug-in
Mohamed Zebib
Date added: **06-Jan-2016**
Easy to use Google noCaptcha reCaptcha v2 plugin....

Dropzone 1.9.2 (👍 10)
Region Plug-in
Daniel Hochleitner
Date added: **06-Jan-2016**
Dropzone is a region type plugin that allows you to...

MBN Large Text 1.1 (👍 9)
Dynamic Action Plug-in
Martin B. Nielsen
Date added: **03-Jan-2016**
Expand/Collapse of large text columns in Interactiv...

Markdown For APEX 1.0.0 (👍 5)
Dynamic Action Plug-in
Ottmar Gobrecht
Date added: **01-Jan-2016**
Turns text areas into Markdown editors, converts te...

pDatetimePicker 1.1 (👍 10)
Item Plug-in
Steffen Clauß
Date added: **22-Dec-2015**
Just another jQuery DateTimePicker. It is based on J...

APEX Screen Capture 1.8 (👍 10)
Dynamic Action Plug-in
Daniel Hochleitner
Date added: **15-Dec-2015**
This plugin allows you to take screenshots/captures...

IR Fixed Toolbar 1.1 (👍 4)
Dynamic Action Plug-in
Daniel Hochleitner
Date added: **14-Dec-2015**
This plugin is useful for setting an IR toolbar includi...

Quality Assurance 0.2
Region Plug-in
Oliver Lemm
Date added: **07-Dec-2015**
With that Plugin you can define Rules based on SQ...

Texfield with buttons 1.0 (👍 1)
Item Plug-in
Josep Coves
Date added: **17-Nov-2015**
APEX item which creates a single item with one or t...

Date and hour picker 1.0
Item Plug-in
Josep Coves
Date added: **17-Nov-2015**
This plugin allows user to select date and time in tw...

WARP11 GMAPS - Relational ... (👍 1)
Item Plug-in
Josep Coves
Date added: **17-Nov-2015**
This is an extension of Warp11 Googlemaps plugin....

Flexfields 1.0 (👍 1)
Item Plug-in
Josep Coves
Date added: **17-Nov-2015**
The aim of this plugin is to provide dynamic item ty...

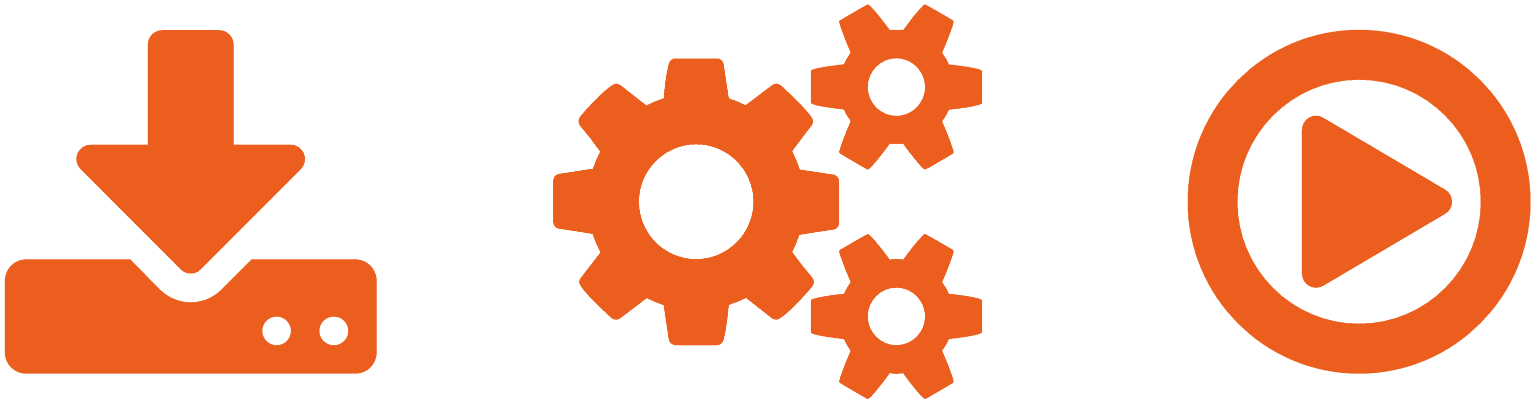
iranapex-easypiechart 1.1 (👍 5)

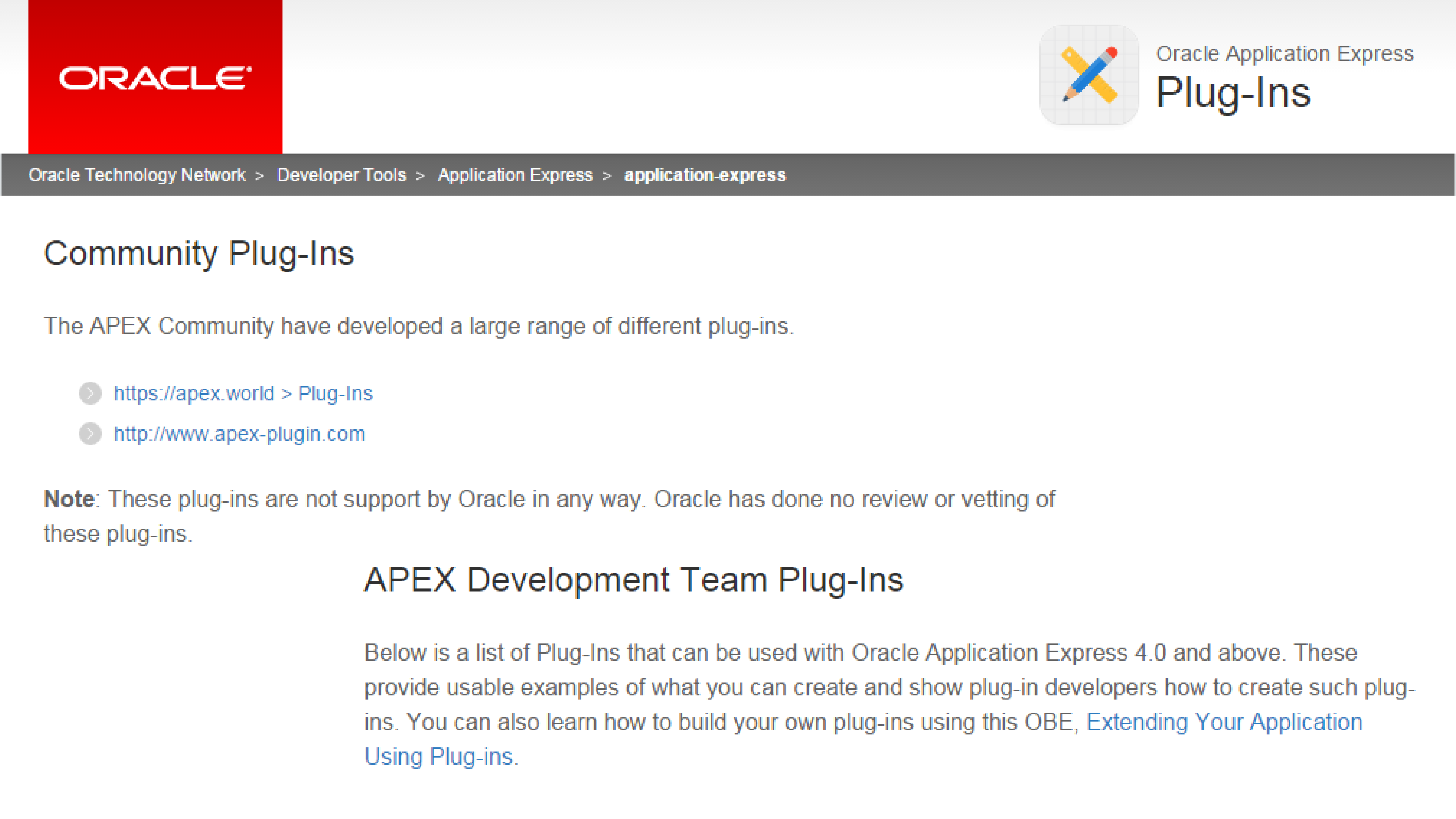
smart4solutions apex oauth 0...(👍 14)

Pretius Client Side Validation ... (👍 10)

Pretius Enhanced LOV Item 2.... (👍 5)

Pretius Enhanced Column Attr... (👍 6)





ORACLE
Oracle Application Express
Plug-Ins
Oracle Technology Network > Developer Tools > Application Express > application-express

Community Plug-Ins

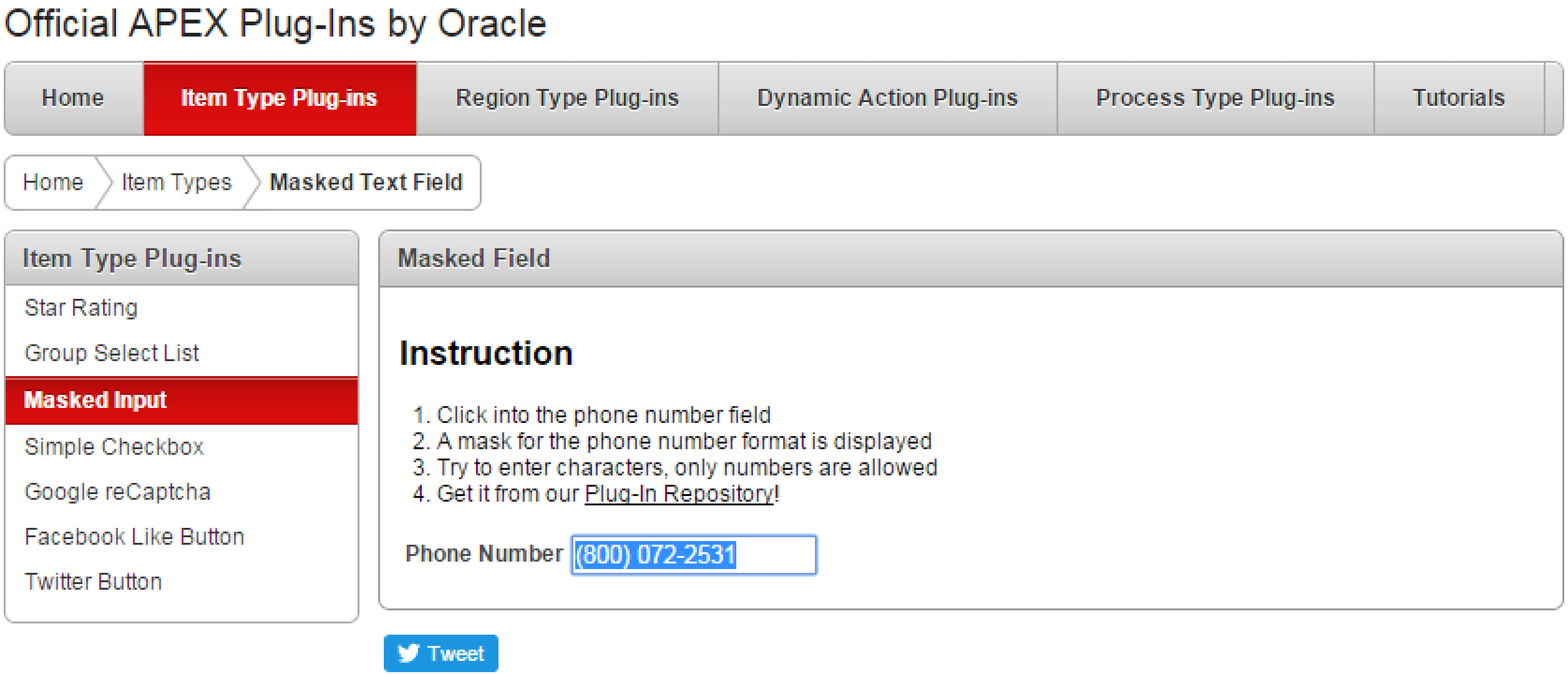
The APEX Community have developed a large range of different plug-ins.

- https://apex.world > Plug-Ins
- http://www.apex-plugin.com

Note: These plug-ins are not support by Oracle in any way. Oracle has done no review or vetting of these plug-ins.

APEX Development Team Plug-Ins

Below is a list of Plug-Ins that can be used with Oracle Application Express 4.0 and above. These provide usable examples of what you can create and show plug-in developers how to create such plug-ins. You can also learn how to build your own plug-ins using this OBE, Extending Your Application Using Plug-ins.



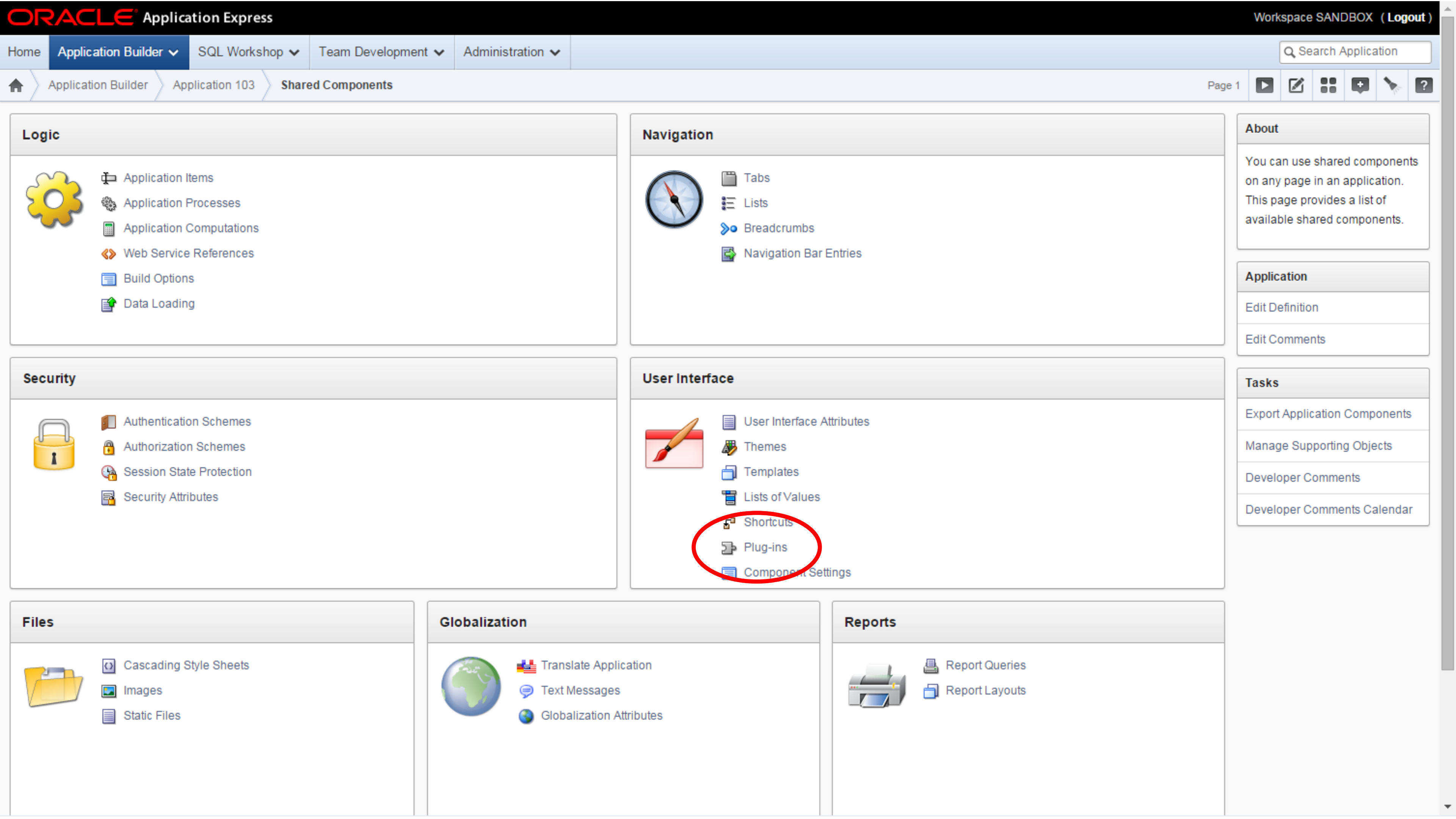
Official APEX Plug-Ins by Oracle

Masked Field

Instruction

1. Click into the phone number field
2. A mask for the phone number format is displayed
3. Try to enter characters, only numbers are allowed
4. Get it from our Plug-In Repository!

Phone Number (800) 072-2531



ORACLE Application Express

Workspace SANDBOX (Logout)

Home | Application Builder | SQL Workshop | Team Development | Administration

Search Application

Application Builder > Application 103 > Shared Components

Page 1

Logic

- Application Items
- Application Processes
- Application Computations
- Web Service References
- Build Options
- Data Loading

Navigation

- Tabs
- Lists
- Breadcrumbs
- Navigation Bar Entries

About

You can use shared components on any page in an application. This page provides a list of available shared components.

Application

- Edit Definition
- Edit Comments

Tasks

- Export Application Components
- Manage Supporting Objects
- Developer Comments
- Developer Comments Calendar

Security

- Authentication Schemes
- Authorization Schemes
- Session State Protection
- Security Attributes

User Interface

- User Interface Attributes
- Themes
- Templates
- Lists of Values
- Shortcuts
- Plug-ins
- Component Settings

Files

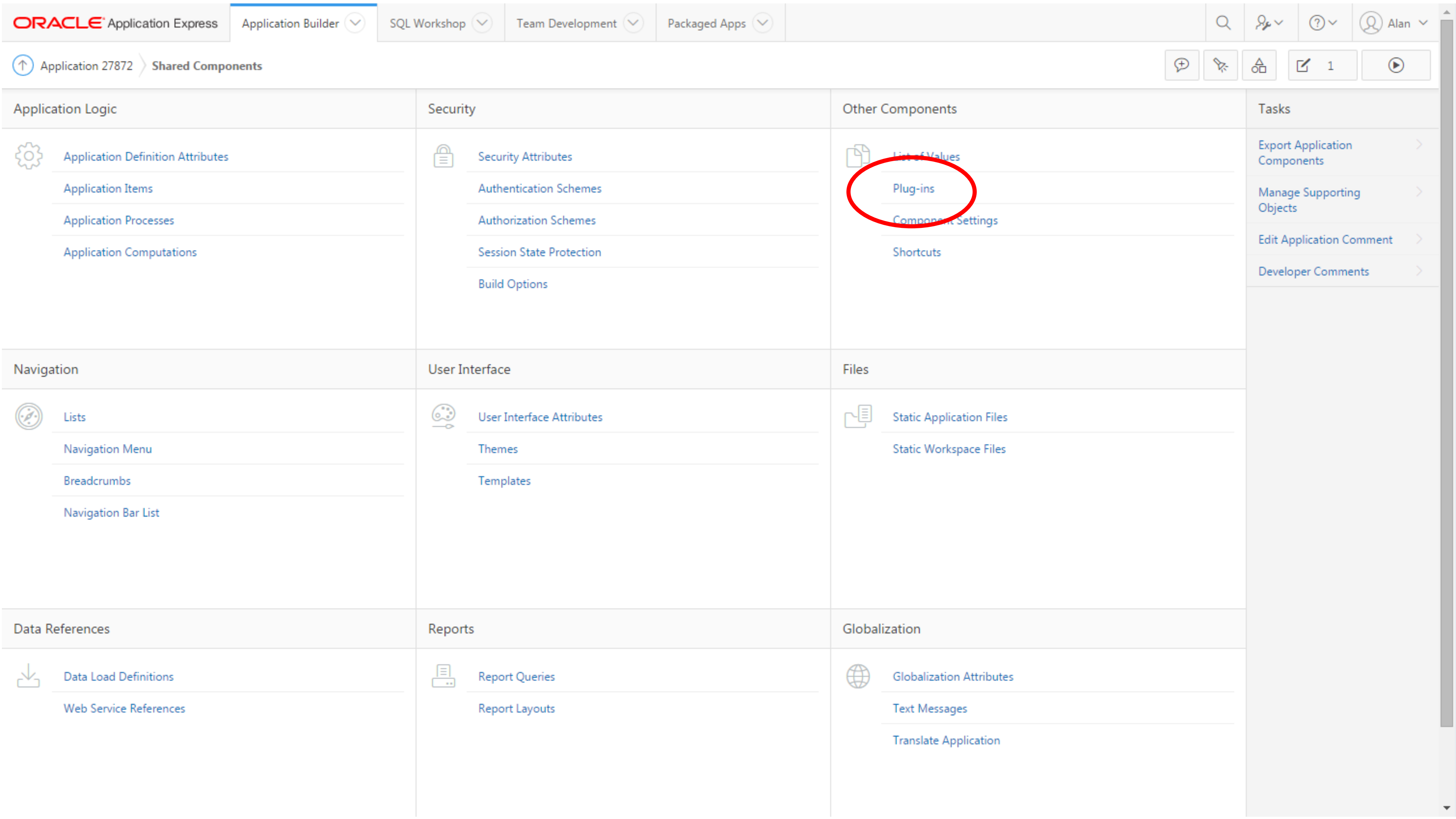
- Cascading Style Sheets
- Images
- Static Files

