

## New APIs in Oracle Application Express 4.2.3

Joel R. Kallman

07/24/2013

Version 1

### APEX\_LANG.CREATE\_LANGUAGE\_MAPPING Procedure

Use this procedure to create the language mapping for the translation of an application. Translated applications are published as new applications but are not directly editable in the Application Builder.

#### Syntax

```
APEX_LANG.CREATE_LANGUAGE_MAPPING (  
    p_application_id IN NUMBER,  
    p_language IN VARCHAR2,  
    p_translation_application_id IN NUMBER )
```

#### Parameters

Parameter	Description
p_application_id	The ID of the application for which we wish to create the language mapping. This is the ID of the primary language application.
p_language	IANA language code for the mapping. Examples include en-us, ca-fr, ja, he.
p_translation_application_id	Unique integer value for the ID of the underlying translated application. This number cannot end in 0.

#### Example

The following example demonstrates the creation of the language mapping for an existing APEX application:

```
begin  
    --  
    -- If running from SQL*Plus, we need to set the environment  
    -- for the APEX workspace associated with this schema. The  
    -- call to apex_util.set_security_group_id is not necessary if  
    -- you're running within the context of the Application Builder  
    -- or an APEX application.
```

```

--
for c1 in (select workspace_id
            from apex_workspaces) loop
    apex_util.set_security_group_id( c1.workspace_id );
    exit;
end loop;

-- Now, actually create the language mapping
apex_lang.create_language_mapping(
    p_application_id      => 63969,
    p_language            => 'ja',
    p_translation_application_id => 778899 );

commit;

--
-- Print what we just created to confirm
--
for c1 in (select *
            from apex_application_trans_map
            where primary_application_id = 63969) loop
    dbms_output.put_line( 'translated_application_id: ' ||
c1.translated_application_id );
    dbms_output.put_line( 'translated_app_language:   ' ||
c1.translated_app_language );
end loop;

end;
/

```

## APEX\_LANG.UPDATE\_LANGUAGE\_MAPPING Procedure

Use this procedure to update the language mapping for the translation of an application. Translated applications are published as new applications but are not directly editable in the Application Builder.

### Syntax

```

APEX_LANG.UPDATE_LANGUAGE_MAPPING (
    p_application_id IN NUMBER,
    p_language IN VARCHAR2,
    p_new_trans_application_id IN NUMBER )

```

### Parameters

Parameter	Description
p_application_id	The ID of the application for which we wish to update the language mapping. This is the ID of the primary language application.
p_language	IANA language code for the existing mapping. Examples include en-us, ca-fr, ja, he. The language of the mapping cannot be updated with this procedure, only the new translation application ID.
p_new_trans_application_id	New unique integer value for the ID of the underlying translated application. This number cannot end in 0.

## Example

The following