Globalization

- Translate Application
- Text Messages
- Globalization Attributes

Reports

- Report Queries
- Report Layouts



Application Logic

- Application Definition Attributes
- Application Items
- Application Processes
- Application Computations

Security

- Security Attributes
- Authentication Schemes
- Authorization Schemes
- Session State Protection
- Build Options

Other Components

- List of Values
- Plug-ins
- Component Settings
- Shortcuts

Tasks

- Export Application Components
- Manage Supporting Objects
- Edit Application Comment
- Developer Comments

Navigation

- Lists
- Navigation Menu
- Breadcrumbs
- Navigation Bar List

User Interface

- User Interface Attributes
- Themes
- Templates

Files

- Static Application Files
- Static Workspace Files

Data References

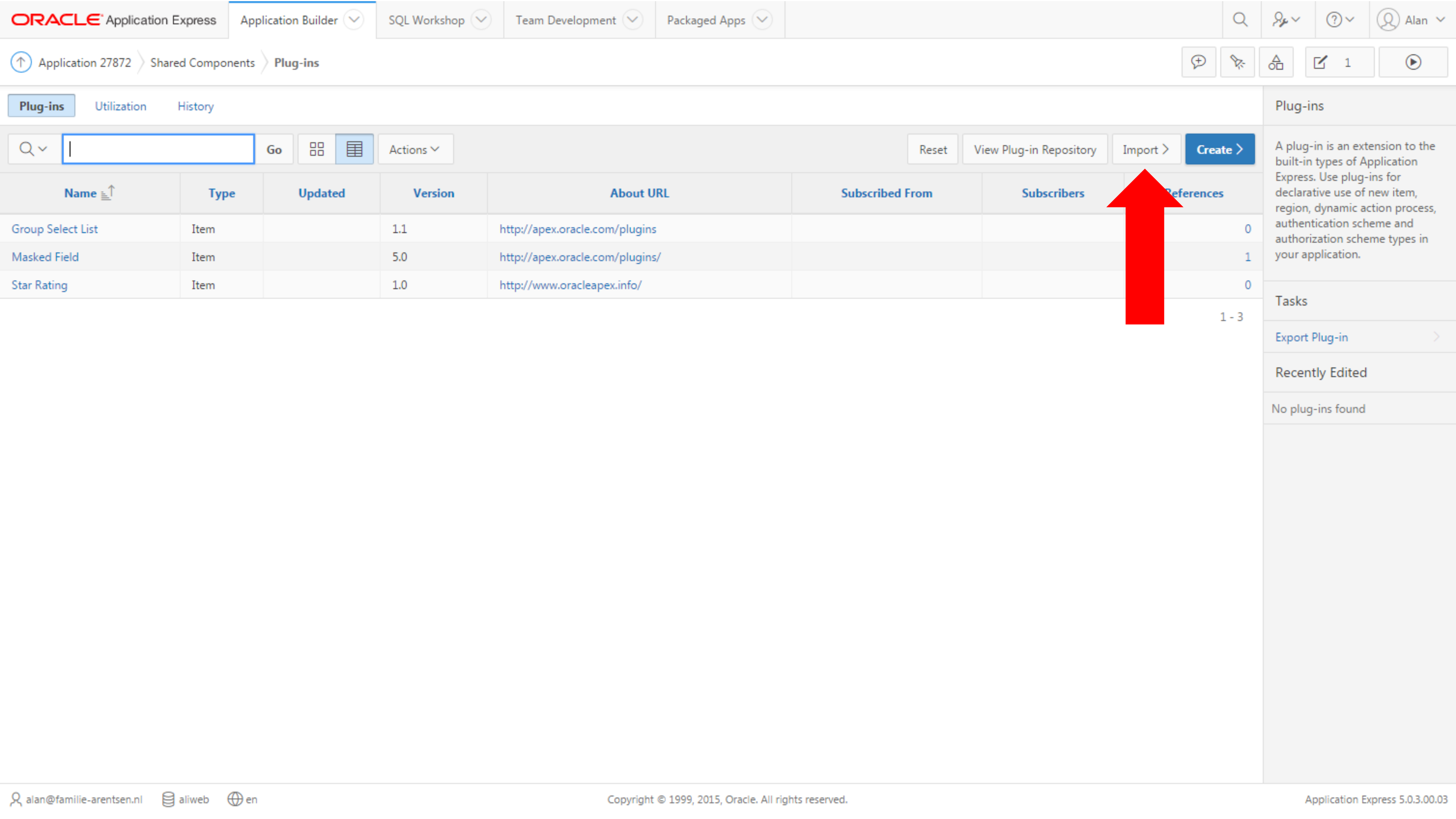
- Data Load Definitions
- Web Service References

Reports

- Report Queries
- Report Layouts

Globalization

- Globalization Attributes
- Text Messages
- Translate Application



Name	Type	Updated	Version	About URL	Subscribed From	Subscribers	References
Group Select List	Item		1.1	http://apex.oracle.com/plugins			0
Masked Field	Item		5.0	http://apex.oracle.com/plugins/			1
Star Rating	Item		1.0	http://www.oracleapex.info/			0

1 - 3

Plug-ins

A plug-in is an extension to the built-in types of Application Express. Use plug-ins for declarative use of new item, region, dynamic action process, authentication scheme and authorization scheme types in your application.

Tasks

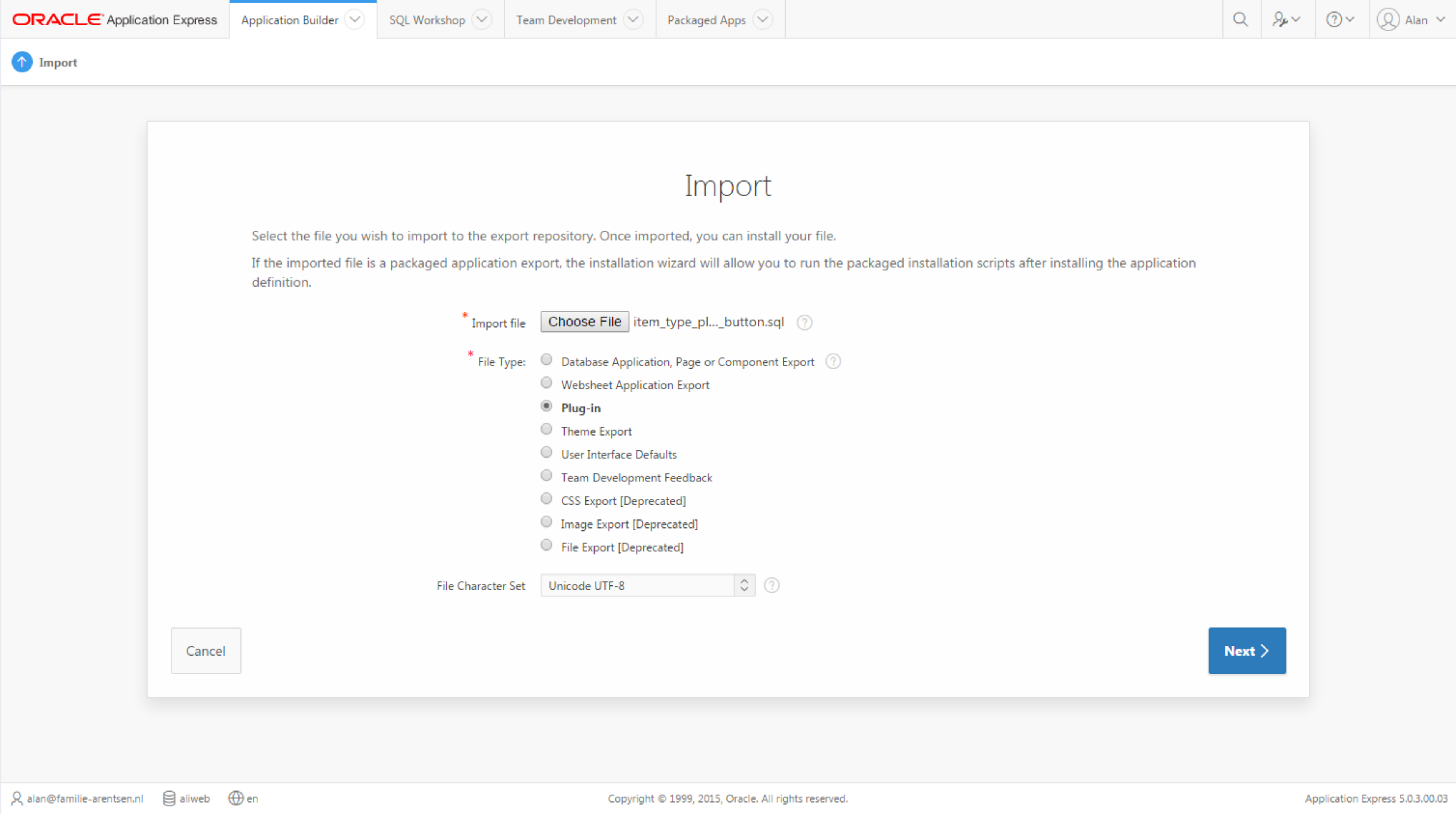
Export Plug-in

Recently Edited

No plug-ins found

Copyright © 1999, 2015, Oracle. All rights reserved.

Application Express 5.0.3.00.03



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 packaged installation scripts after installing the application definition.

* Import file: Choose File item_type_pl..._button.sql

* File Type:

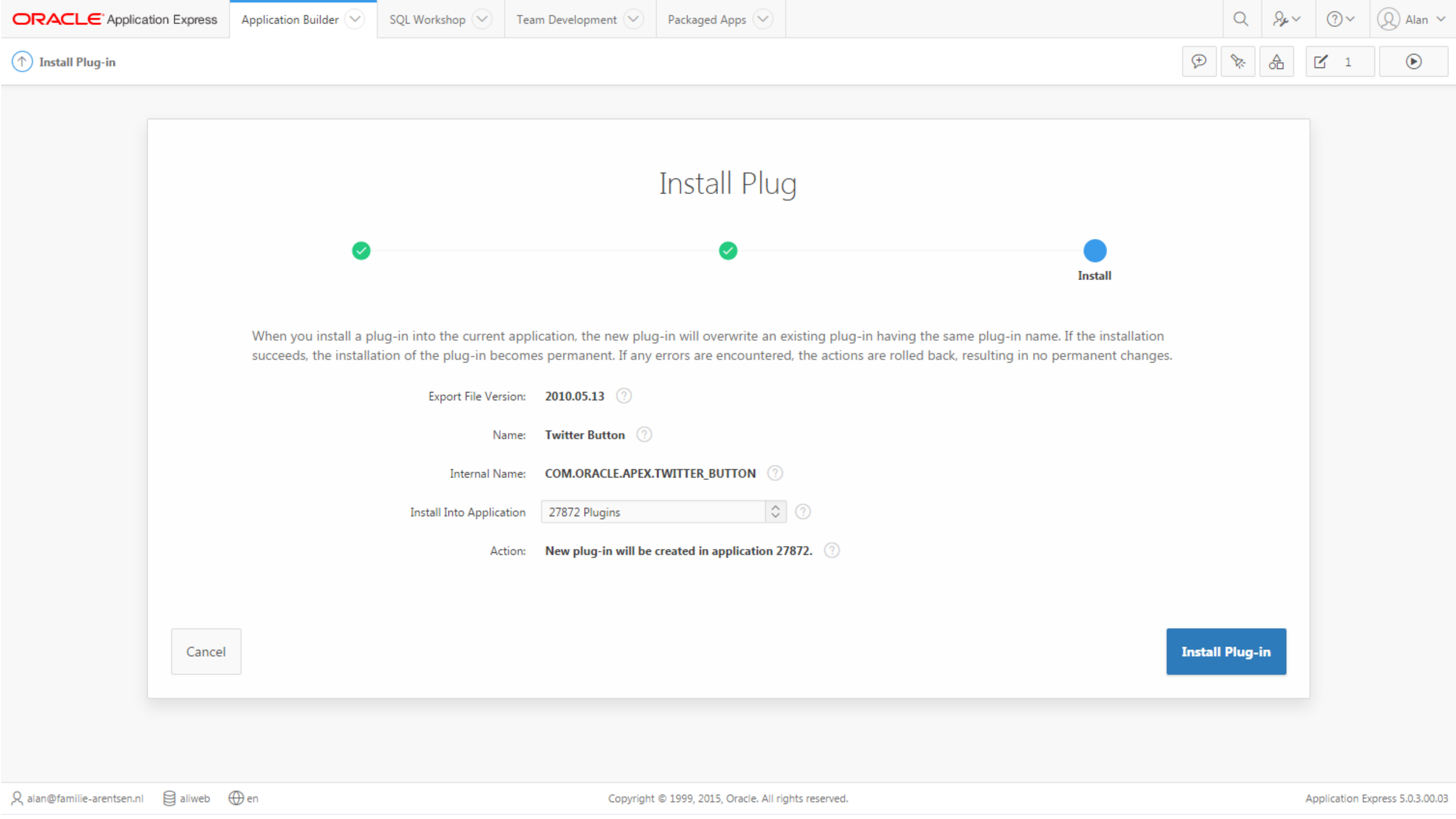
- ○ Database Application, Page or Component Export
- ○ Websheet Application Export
- ● **Plug-in**
- ○ Theme Export
- ○ User Interface Defaults
- ○ Team Development Feedback
- ○ CSS Export [Deprecated]
- ○ Image Export [Deprecated]
- ○ File Export [Deprecated]

File Character Set: Unicode UTF-8

alan@familie-arentsen.nl aliweb en

Copyright © 1999, 2015, Oracle. All rights reserved.

Application Express 5.0.3.00.03



Install Plug

When you install a plug-in into the current application, the new plug-in will overwrite an existing plug-in having the same plug-in name. If the installation succeeds, the installation of the plug-in becomes permanent. If any errors are encountered, the actions are rolled back, resulting in no permanent changes.

Export File Version: 2010.05.13

Name: Twitter Button

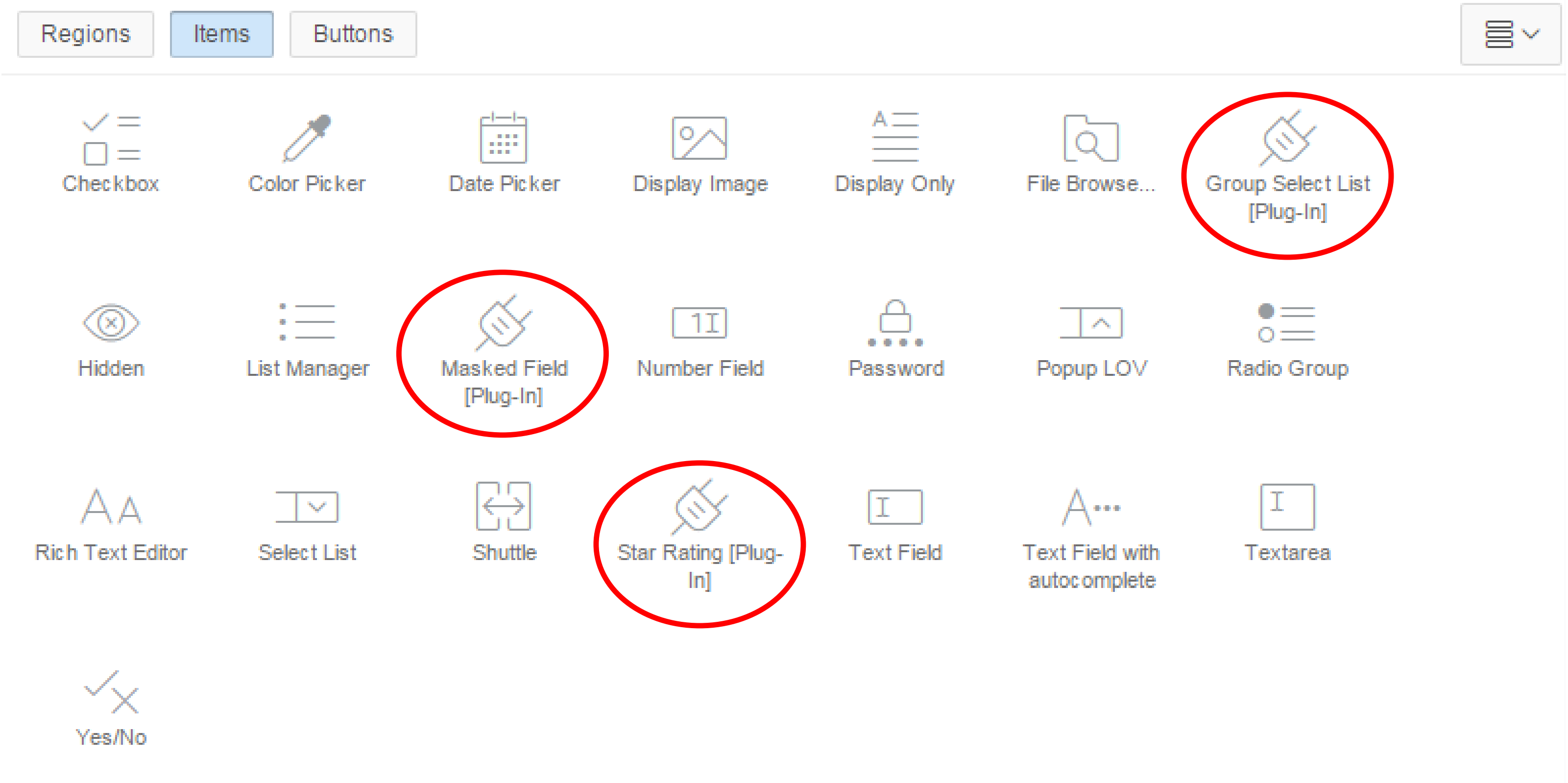
Internal Name: COM.ORACLE.APEX.TWITTER_BUTTON

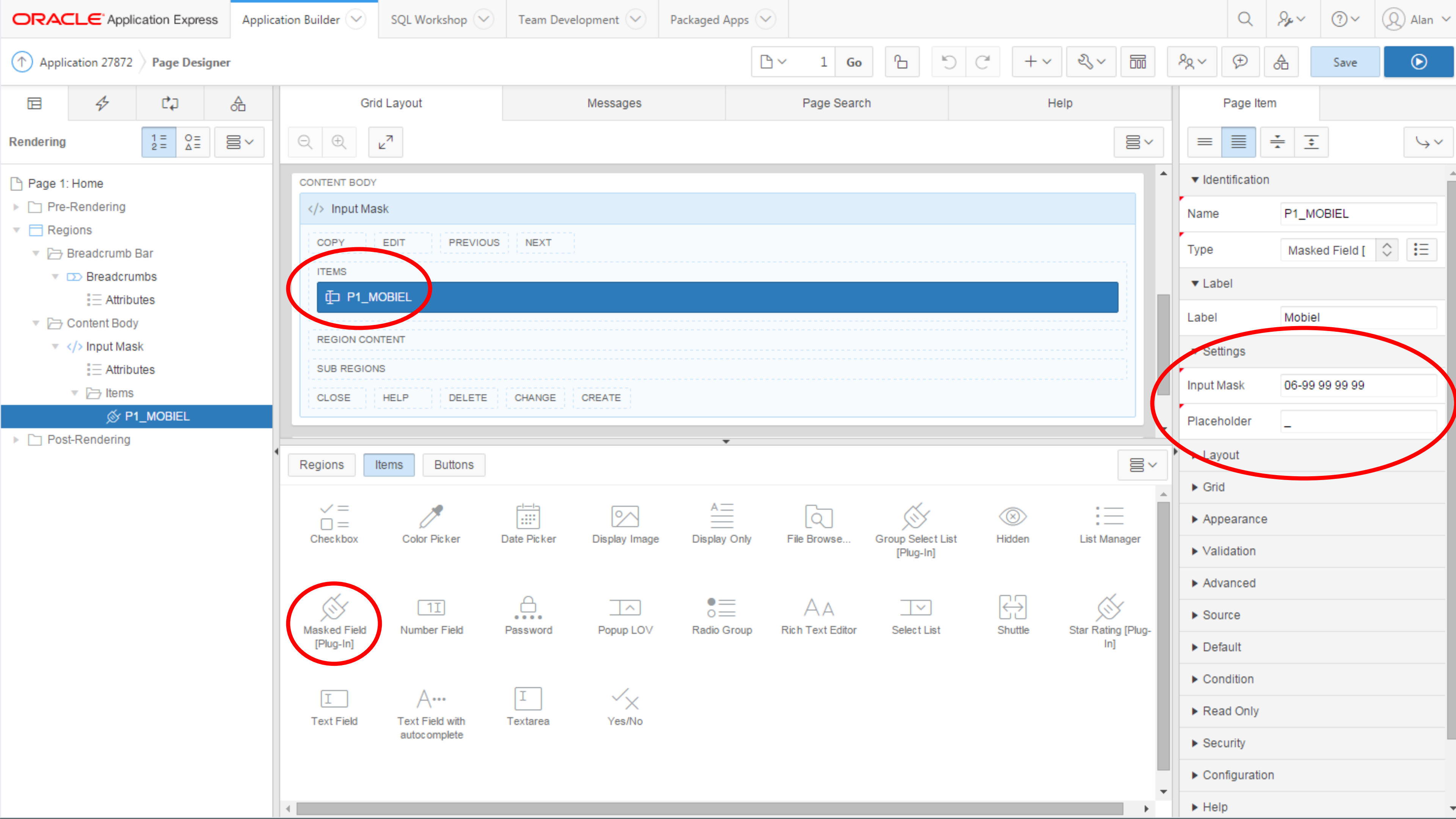
Install Into Application 27872 Plugins

Action: New plug-in will be created in application 27872.

Copyright © 1999, 2015, Oracle. All rights reserved.

Application Express 5.0.3.00.03


Regions
Items
Buttons
Checkbox
Color Picker
Date Picker
Display Image
Display Only
File Browse...
Group Select List [Plug-In]
Hidden
List Manager
Masked Field [Plug-In]
Number Field
Password
Popup LOV
Radio Group
Rich Text Editor
Select List
Shuttle
Star Rating [Plug-In]
Text Field
Text Field with autocomplete
Textarea
Yes/No



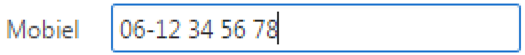
Name: P1_MOBIEL

Type: Masked Field [

Label: Mobiel

Input Mask: 06-99 99 99 99

Placeholder: _



Mobiel 06-12 34 56 78

DEMO

- ☑ **Plug-in Repository**
- 🔌 **Create from scratch**
- 🔌 **Copy of existing Plug-in**

Create from scratch

```
apex.item("P1_STANDARD_CHECK").getValue()
```

☑

```
'On'
```

```
apex.item("P1_STANDARD_CHECK").getValue()
```

□

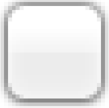
```
null
```

```
apex.item("P1_STANDARD_CHECK").getValue()
```

□

~~null~~

```
apex.item("P1_STANDARD_CHECK").getValue()
```



```
'Off'
```

```
<input type="checkbox"
       id="P1_PLUGIN_CHECK"
       class="custom_checkbox"
       value="" />


<input type="hidden"
       id="P1_PLUGIN_CHECK_HIDDEN"
       name="p_t01"
       value="Off" />


<label for="P1_PLUGIN_CHECK">Does this work?</label>
```

HTML

```
<input type="checkbox"
       id="P1_PLUGIN_CHECK"
       class="custom_checkbox"
       value="" />


<input type="hidden"
       id="P1_PLUGIN_CHECK_HIDDEN"
       name="p_t01"
       value="Off" />


<label for="P1_PLUGIN_CHECK">Does this work?</label>
```

```
<input type="checkbox"
       id="P1_PLUGIN_CHECK"
       class="custom_checkbox"
       value="" />

<input type="hidden"
       id="P1_PLUGIN_CHECK_HIDDEN"
       name="p_t01"
       value="Off" />

<label for="P1_PLUGIN_CHECK">Does this work?</label>
```

```
function custom_checkbox (pPageItemName, pOptions)
{
  var g$Checkbox = $('#' + pPageItemName),
      g$Hidden   = $('#' + pPageItemName + '_HIDDEN');

  function syncHiddenField() {
        g$Hidden.val((g$Checkbox.is(':checked') === true) ?
                           pOptions.checked :
                                 pOptions.unchecked);
  };

  ...


};
```

```
function custom_checkbox (pPageItemName, pOptions)
{
  var g$Checkbox = $('#' + pPageItemName),
      g$Hidden   = $('#' + pPageItemName + '_HIDDEN');

  function syncHiddenField() {
        g$Hidden.val((g$Checkbox.is(':checked') === true) ?
                           pOptions.checked :
                                 pOptions.unchecked);
  };

  ...


};
```

```
function custom_checkbox (pPageItemName, pOptions)
{
  var g$Checkbox = $('#' + pPageItemName),
      g$Hidden   = $('#' + pPageItemName + '_HIDDEN');

  function syncHiddenField() {
        g$Hidden.val((g$Checkbox.is(':checked') === true) ?
                           pOptions.checked :
                                 pOptions.unchecked);
  };

  ...


};
```

```
function custom_checkbox (pPageItemName, pOptions)
{
  var g$Checkbox = $('#' + pPageItemName),
      g$Hidden   = $('#' + pPageItemName + '_HIDDEN');

  function syncHiddenField() {
        g$Hidden.val((g$Checkbox.is(':checked') === true) ?
                           pOptions.checked :
                                  pOptions.unchecked);
  };

  ...


};
```

```
function custom_checkbox (pPageItemName, pOptions)
{

  ...

  g$Checkbox.change(syncHiddenField);

  ...

};
```

```
function custom_checkbox (pPageItemName, pOptions)
{
  ...

  apex.widget.initPageItem(pPageItemName, {
     setValue: function(pValue) {
        g$Checkbox.attr('checked'
                       ,(pValue === pOptions.checked));
        syncHiddenField();
     },
     getValue:function() {
        return g$Hidden.val();
     }
  });
};
```

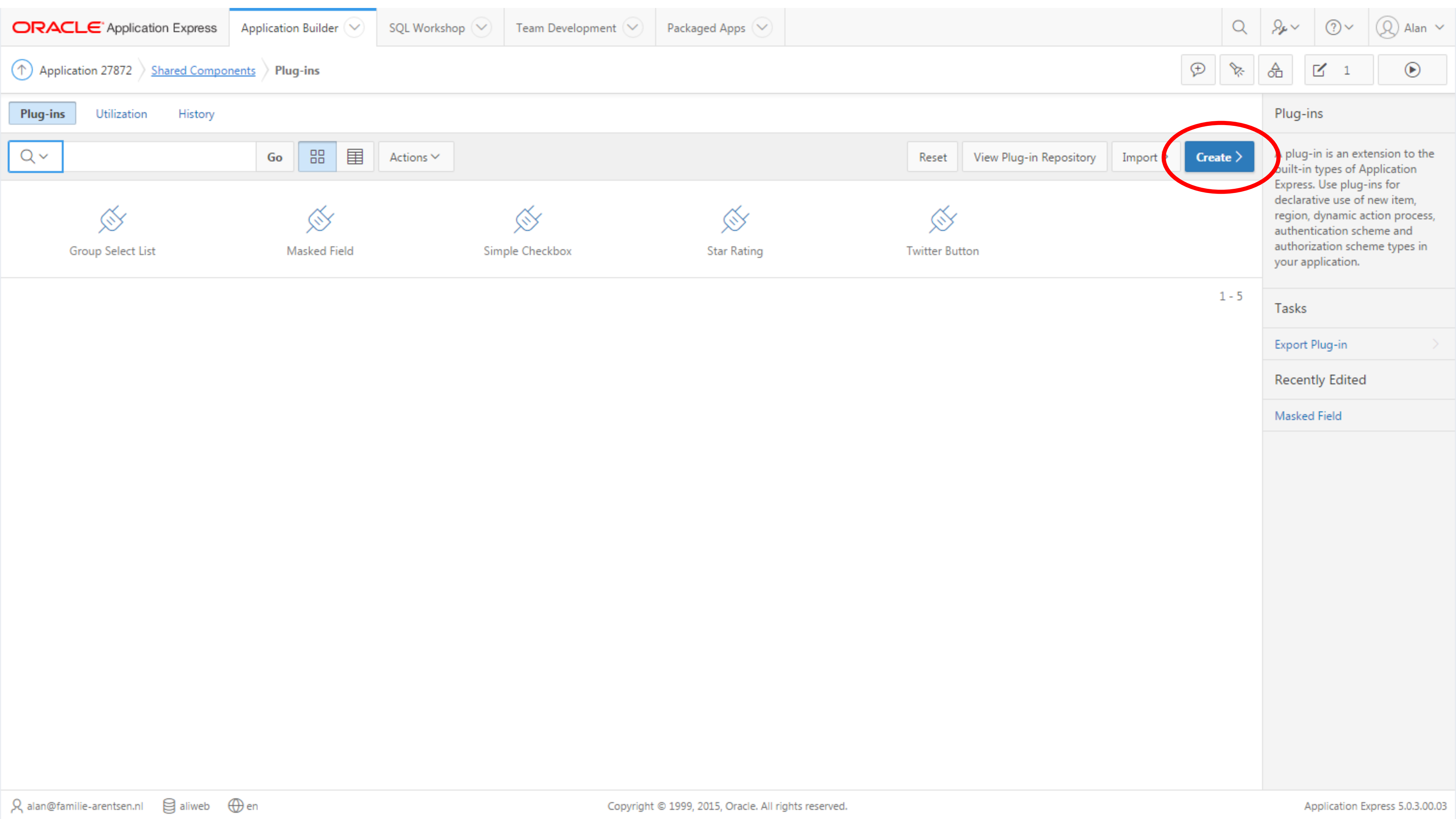
```
function custom_checkbox (pPageItemName, pOptions)
{
  ...

  apex.widget.initPageItem(pPageItemName, {
     setValue: function(pValue) {
        g$Checkbox.attr('checked'
                       ,(pValue === pOptions.checked));
        syncHiddenField();
     },
     getValue:function() {
        return g$Hidden.val();
     }
  });
};
```

```
function custom_checkbox (pPageItemName, pOptions)
{
  ...

  apex.widget.initPageItem(pPageItemName, {
     setValue: function(pValue) {
        g$Checkbox.attr('checked'
                       ,(pValue === pOptions.checked));
        syncHiddenField();
     },
     getValue:function() {
        return g$Hidden.val();
     }
  });
};
```

☑ □

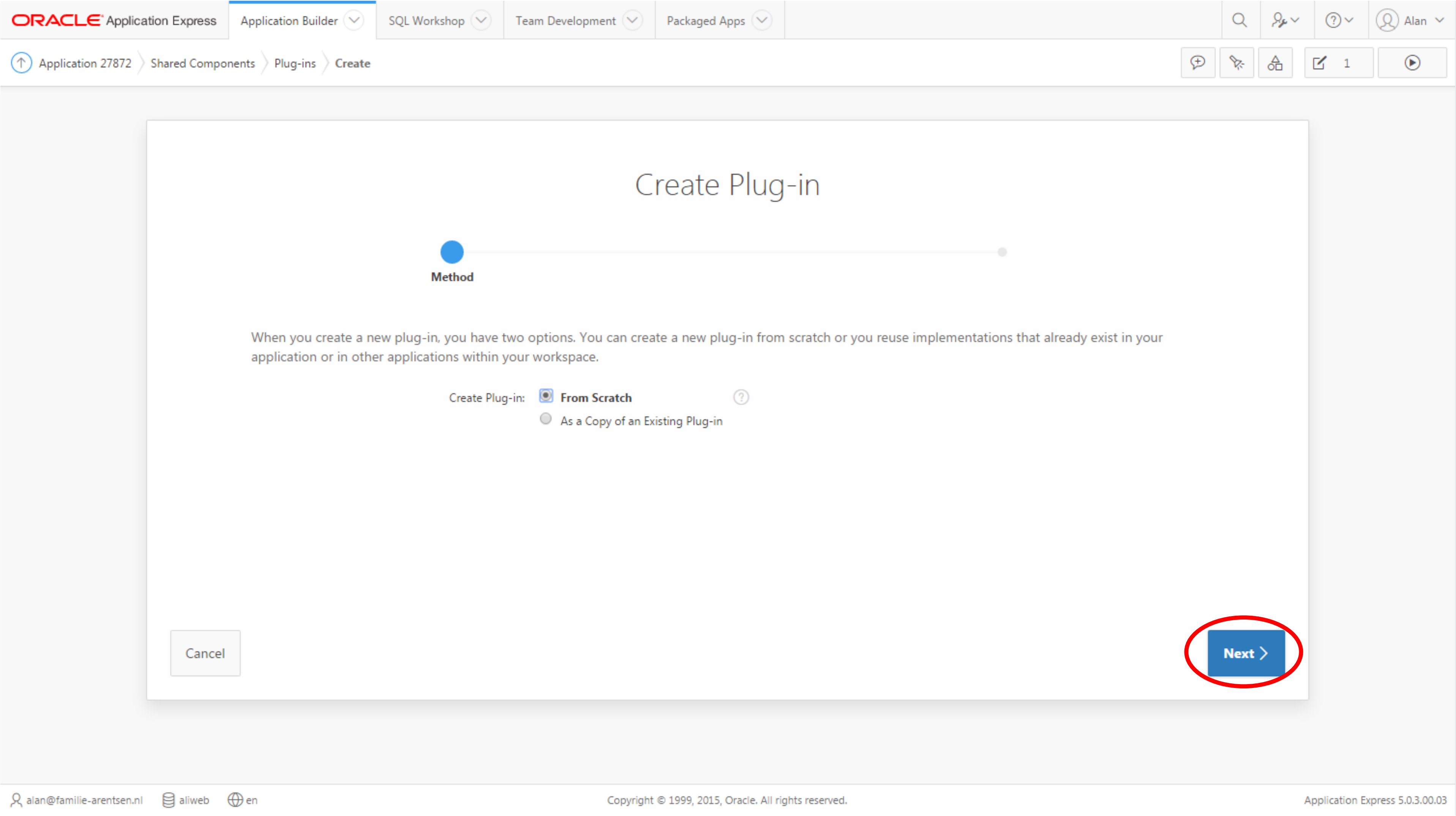
HTML

JavaScript



A plug-in is an extension to the built-in types of Application Express. Use plug-ins for declarative use of new item, region, dynamic action process, authentication scheme and authorization scheme types in your application.

Copyright © 1999, 2015, Oracle. All rights reserved.

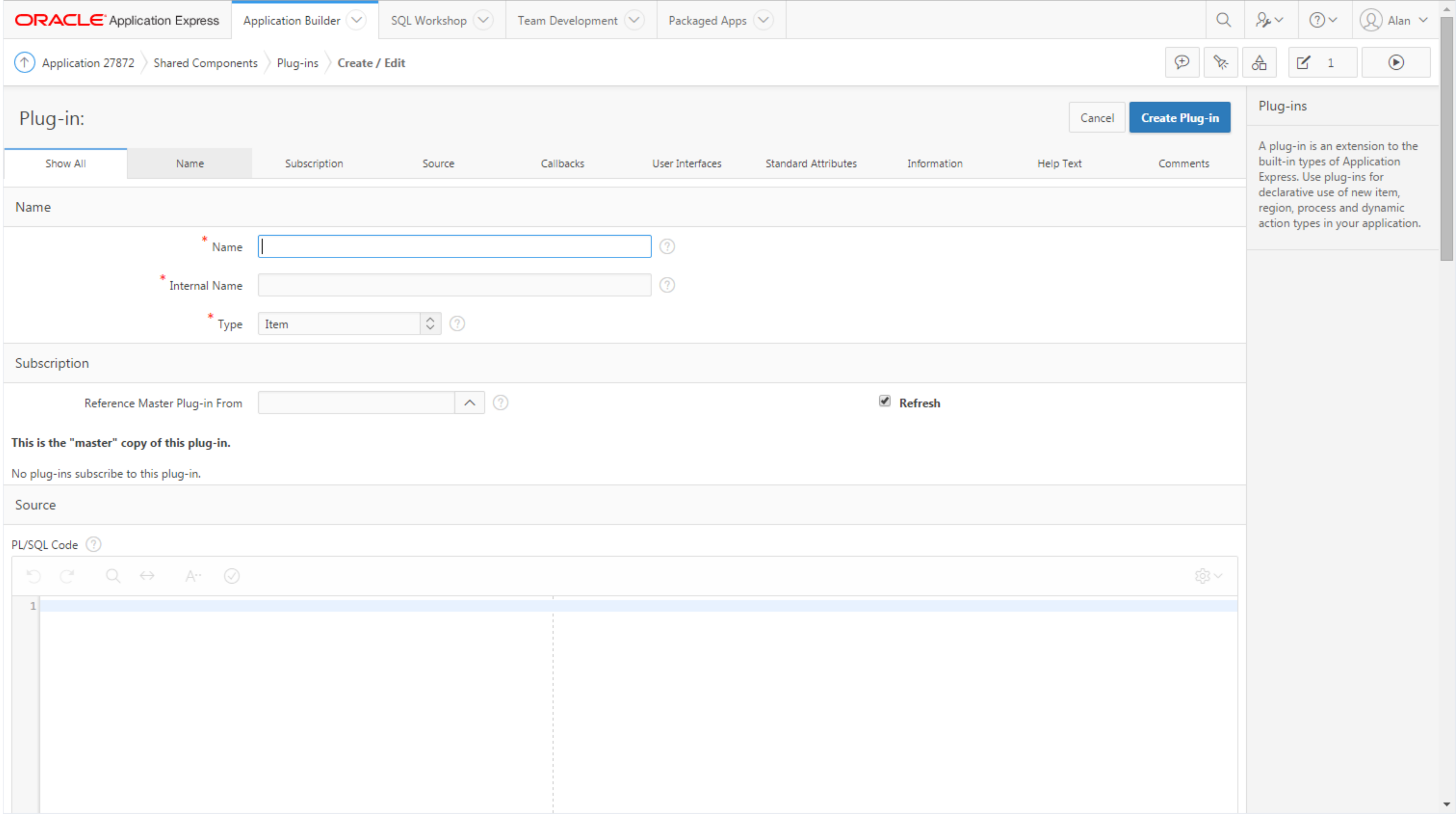


Create Plug-in

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

Create Plug-in: ● From Scratch

○ As a Copy of an Existing Plug-in



Plug-in:

Name

* Name

* Internal Name

* Type: Item

Subscription

Reference Master Plug-in From

☑ **Refresh**

This is the "master" copy of this plug-in.

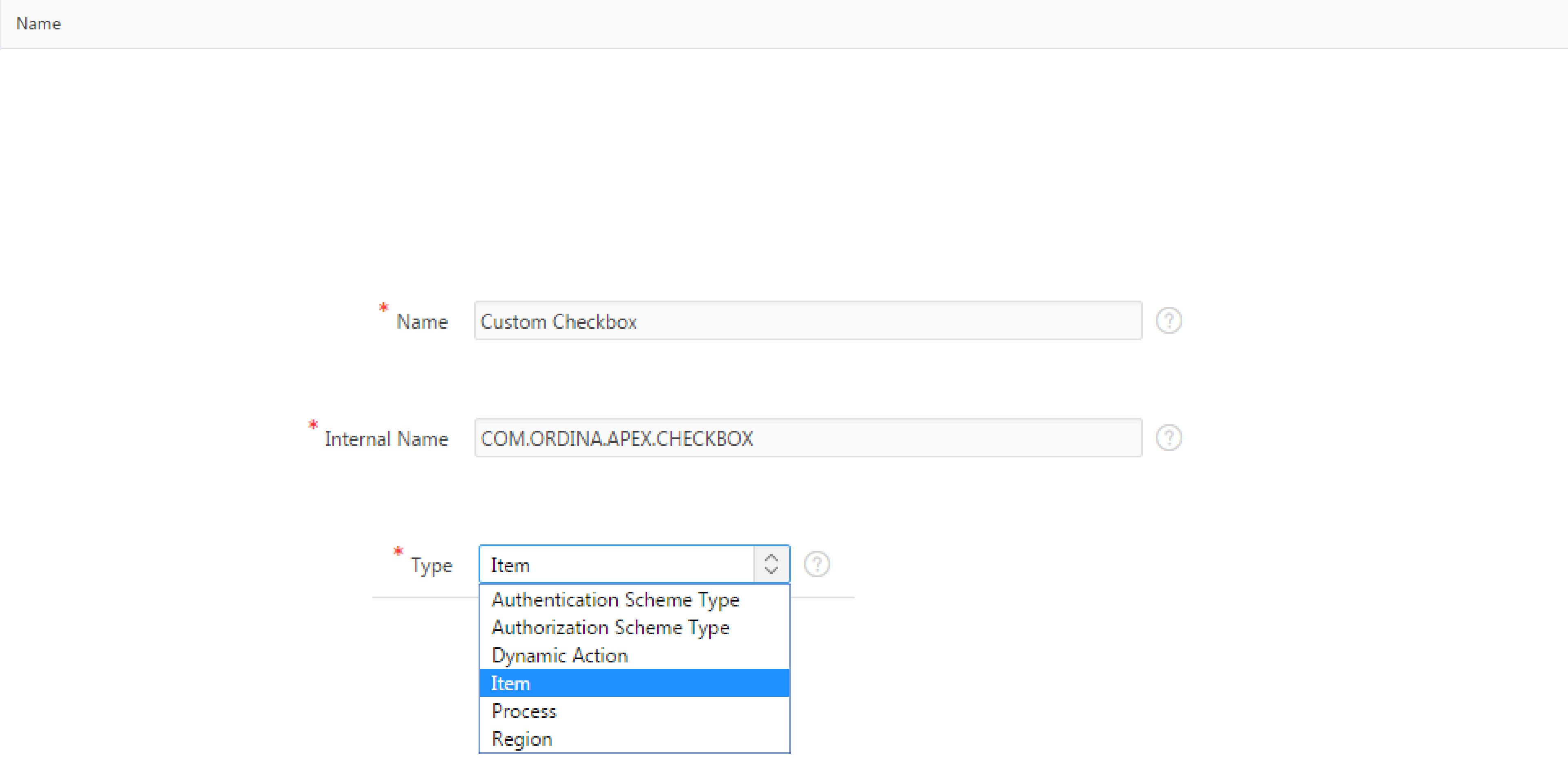
No plug-ins subscribe to this plug-in.

Source

PL/SQL Code

Plug-ins

A plug-in is an extension to the built-in types of Application Express. Use plug-ins for declarative use of new item, region, process and dynamic action types in your application.

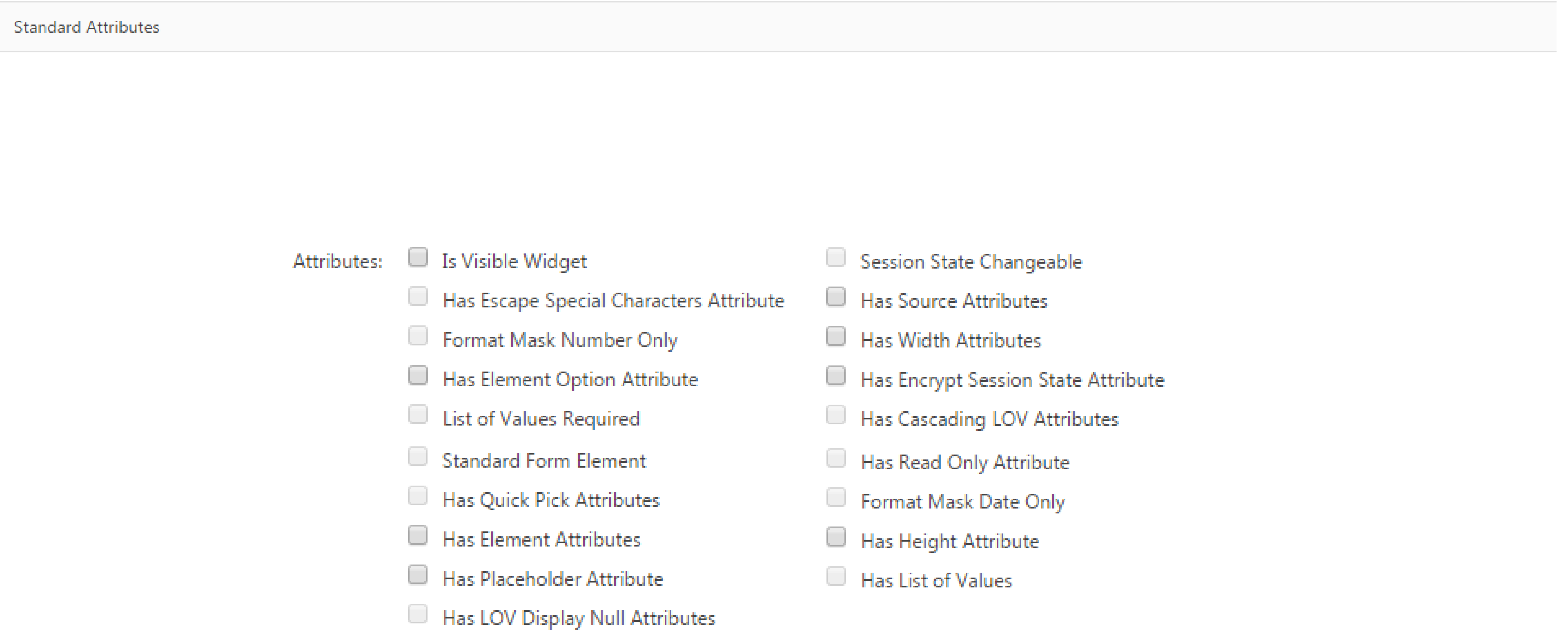


Name

* Name: Custom Checkbox

* Internal Name: COM.ORDINA.APEX.CHECKBOX

* Type: Item

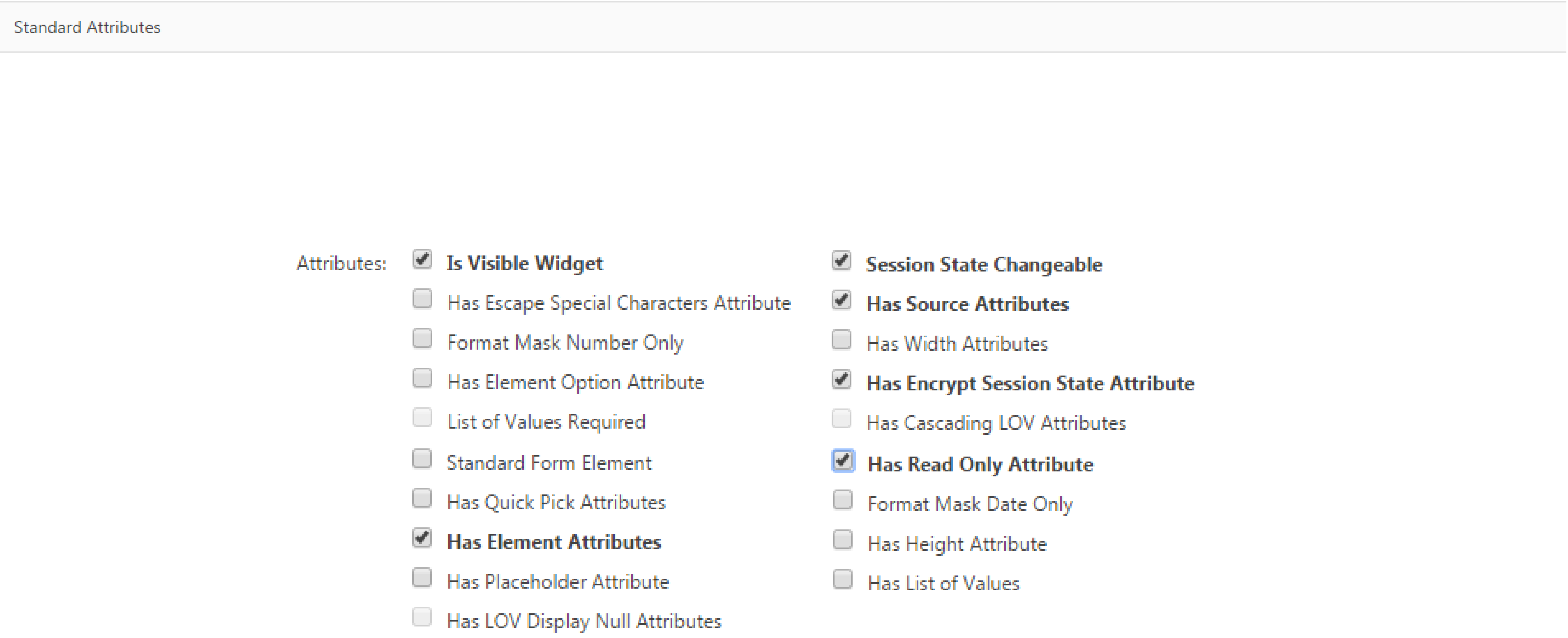


Standard Attributes

Attributes:

□ Is Visible Widget
□ Has Escape Special Characters Attribute
□ Format Mask Number Only
□ Has Element Option Attribute
□ List of Values Required
□ Standard Form Element
□ Has Quick Pick Attributes
□ Has Element Attributes
□ Has Placeholder Attribute
□ Has LOV Display Null Attributes

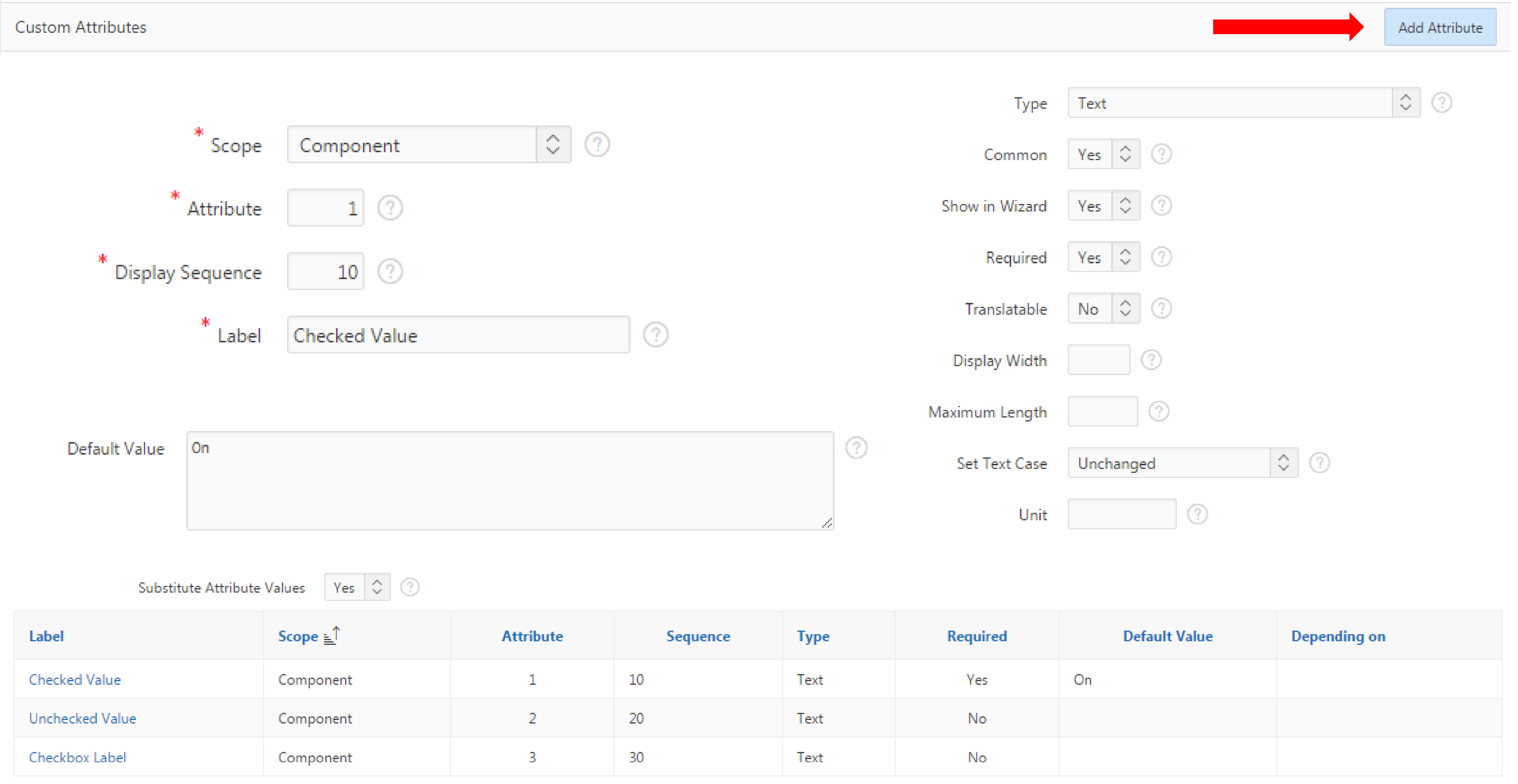
□ Session State Changeable
□ Has Source Attributes
□ Has Width Attributes
□ Has Encrypt Session State Attribute
□ Has Cascading LOV Attributes
□ Has Read Only Attribute
□ Format Mask Date Only
□ Has Height Attribute
□ Has List of Values



Standard Attributes

Attributes:

- ☑ **Is Visible Widget**
- ☐ Has Escape Special Characters Attribute
- ☐ Format Mask Number Only
- ☐ Has Element Option Attribute
- ☐ List of Values Required
- ☐ Standard Form Element
- ☐ Has Quick Pick Attributes
- ☑ **Has Element Attributes**
- ☐ Has Placeholder Attribute
- ☐ Has LOV Display Null Attributes
- ☑ **Session State Changeable**
- ☑ **Has Source Attributes**
- ☐ Has Width Attributes
- ☑ **Has Encrypt Session State Attribute**
- ☐ Has Cascading LOV Attributes
- ☑ **Has Read Only Attribute**
- ☐ Format Mask Date Only
- ☐ Has Height Attribute
- ☐ Has List of Values



Custom Attributes

* Scope Component

* Attribute 1

* Display Sequence 10

* Label Checked Value

Default Value On

Type Text

Common Yes

Show in Wizard Yes

Required Yes

Translatable No

Display Width

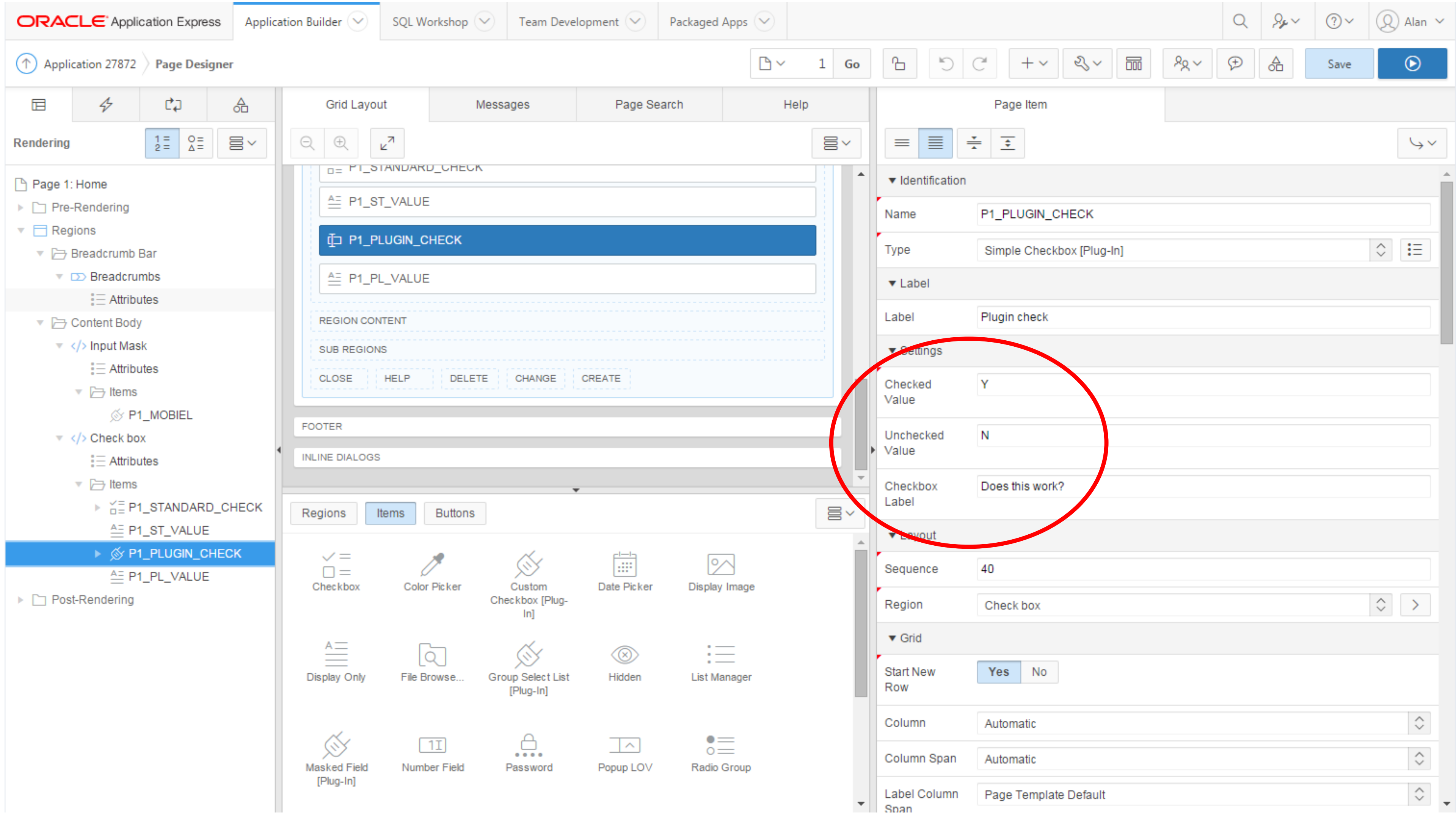
Maximum Length

Set Text Case Unchanged

Unit

Substitute Attribute Values Yes

Label	Scope	Attribute	Sequence	Type	Required	Default Value	Depending on
Checked Value	Component	1	10	Text	Yes	On	
Unchecked Value	Component	2	20	Text	No		
Checkbox Label	Component	3	30	Text	No		



Identification

Name: P1_PLUGIN_CHECK

Type: Simple Checkbox [Plug-In]

Label

Label: Plugin check

Settings

Checked Value: Y

Unchecked Value: N

Checkbox Label: Does this work?

Layout

Sequence: 40

Region: Check box

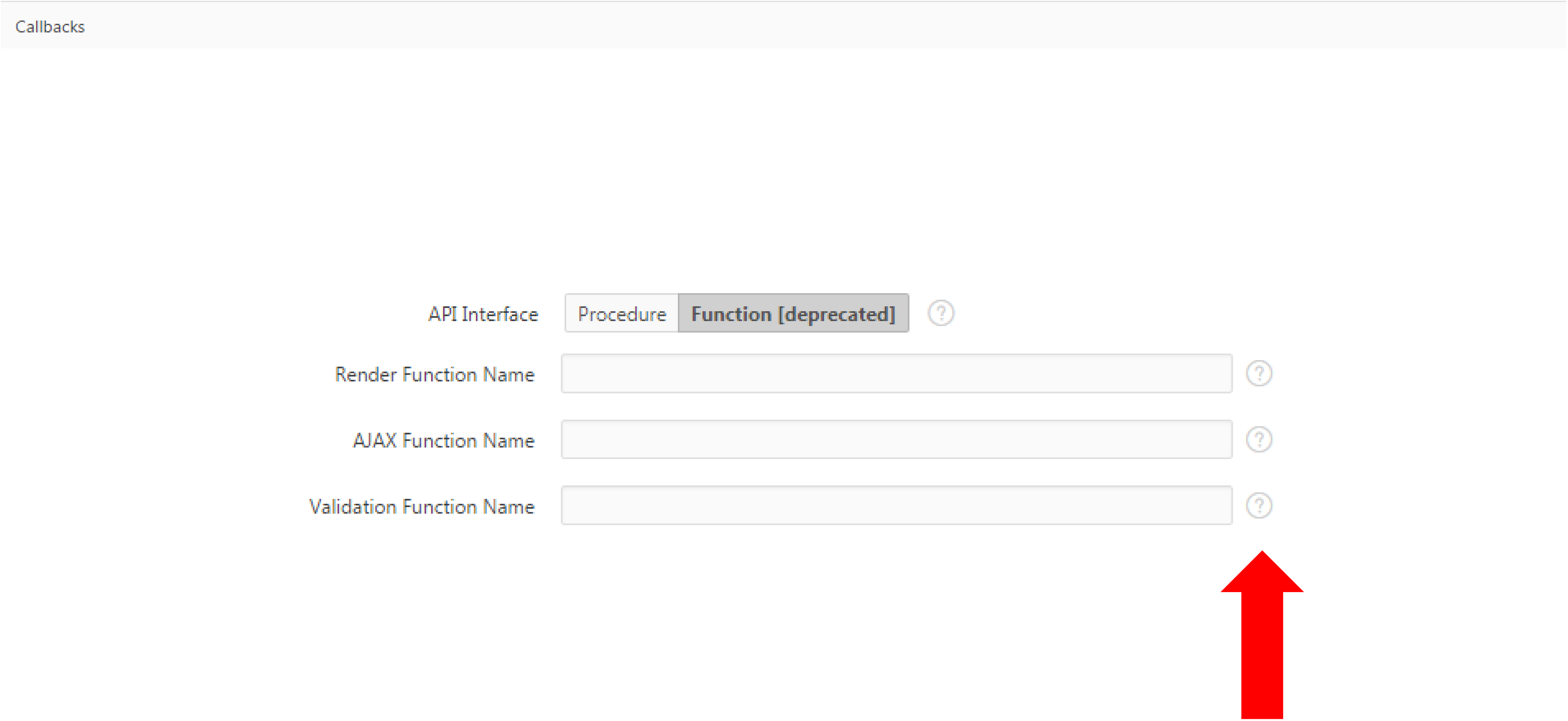
Grid

Start New Row: [✓ Yes] No

Column: Automatic

Column Span: Automatic

Label Column Span: Page Template Default



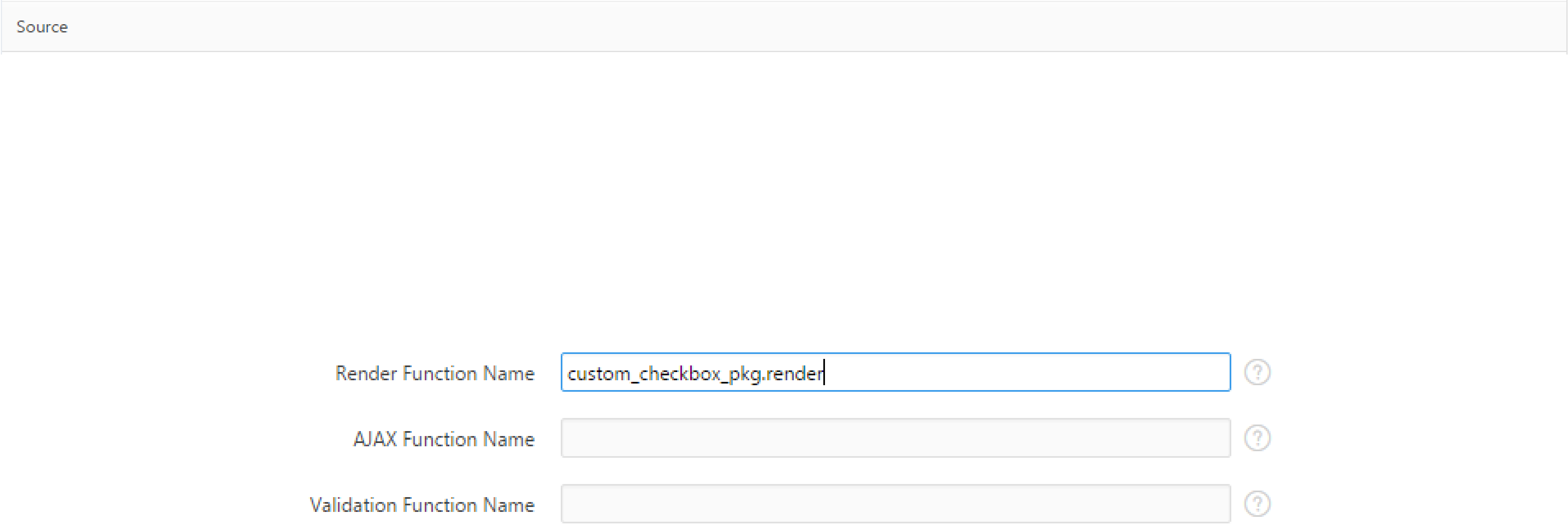
Callbacks

API Interface: Procedure [✓ Function [deprecated]]

Render Function Name:

AJAX Function Name:

Validation Function Name:

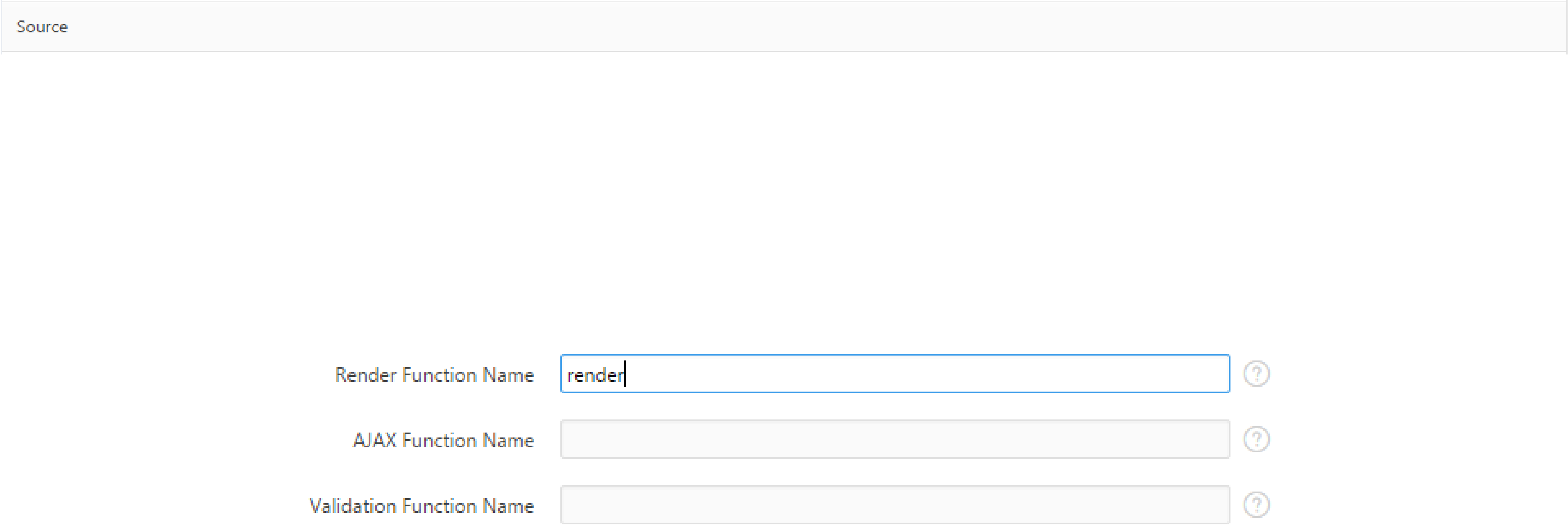


Source

Render Function Name: custom_checkbox_pkg.render

AJAX Function Name

Validation Function Name

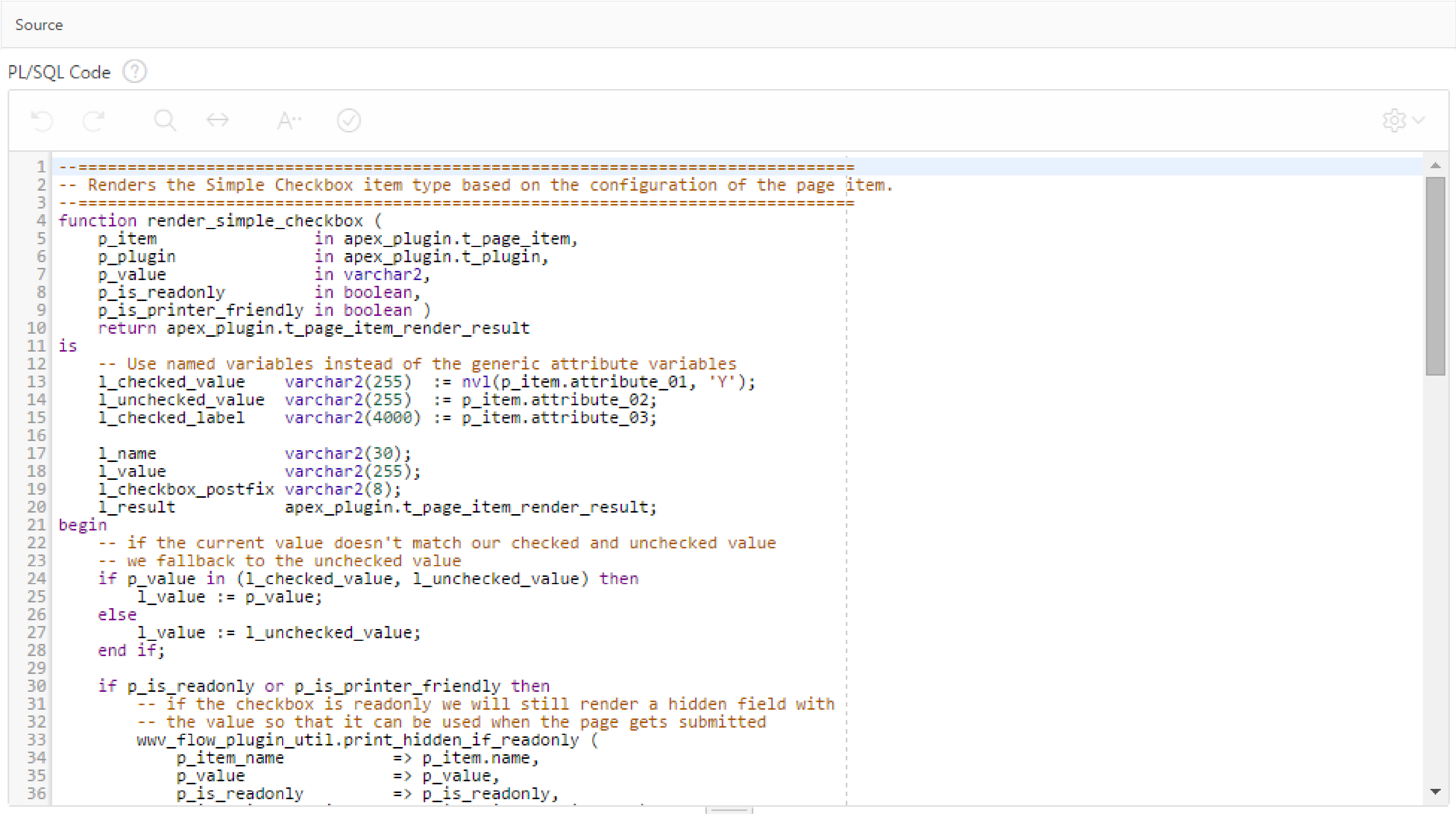


Source

Render Function Name: render

AJAX Function Name:

Validation Function Name:



```
--==============================================================================
-- Renders the Simple Checkbox item type based on the configuration of the page item.
--==============================================================================
function render_simple_checkbox (
    p_item                in apex_plugin.t_page_item,
    p_plugin              in apex_plugin.t_plugin,
    p_value               in varchar2,
    p_is_readonly         in boolean,
    p_is_printer_friendly in boolean )
    return apex_plugin.t_page_item_render_result
is
    -- Use named variables instead of the generic attribute variables
    l_checked_value    varchar2(255)  := nvl(p_item.attribute_01, 'Y');
    l_unchecked_value  varchar2(255)  := p_item.attribute_02;
    l_checked_label    varchar2(4000) := p_item.attribute_03;

    l_name             varchar2(30);
    l_value            varchar2(255);
    l_checkbox_postfix varchar2(8);
    l_result           apex_plugin.t_page_item_render_result;
begin
    -- if the current value doesn't match our checked and unchecked value
    -- we fallback to the unchecked value
    if p_value in (l_checked_value, l_unchecked_value) then
        l_value := p_value;
    else
        l_value := l_unchecked_value;
    end if;

    if p_is_readonly or p_is_printer_friendly then
        -- if the checkbox is readonly we will still render a hidden field with
        -- the value so that it can be used when the page gets submitted
        wwv_flow_plugin_util.print_hidden_if_readonly (
            p_item_name           => p_item.name,
            p_value               => p_value,
            p_is_readonly         => p_is_readonly,
procedure <name of function> (
    p_item   in            apex_plugin.t_item,
    p_plugin in            apex_plugin.t_plugin,
    p_param  in            apex_plugin.t_item_render_param,
    p_result in out nocopy apex_plugin.t_page_item_render_result
)
```

```
procedure <name of function> (
    p_item   in            apex_plugin.t_item,
    p_plugin in            apex_plugin.t_plugin,
    p_param  in            apex_plugin.t_item_render_param,
    p_result in out nocopy apex_plugin.t_page_item_render_result
)
```

PL/SQL

```
procedure <name of function> (
    p_item   in                apex_plugin.t_item,
    p_plugin in                apex_plugin.t_plugin,
    p_param  in                apex_plugin.t_item_render_param,
    p_result in out nocopy apex_plugin.t_page_item_render_result
)
```

```
procedure <name of function> (
    p_item   in                apex_plugin.t_item,
    p_plugin in                apex_plugin.t_plugin,
    p_param  in                apex_plugin.t_item_render_param,
    p_result in out nocopy apex_plugin.t_page_item_render_result
)
```

```
procedure <name of function> (
    p_item   in             apex_plugin.t_item,
    p_plugin in             apex_plugin.t_plugin,
    p_param  in             apex_plugin.t_item_render_param,
    p_result in out nocopy apex_plugin.t_page_item_render_result
)
```

PL/SQL

```
procedure render (
    p_item   in                apex_plugin.t_item,
    p_plugin in                apex_plugin.t_plugin,
    p_param  in                apex_plugin.t_item_render_param,
    p_result in out nocopy apex_plugin.t_page_item_render_result
) is




begin
    ...
end render;
```

PL/SQL

```
procedure render (
    p_item   in                 apex_plugin.t_item,
    p_plugin in                 apex_plugin.t_plugin,
    p_param  in                 apex_plugin.t_item_render_param,
    p_result in out nocopy apex_plugin.t_page_item_render_result
) is

   --Use named variables
   --instead of the generic attribute variables
   l_checked_value   varchar2(255) := nvl(p_item.attribute_01
                                         ,'On');
   l_unchecked_value varchar2(255) := p_item.attribute_02;
   l_checked_label   varchar2(4000) := p_item.attribute_03;


begin
   ...
end render;
```

```
procedure render (
    p_item   in                apex_plugin.t_item,
    p_plugin in                apex_plugin.t_plugin,
    p_param  in                apex_plugin.t_item_render_param,
    p_result in out nocopy apex_plugin.t_page_item_render_result
) is

   --Use named variables
   --instead of the generic attribute variables
   l_checked_value   varchar2(255) := nvl(p_item.attribute_01
                                         ,'On');
   l_unchecked_value varchar2(255) := p_item.attribute_02;
   l_checked_label   varchar2(4000) := p_item.attribute_03;


begin
   ...
end render;
```

21 APEX_PLUGIN

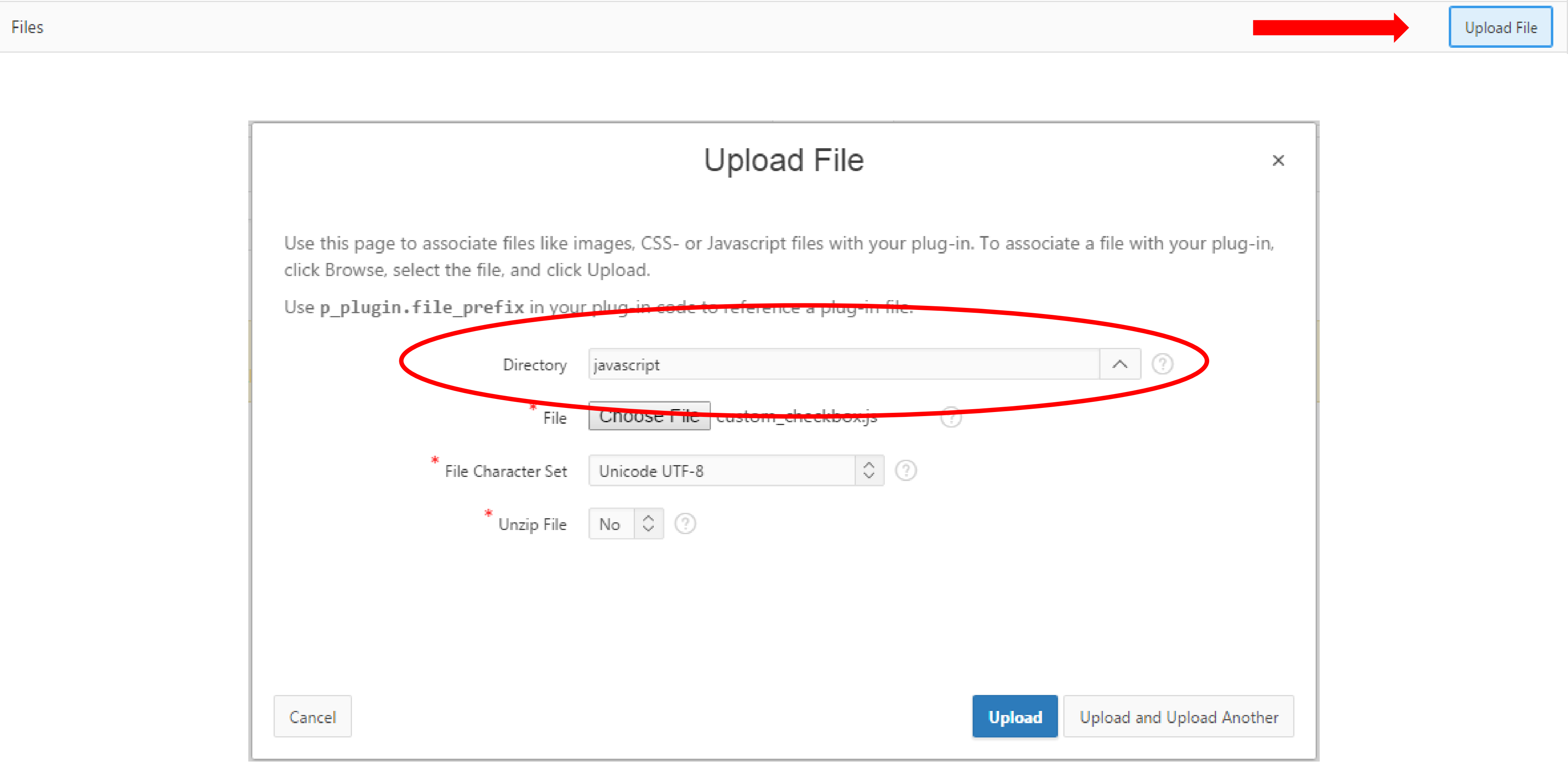
The `APEX_PLUGIN` package provides the interface declarations and some utility functions to work with plug-ins.

JS
JavaScript

HTML
5

JavaScript

`custom_checkbox.js`



Upload File

Use this page to associate files like images, CSS- or Javascript files with your plug-in. To associate a file with your plug-in, click Browse, select the file, and click Upload.

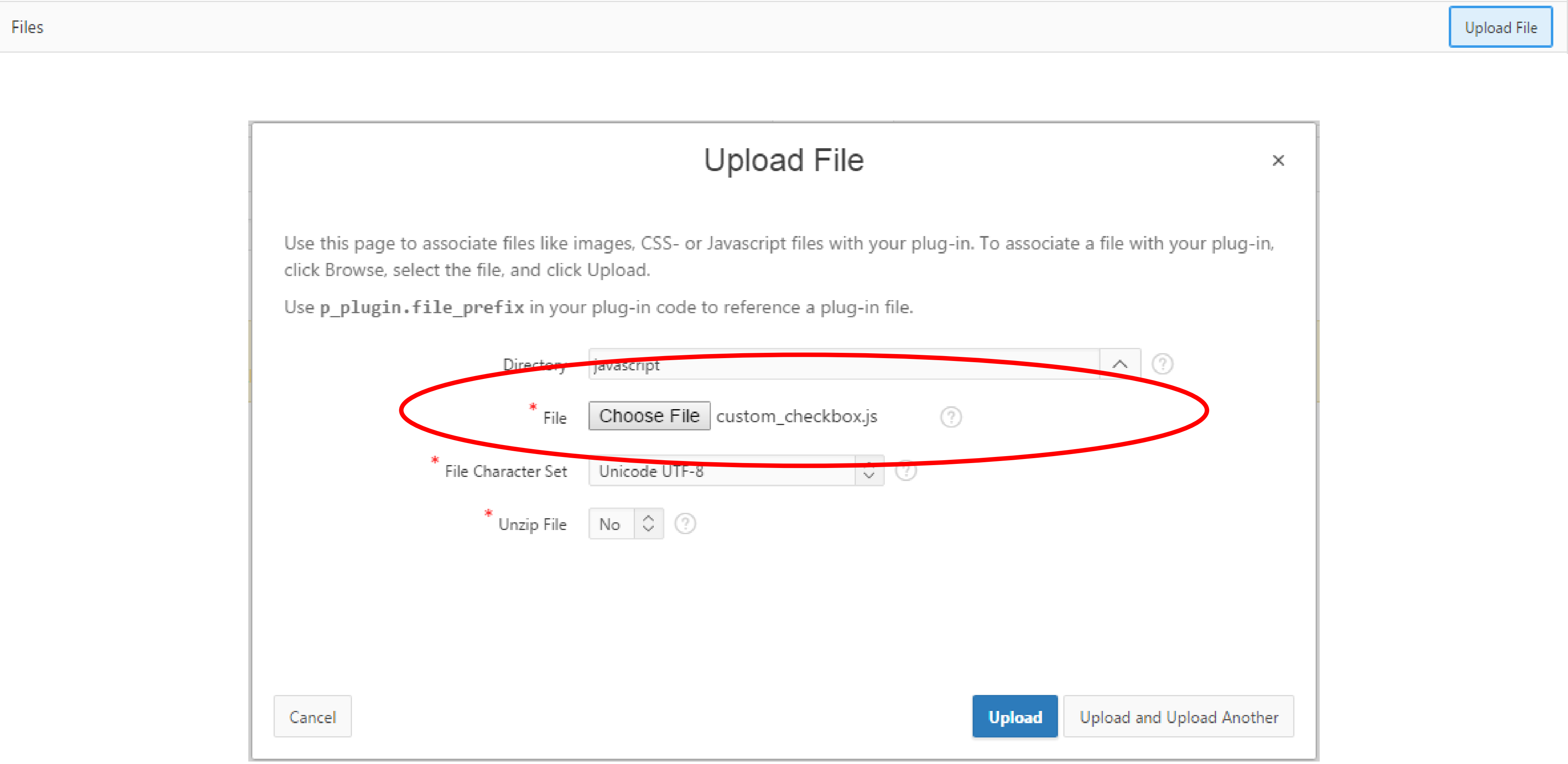
Use `p_plugin.file_prefix` in your plug-in code to reference a plug-in file.

Directory: javascript

* File: Choose File custom_checkbox.js

* File Character Set: Unicode UTF-8

* Unzip File: No

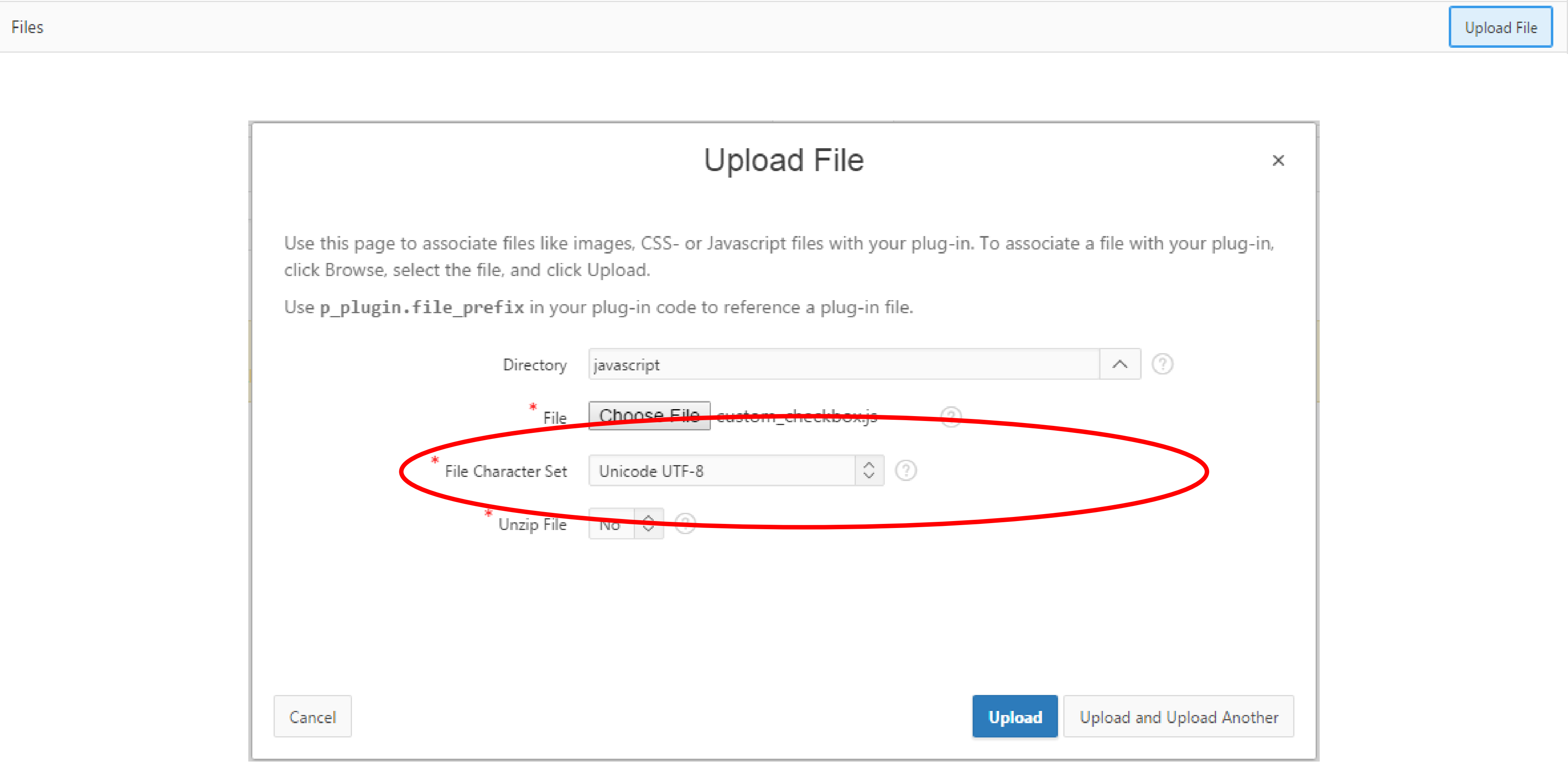


Upload File

Use this page to associate files like images, CSS- or Javascript files with your plug-in. To associate a file with your plug-in, click Browse, select the file, and click Upload.

Use `p_plugin.file_prefix` in your plug-in code to reference a plug-in file.

Directory	javascript
* File	Choose File custom_checkbox.js
* File Character Set	Unicode UTF-8
* Unzip File	No



Upload File

Use this page to associate files like images, CSS- or Javascript files with your plug-in. To associate a file with your plug-in, click Browse, select the file, and click Upload.

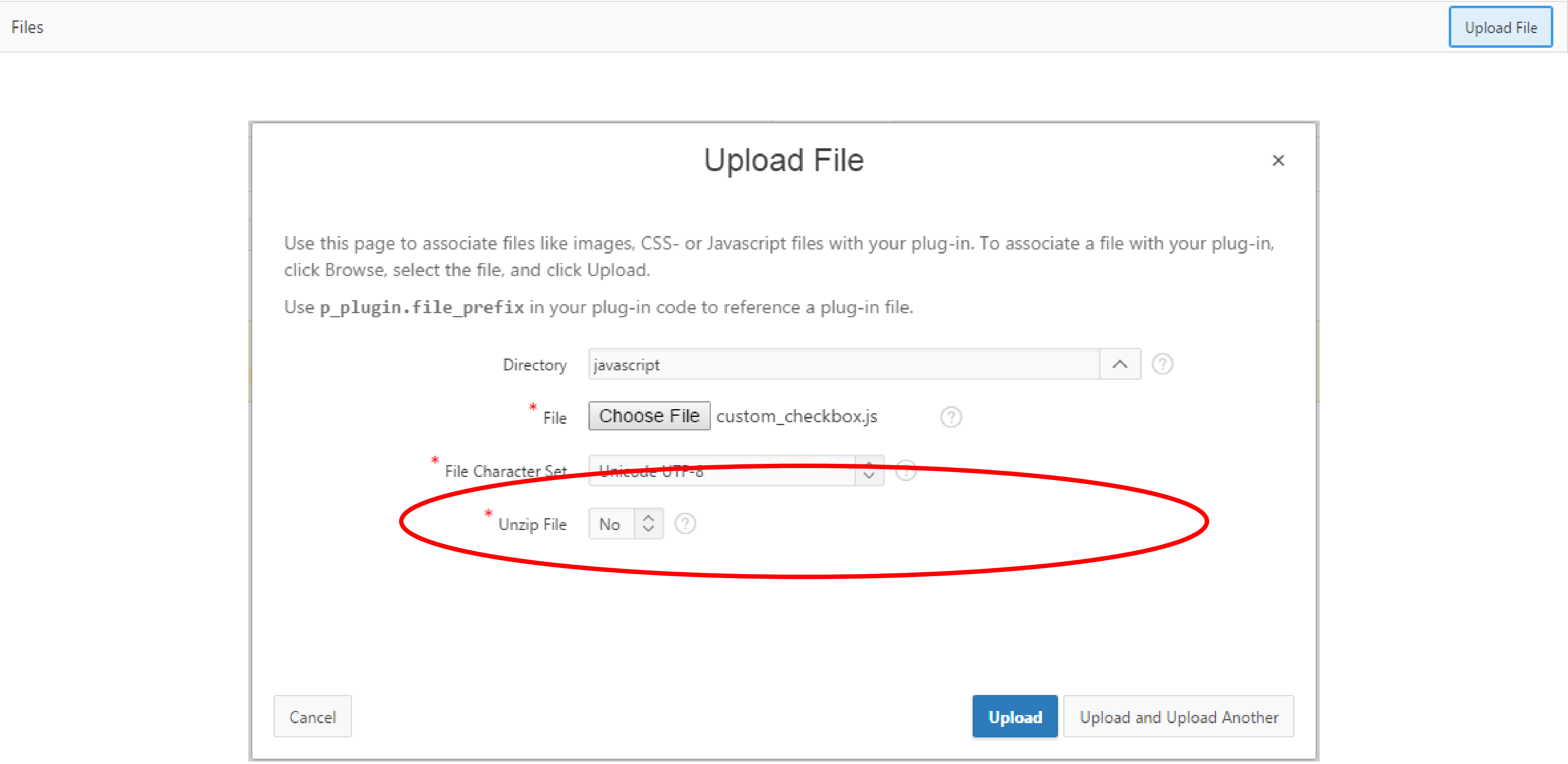
Use `p_plugin.file_prefix` in your plug-in code to reference a plug-in file.

Directory: javascript

* File: Choose File custom_checkbox.js

* File Character Set: Unicode UTF-8

* Unzip File: No



Upload File

Use this page to associate files like images, CSS- or Javascript files with your plug-in. To associate a file with your plug-in, click Browse, select the file, and click Upload.

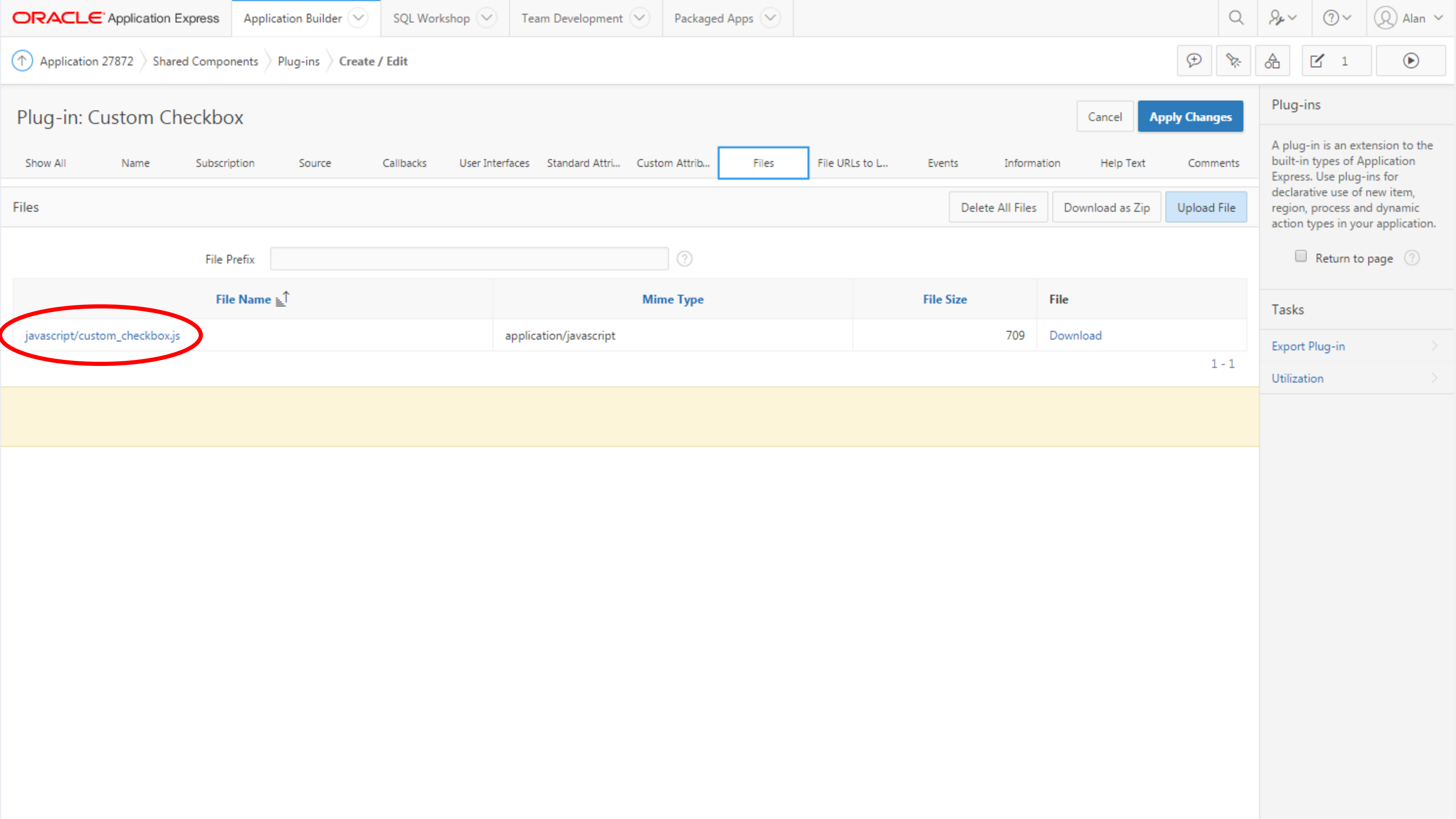
Use `p_plugin.file_prefix` in your plug-in code to reference a plug-in file.

Directory: javascript

* File: custom_checkbox.js

* File Character Set: Unicode UTF-8

* Unzip File: No



Plug-in: Custom Checkbox

Files

File Prefix

File Name	Mime Type	File Size	File
javascript/custom_checkbox.js	application/javascript	709	Download

1 - 1

Plug-ins

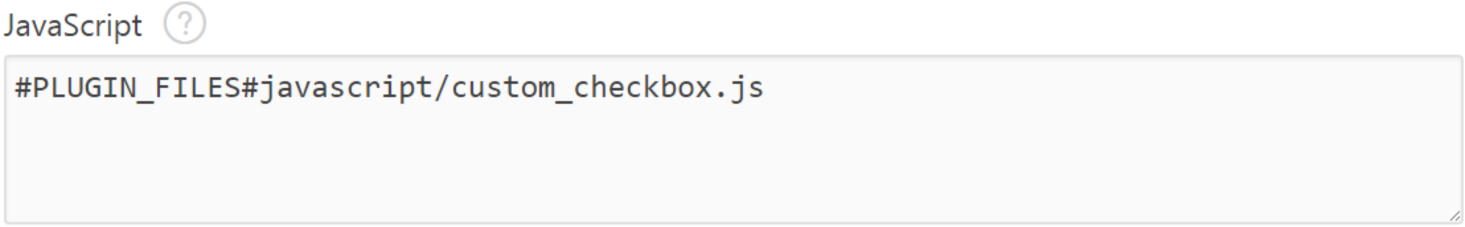
A plug-in is an extension to the built-in types of Application Express. Use plug-ins for declarative use of new item, region, process and dynamic action types in your application.

□ Return to page

```
begin

   ...

   -- Add the JavaScript library
   apex_javascript.add_library(
               p_name      => 'custom_checkbox'
              ,p_directory => p_plugin.file_prefix ||
                                 'javascript/'
              ,p_version   => null
   );

   ...


end render;
```

File URLs to Load

```
-- Add the JavaScript library
apex_javascript.add_library(
            p_name      => 'custom_checkbox'
           ,p_directory => p_plugin.file_prefix ||
                              'javascript/'
           ,p_version   => null
);
```



```
#PLUGIN_FILES#javascript/custom_checkbox.js
```

```
function custom_checkbox (pPageItemName, pOptions){...};



custom_checkbox ('P1_PLUGIN_CHECK'
                ,{'unchecked':'Off','checked':'On'});
```

JavaScript

```
pPageItemName
```

PL/SQL

```
apex_javascript.add_value(p_item.name)
```

JavaScript

```
'P1_PLUGIN_CHECK',
```

JavaScript

```
pOptions
```

PL/SQL

```
apex_javascript.add_attribute('unchecked'
                             ,l_unchecked_value
                             ,false)
```

JavaScript

```
'unchecked':'Off',
begin
   ...

   -- Add the call to initialize the widget
   l_code := 'custom_checkbox(' ||
         apex_javascript.add_value(p_item.name) || '{' ||
         apex_javascript.add_attribute('unchecked'
                                      ,l_unchecked_value
                                      ,false) ||
         apex_javascript.add_attribute('checked'
                                      ,l_checked_value
                                      ,false
                                      ,false) || '});';
   apex_javascript.add_onload_code(p_code => l_code);

   ...
end render;
```

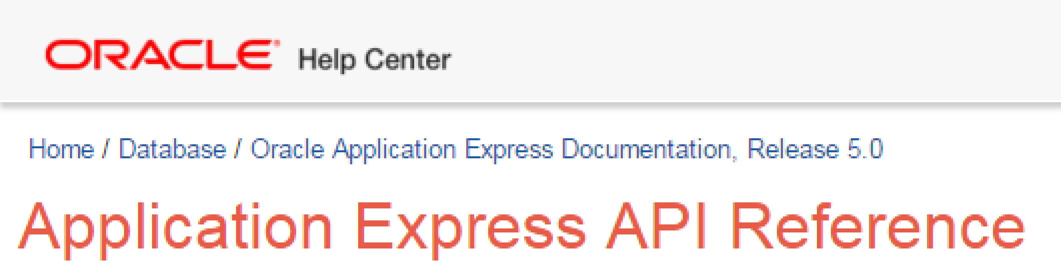
```
begin
   ...

   -- Add the call to initialize the widget
   l_code := 'custom_checkbox(' ||
         apex_javascript.add_value(p_item.name) || '{' ||
         apex_javascript.add_attribute('unchecked'
                                      ,l_unchecked_value
                                      ,false) ||
         apex_javascript.add_attribute('checked'
                                      ,l_checked_value
                                      ,false
                                      ,false) || '});';
   apex_javascript.add_onload_code(p_code => l_code);

   ...
end render;
```

```
begin
   ...

   -- Add the call to initialize the widget
   l_code := 'custom_checkbox(' ||
         apex_javascript.add_value(p_item.name) || '{' ||
         apex_javascript.add_attribute('unchecked'
                                      ,l_unchecked_value
                                      ,false) ||
         apex_javascript.add_attribute('checked'
                                      ,l_checked_value
                                      ,false
                                      ,false) || '});';
   apex_javascript.add_onload_code(p_code => l_code);

   ...
end render;
```

```
begin
   ...

   -- Add the call to initialize the widget
   l_code := 'custom_checkbox(' ||
         apex_javascript.add_value(p_item.name) || '{' ||
         apex_javascript.add_attribute('unchecked'
                                      ,l_unchecked_value
                                      ,false) ||
         apex_javascript.add_attribute('checked'
                                      ,l_checked_value
                                      ,false
                                      ,false) || '});';
   apex_javascript.add_onload_code(p_code => l_code);

   ...
end render;
```

```
begin
   ...

   -- Add the call to initialize the widget
   l_code := 'custom_checkbox(' ||
         apex_javascript.add_value(p_item.name) || '{' ||
         apex_javascript.add_attribute('unchecked'
                                      ,l_unchecked_value
                                      ,false) ||
         apex_javascript.add_attribute('checked'
                                      ,l_checked_value
                                      ,false
                                      ,false) || '});';
   apex_javascript.add_onload_code(p_code => l_code);

   ...
end render;
```

```
begin
   ...

   -- Add the call to initialize the widget
   l_code := 'custom_checkbox(' ||
         apex_javascript.add_value(p_item.name) || '{' ||
         apex_javascript.add_attribute('unchecked'
                                      ,l_unchecked_value
                                      ,false) ||
         apex_javascript.add_attribute('checked'
                                      ,l_checked_value
                                      ,false
                                      ,false) || '});';
   apex_javascript.add_onload_code(p_code => l_code);

   ...
end render;
```



Home / Database / Oracle Application Express Documentation, Release 5.0

Application Express API Reference

14 APEX_JAVASCRIPT

The `APEX_JAVASCRIPT` package provides utility functions for adding dynamic JavaScript code to HTTP output. This package is usually used for plug-in development.

```
<input type="hidden"
       id="P1_PLUGIN_CHECK_HIDDEN"
       name="p_t01"
       value="Off" />


<input type="checkbox"
       id="P1_PLUGIN_CHECK"
       class="custom_checkbox"
       value="" />


<label for="P1_PLUGIN_CHECK">Does this work?</label>
```

HTML

```
<input type="hidden"
       id="P1_PLUGIN_CHECK_HIDDEN"
       name="p_t01"
       value="Off" />
```

PL/SQL

```
sys.htp.prn()
```

```
begin
   ...

   -- Get the name of the page item
   l_name := apex_plugin.get_input_name_for_page_item(false);

   -- Render the hidden field
   -- which actually stores the checkbox value
   sys.htp.prn('<input type="hidden" ' ||
                'id="' || p_item.name || '_HIDDEN" ' ||
                'name="' || l_name || '" ' ||
                'value="' || p_value || '" />');


   ...
end render;
```

```
begin
   ...

   -- Get the name of the page item
   l_name := apex_plugin.get_input_name_for_page_item(false);

   -- Render the hidden field
   -- which actually stores the checkbox value
   sys.htp.prn('<input type="hidden" ' ||
                'id="' || p_item.name || '_HIDDEN" ' ||
                'name="' || l_name || '" ' ||
                'value="' || p_value || '" />');


   ...
end render;
```

```
begin
   ...

   -- Get the name of the page item
   l_name := apex_plugin.get_input_name_for_page_item(false);

   -- Render the hidden field
   -- which actually stores the checkbox value
   sys.htp.prn('<input type="hidden" ' ||
                'id="' || p_item.name || '_HIDDEN" ' ||
                'name="' || l_name || '" ' ||
                'value="' || p_value || '" />');


   ...
end render;
```

```
begin
   ...

   -- Get the name of the page item
   l_name := apex_plugin.get_input_name_for_page_item(false);

   -- Render the hidden field
   -- which actually stores the checkbox value
   sys.htp.prn('<input type="hidden" ' ||
                'id="' || p_item.name || '_HIDDEN" ' ||
                'name="' || l_name || '" ' ||
                'value="' || p_value || '" />');


   ...
end render;
```

```
begin
   ...

   -- Get the name of the page item
   l_name := apex_plugin.get_input_name_for_page_item(false);

   -- Render the hidden field
   -- which actually stores the checkbox value
   sys.htp.prn('<input type="hidden" ' ||
                'id="' || p_item.name || '_HIDDEN" ' ||
                'name="' || l_name || '" ' ||
                'value="' || p_value || '" />');


   ...
end render;
```

HTML

```
<input type="checkbox"
       id="P1_PLUGIN_CHECK"
       class="custom_checkbox"
       value="" />
```

PL/SQL

```
sys.htp.prn()
```

```
begin
   ...

   -- render the checkbox widget
   sys.htp.prn('<input type="checkbox" ' ||
               'id="' || p_item.name || '" ' ||
               'value="' || p_value || '" ' ||
               case
                  when p_value = l_checked_value
                  then 'checked="checked" '
               end ||
               coalesce(p_item.element_attributes
                       ,'class="simple_checkbox"') || ' />');
   ...
end render;
```

```
begin
   ...

   -- render the checkbox widget
   sys.htp.prn('<input type="checkbox" ' ||
               'id="' || p_item.name || '" ' ||
               'value="' || p_value || '" ' ||
               case
                  when p_value = l_checked_value
                  then 'checked="checked" '
               end ||
               coalesce(p_item.element_attributes
                       ,'class="simple_checkbox"') || ' />');
   ...
end render;
```

```
begin
   ...

   -- render the checkbox widget
   sys.htp.prn('<input type="checkbox" ' ||
               'id="' || p_item.name || '" ' ||
               'value="' || p_value || '" ' ||
               case
                  when p_value = l_checked_value
                  then 'checked="checked" '
               end ||
               coalesce(p_item.element_attributes
                       ,'class="simple_checkbox"') || ' />');
   ...
end render;
```

```
begin
   ...

   -- render the checkbox widget
   sys.htp.prn('<input type="checkbox" ' ||
               'id="' || p_item.name || '" ' ||
               'value="' || p_value || '" ' ||
               case
                  when p_value = l_checked_value
                  then 'checked="checked" '
               end ||
               coalesce(p_item.element_attributes
                       ,'class="simple_checkbox"') || ' />');
   ...
end render;
```

```
begin
   ...

   -- render the checkbox widget
   sys.htp.prn('<input type="checkbox" ' ||
               'id="' || p_item.name || '" ' ||
               'value="' || p_value || '" ' ||
               case
                  when p_value = l_checked_value
                  then 'checked="checked" '
               end ||
               coalesce(p_item.element_attributes
                       ,'class="simple_checkbox"') || ' />');
   ...
end render;
```

HTML

```
<label for="P1_PLUGIN_CHECK">Does this work?</label>
```

PL/SQL

```
sys.htp.prn()
```

```
begin
   ...

   -- print label after checkbox
   if l_checked_label is not null
   then
      sys.htp.prn('<label for="' || p_item.name || '">' ||
                  l_checked_label || '</label>');
   end if;

   ...
end render;
```

```
begin
   ...

   -- print label after checkbox
   if l_checked_label is not null
   then
      sys.htp.prn('<label for="' || p_item.name || '">' ||
                  l_checked_label || '</label>');
   end if;

   ...
end render;
```

```
begin
   ...

   -- print label after checkbox
   if l_checked_label is not null
   then
      sys.htp.prn('<label for="' || p_item.name || '">' ||
                  l_checked_label || '</label>');
   end if;

   ...
end render;
```

DEMO

☑ Plug-in Repository

☑ Create from scratch

🔌 Copy of existing Plug-in

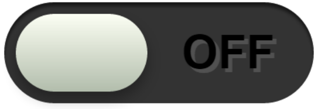
Copy of existing Plug-in

☑

on

□

off



on

off

```
.fancy_checkbox {
    width: 80px;
    height: 26px;
    background: #333;
    margin: 20px auto;
    position: relative;
    border-radius: 50px;
    box-shadow: inset 0px 1px 1px rgba(0, 0, 0, 0.5), 0px 1px 0px rgba(255, 255, 255, 0.2);
}
```

```css
.fancy_checkbox:after {
    content: 'OFF';
    color: #000;
    position: absolute;
    right: 10px;
    z-index: 0;
    font: 12px/26px Arial, sans-serif;
    font-weight: bold;
    text-shadow: 1px 1px 0px rgba(255, 255, 255, 0.15);
}
```

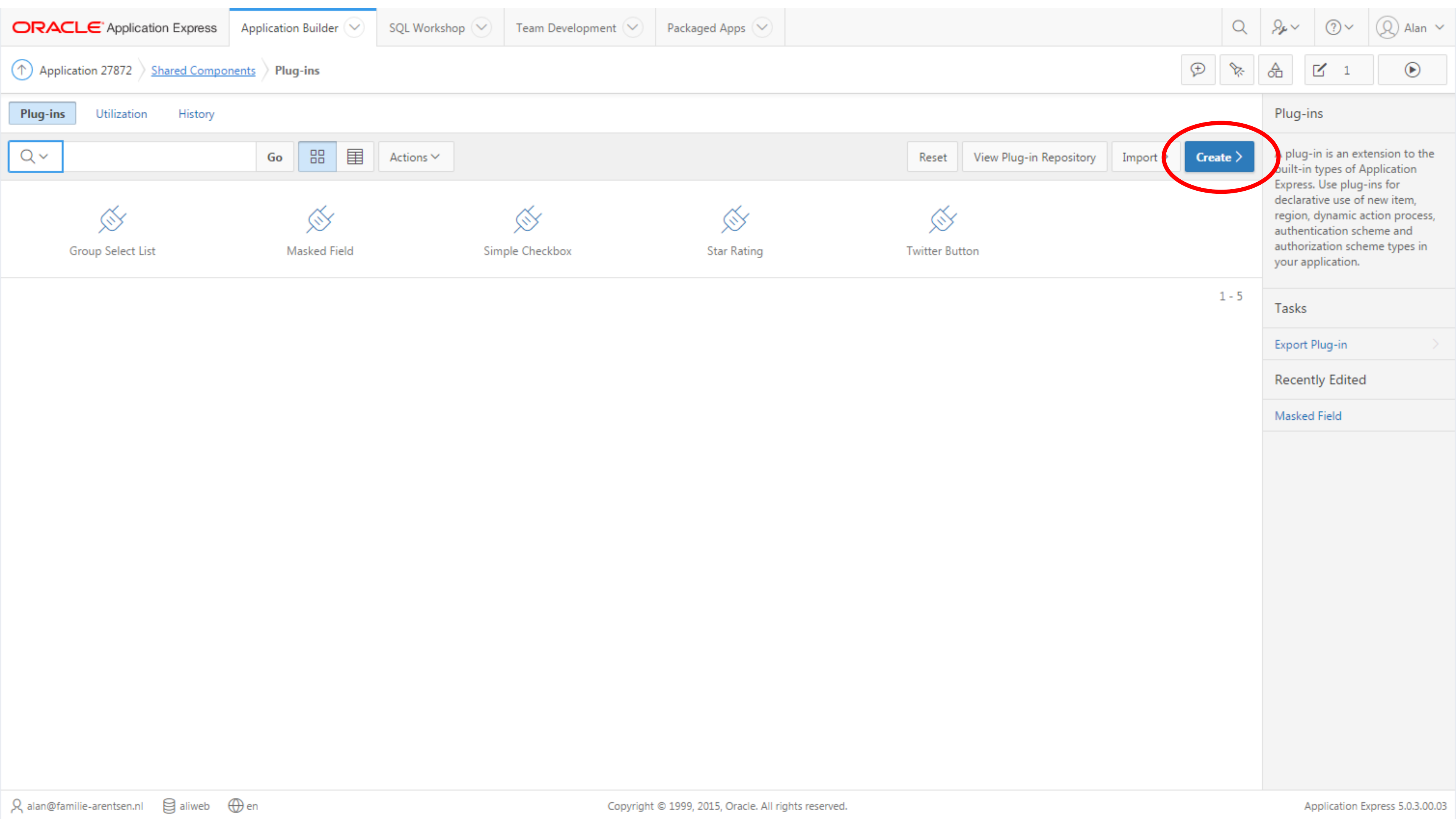
```css
.fancy_checkbox:before {
   content: 'ON';
   color: #27ae60;
   position: absolute;
   left: 10px;
   z-index: 0;
   font: 12px/26px Arial, sans-serif;
   font-weight: bold;
}
```

```
.fancy_checkbox label {
    display: block;
    width: 34px;
    height: 20px;
    cursor: pointer;
    position: absolute;
    top: 3px;
    left: 3px;
    z-index: 1;
    background: #fcfff4;
    background: -webkit-linear-gradient(top, #fcfff4 0%, #dfe5d7 40%, #b3bead 100%);
    background: linear-gradient(top, #fcfff4 0%, #dfe5d7 40%, #b3bead 100%);
    border-radius: 50px;
    -webkit-transition: all 0.4s ease;
    transition: all 0.4s ease;
    box-shadow: 0px 2px 5px 0px rgba(0, 0, 0, 0.3);
}
```

```
.fancy_checkbox input[type=checkbox] {
   visibility: hidden;
}


.fancy_checkbox input[type=checkbox]:checked + label {
   left: 43px;
}
```

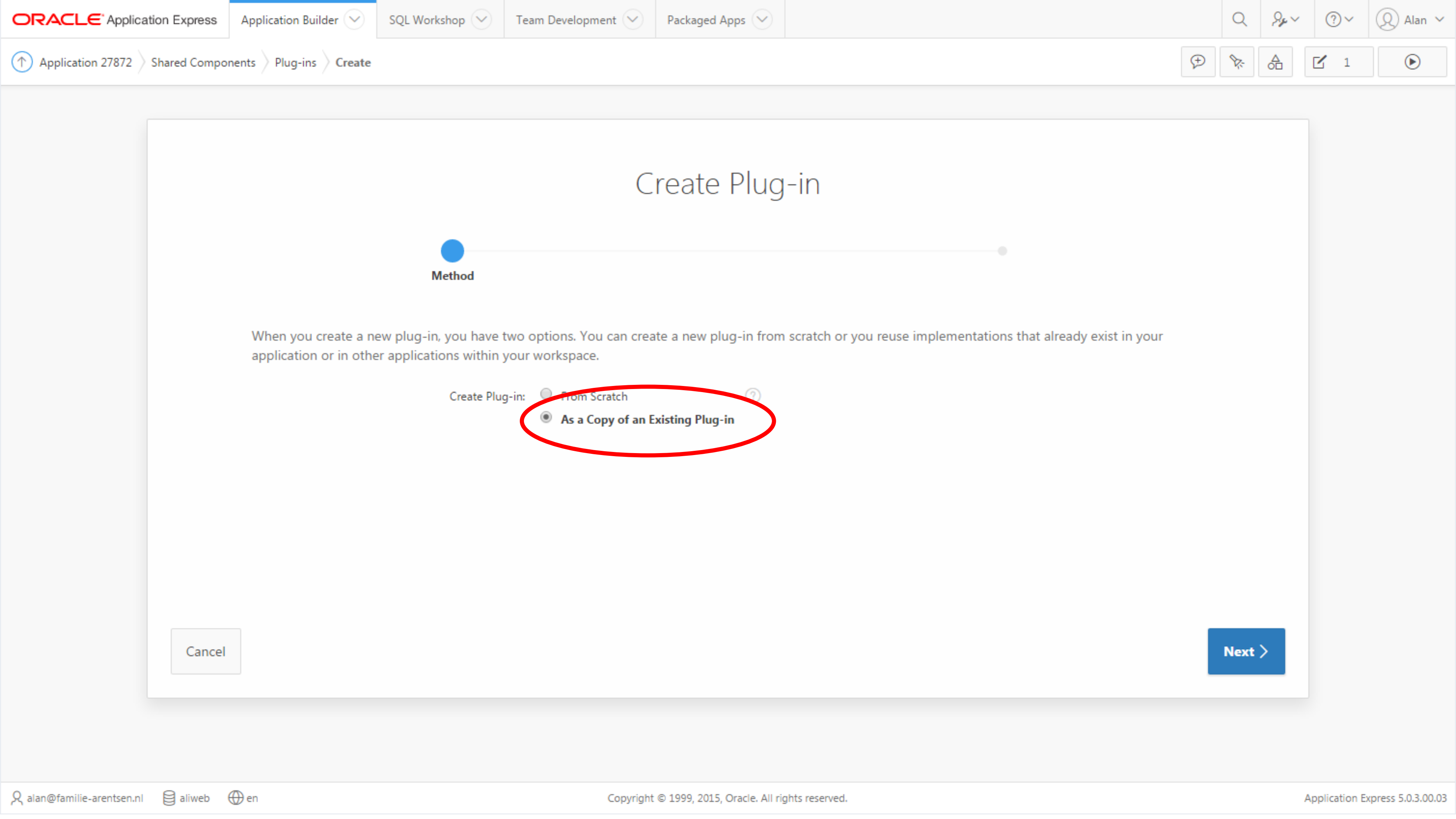
```
<input type="checkbox"
       id="P1_PLUGIN_CHECK"
       class="custom_checkbox"
       value="" />

<label for="P1_PLUGIN_CHECK">Does this work?</label>
```

```
<div class="fancy_checkbox">
   <input type="checkbox"
          id="P1_PLUGIN_CHECK"
          class="custom_checkbox"
          value="" />

   <label for="P1_PLUGIN_CHECK">Does this work?</label>
</div>
```



A plug-in is an extension to the built-in types of Application Express. Use plug-ins for declarative use of new item, region, dynamic action process, authentication scheme and authorization scheme types in your application.

Copyright © 1999, 2015, Oracle. All rights reserved.



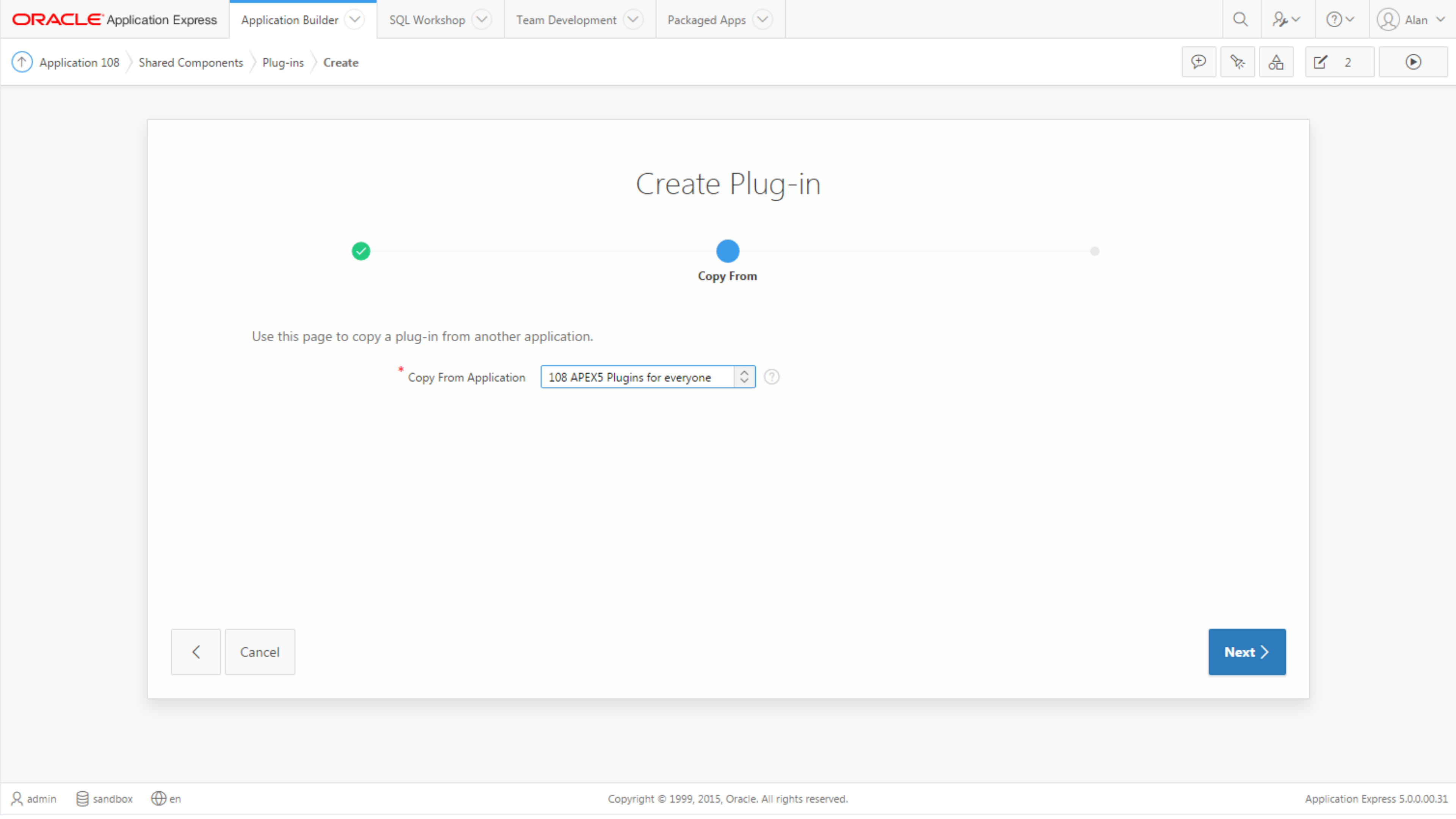
Create Plug-in

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

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

Copyright © 1999, 2015, Oracle. All rights reserved.

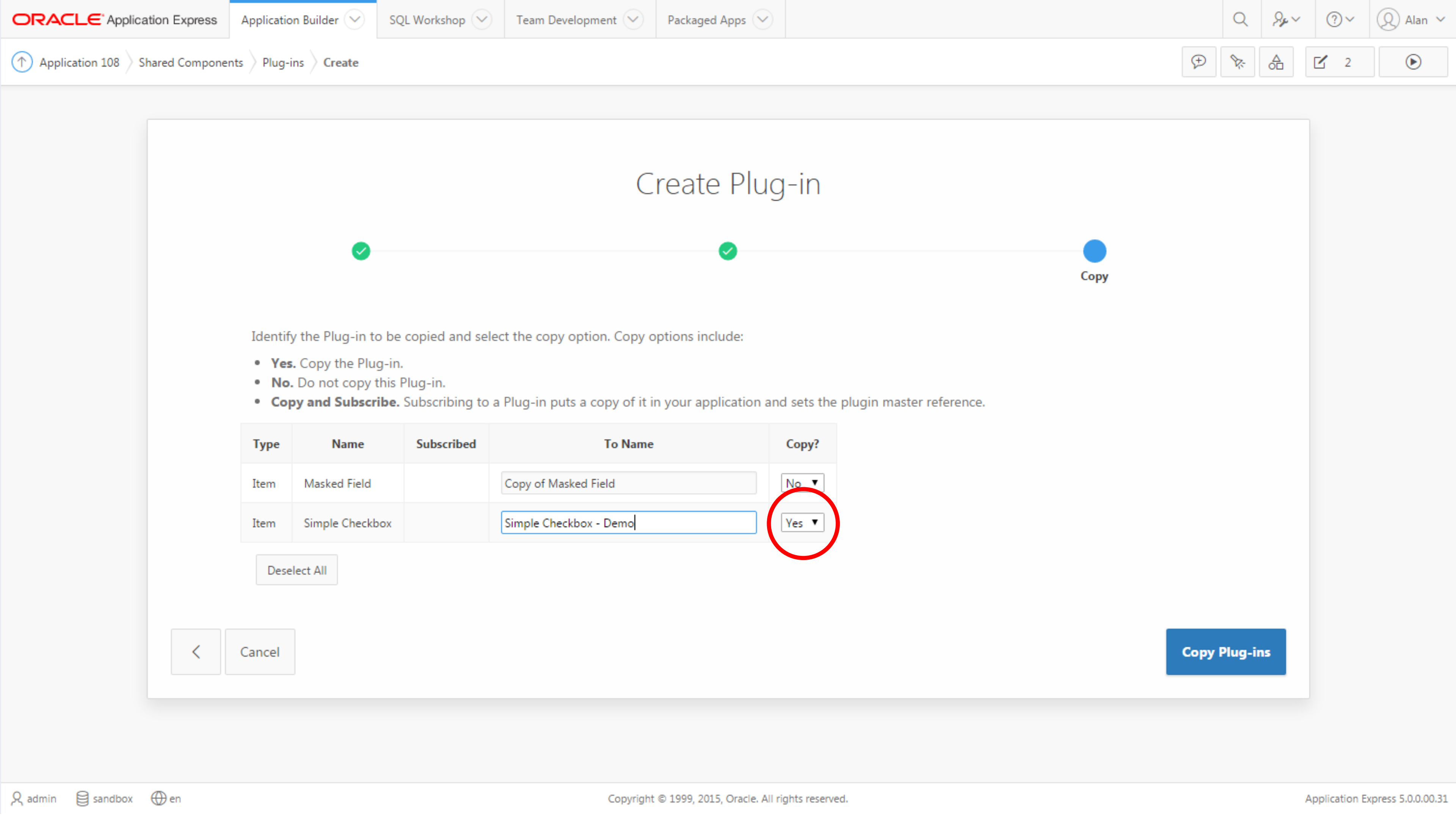
Application Express 5.0.3.00.03



Create Plug-in

Use this page to copy a plug-in from another application.

* Copy From Application: 108 APEX5 Plugins for everyone



Create Plug-in

Identify the Plug-in to be copied and select the copy option. Copy options include:

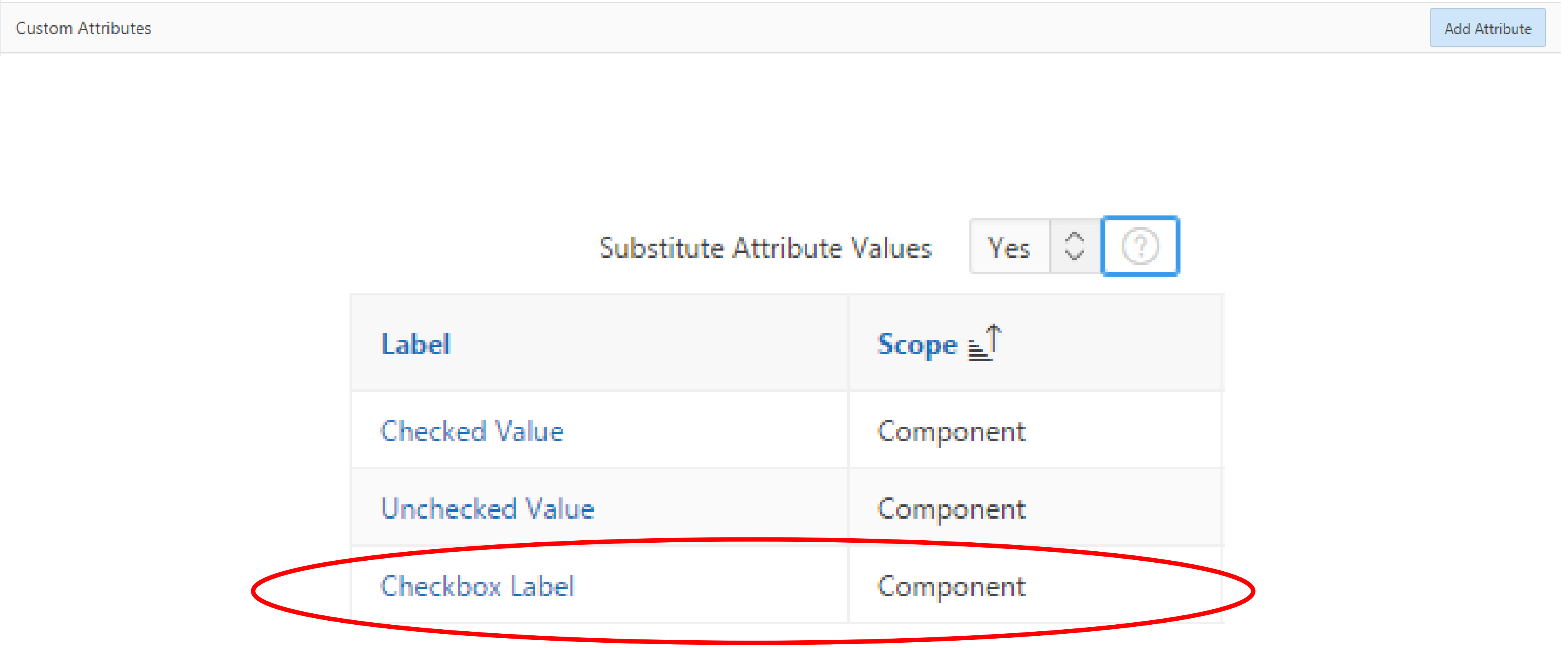
- **Yes.** Copy the Plug-in.
- **No.** Do not copy this Plug-in.
- **Copy and Subscribe.** Subscribing to a Plug-in puts a copy of it in your application and sets the plugin master reference.

Type	Name	Subscribed	To Name	Copy?
Item	Masked Field		Copy of Masked Field	No ▼
Item	Simple Checkbox		Simple Checkbox - Demo	Yes ▼

Application Express 5.0.0.00.31

```
begin
   ...

   -- render the checkbox widget
   sys.htp.prn('<div class="fancy_checkbox"> ' ||
               '<input type="checkbox" ' ||
               'id="' || p_item.name || '" ' ||
               'value="' || p_value || '" ' ||
               case
                  when p_value = l_checked_value
                  then 'checked="checked" '
               end ||
               coalesce(p_item.element_attributes
                       ,'class="simple_checkbox"') || ' />');
   ...
end render;
```

```
begin
   ...

   -- print label after checkbox
   sys.htp.prn('<label for="' || p_item.name || '">' ||
                  l_checked_label || '</label>');
   sys.htp.prn('</div>');

   ...
end render;
```

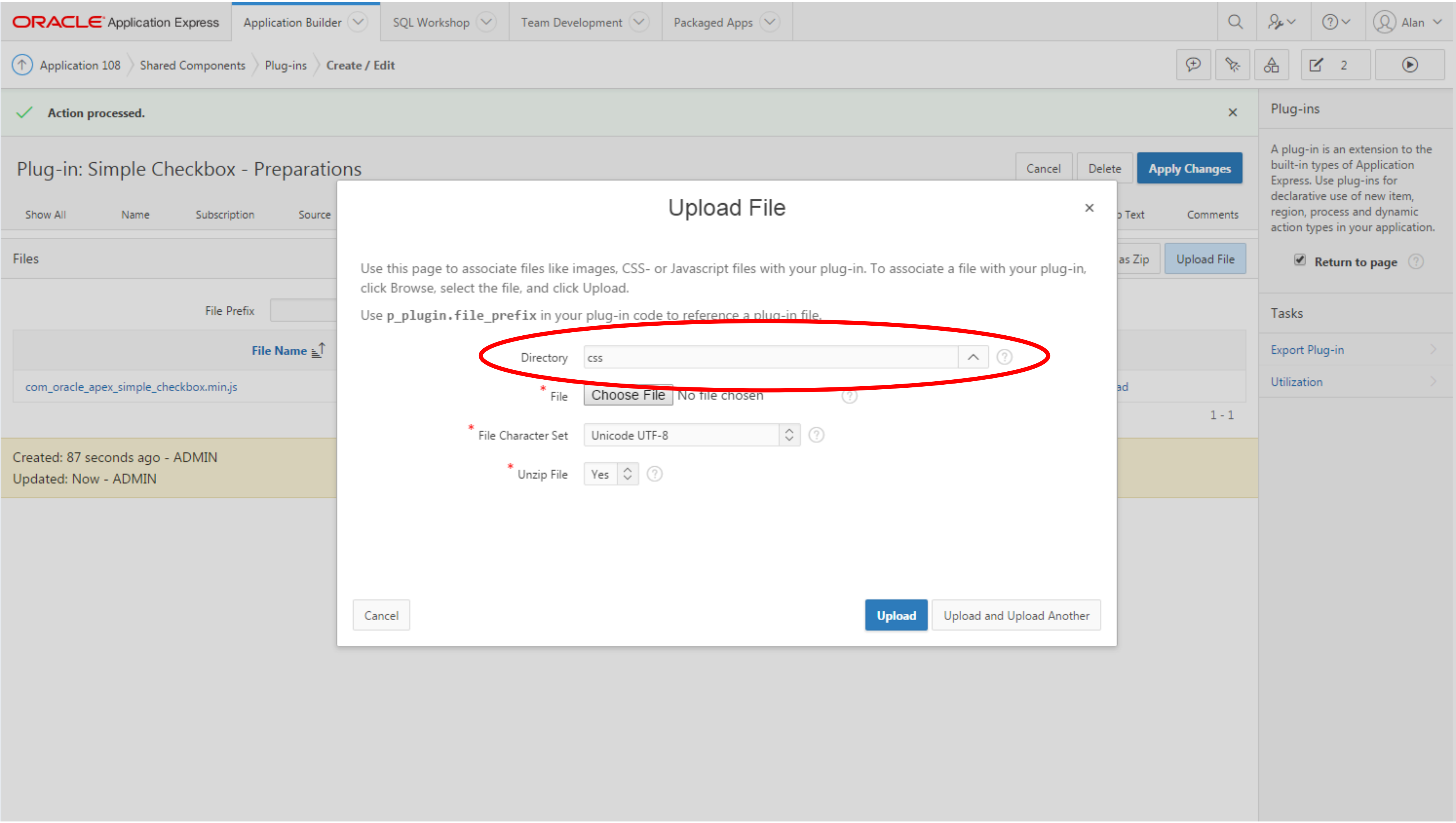


Custom Attributes

Substitute Attribute Values: Yes

Label	Scope
Checked Value	Component
Unchecked Value	Component
Checkbox Label	Component

fancy_checkbox.css



Action processed.

Plug-in: Simple Checkbox - Preparations

com_oracle_apex_simple_checkbox.min.js

1 - 1

Created: 87 seconds ago - ADMIN
Updated: Now - ADMIN

Upload File

Use this page to associate files like images, CSS- or Javascript files with your plug-in. To associate a file with your plug-in, click Browse, select the file, and click Upload.

Use `p_plugin.file_prefix` in your plug-in code to reference a plug-in file.

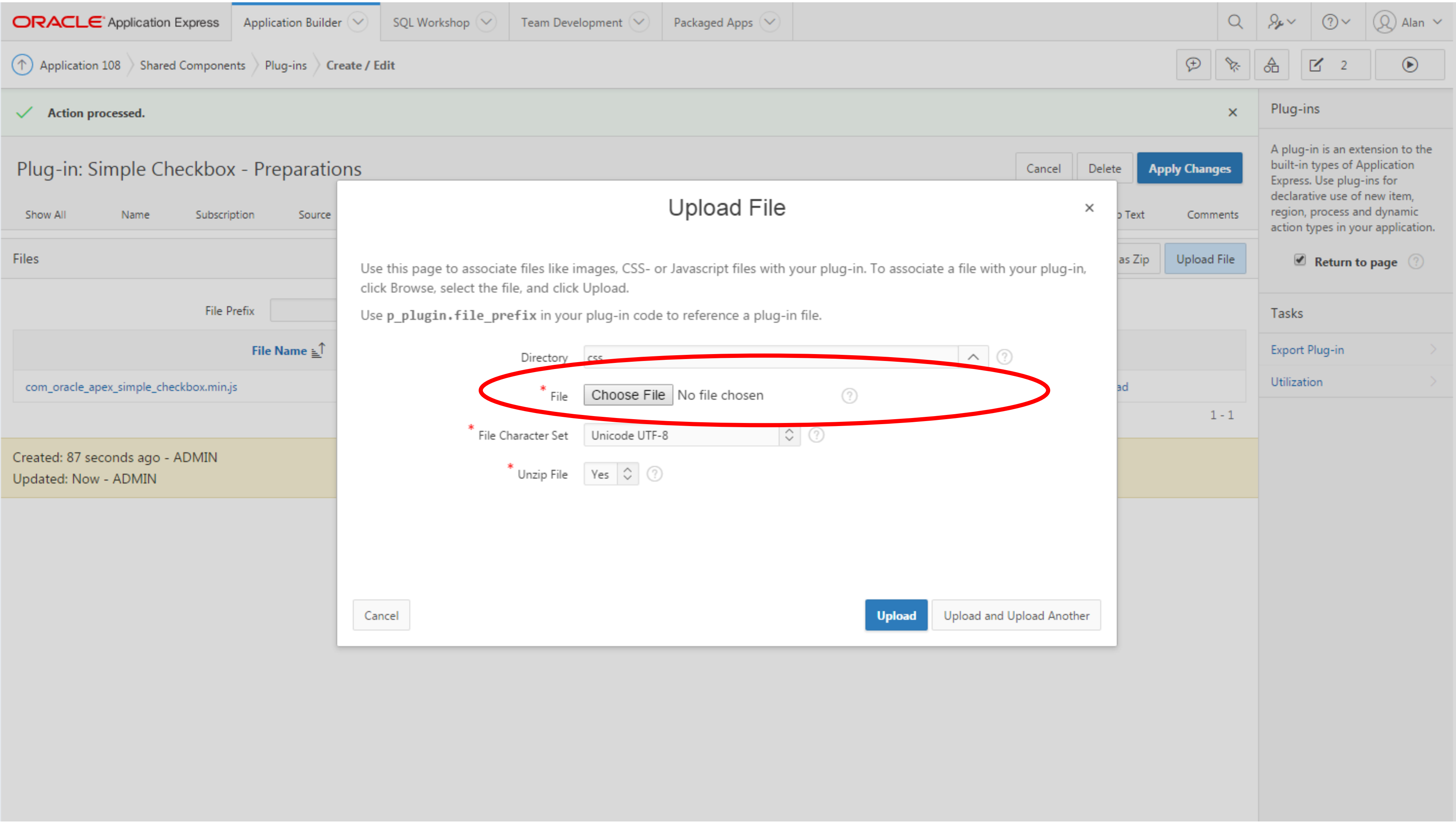
Directory	css
* File	Choose File No file chosen
* File Character Set	Unicode UTF-8
* Unzip File	Yes

Plug-ins

A plug-in is an extension to the built-in types of Application Express. Use plug-ins for declarative use of new item, region, process and dynamic action types in your application.

☑ Return to page

Tasks



Action processed.

Plug-in: Simple Checkbox - Preparations

File Prefix

File Name
com_oracle_apex_simple_checkbox.min.js

1 - 1

Created: 87 seconds ago - ADMIN
Updated: Now - ADMIN

Upload File

Use this page to associate files like images, CSS- or Javascript files with your plug-in. To associate a file with your plug-in, click Browse, select the file, and click Upload.

Use `p_plugin.file_prefix` in your plug-in code to reference a plug-in file.

Directory	css
* File	Choose File No file chosen
* File Character Set	Unicode UTF-8
* Unzip File	Yes

Plug-ins

A plug-in is an extension to the built-in types of Application Express. Use plug-ins for declarative use of new item, region, process and dynamic action types in your application.

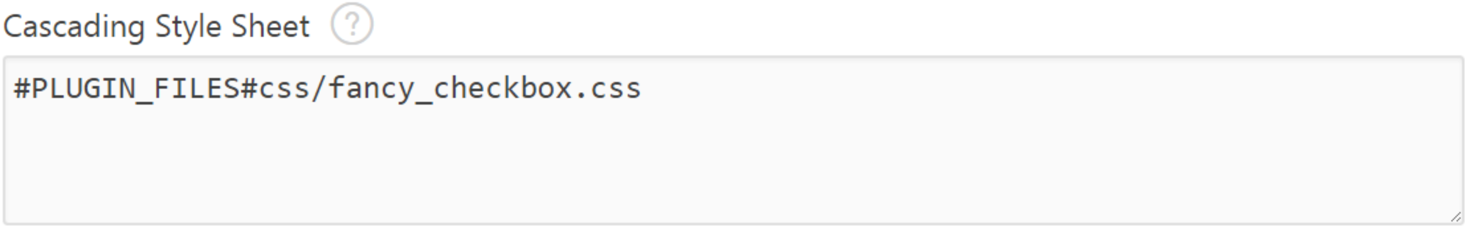
☑ Return to page

Tasks

```
begin

   ...

   -- Add the css file
   apex_css.add_file(
               p_name      => 'fancy_checkbox'
              ,p_directory => p_plugin.file_prefix ||
                                 'css/'
   );

   ...

end render;
```

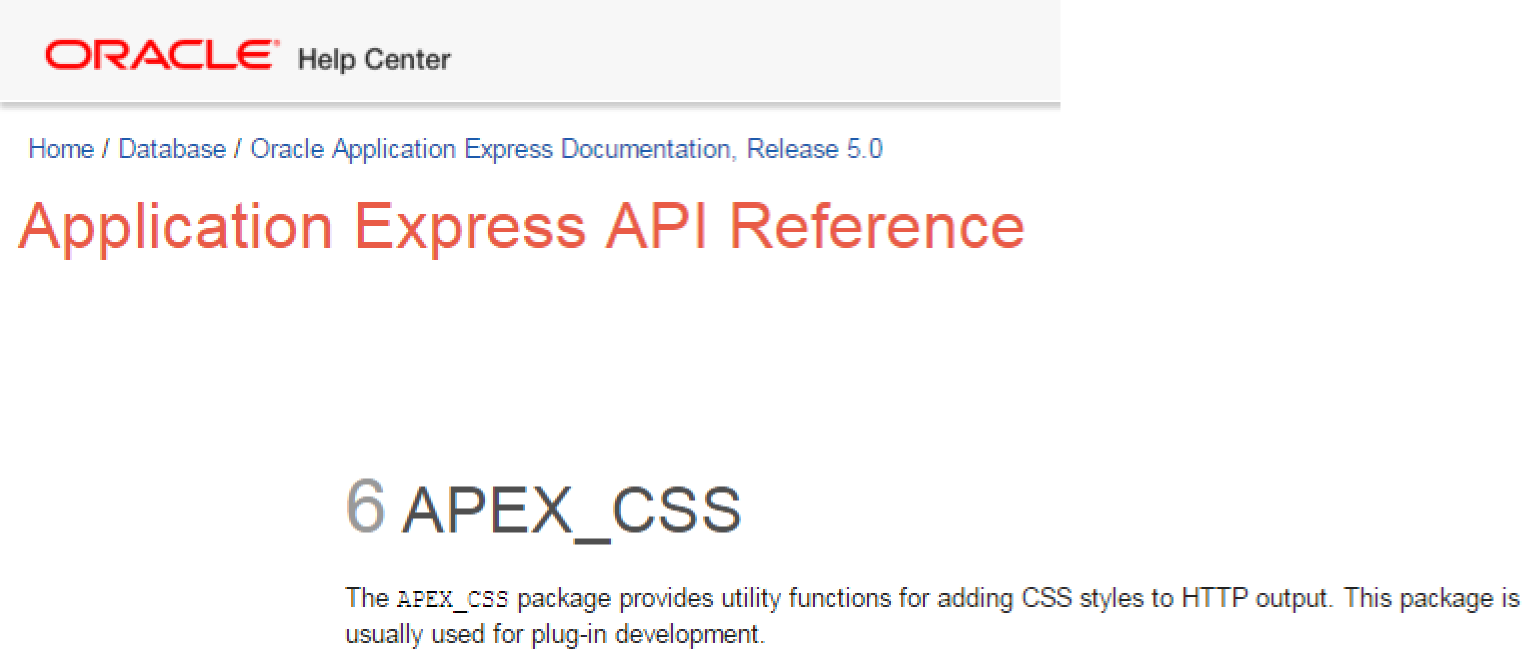
File URLs to Load

```
-- Add the css file
apex_css.add_file(
            p_name        => 'fancy_checkbox'
           ,p_directory => p_plugin.file_prefix ||
                              'css/'
);
```



Cascading Style Sheet

```
#PLUGIN_FILES#css/fancy_checkbox.css
```



ORACLE Help Center

Home / Database / Oracle Application Express Documentation, Release 5.0

Application Express API Reference

6 APEX_CSS

The `APEX_CSS` package provides utility functions for adding CSS styles to HTTP output. This package is usually used for plug-in development.

DEMO

- ☑ **Plug-in Repository**
- ☑ **Create from scratch**
- ☑ **Copy of existing Plug-in**

ORACLE®

DATABASE

October 2015

Oracle Application Express Statement of Direction

Oracle Application Express 5.1

Oracle Application Express 5.1 will focus on both new features and enhancements to existing functionality, and it is planned to incorporate the following:

- Interactive Grid – A new rich, client-side centric region type that allows editing multiple rows of data in a dynamic, JSON-enabled grid, and supports multiple grids on a single page.

https://apex.world

http://apex-plugin.com

http://www.oracle.com/technetwork/developer-tools/apex/application-express/apex-plug-ins-182042.html

https://docs.oracle.com/cd/E59726_01/index.htm www.database-consultancy.nl

Alan Arentsen
alan.arentsen@database-consultancy.nl

alanarentsen.blogspot.com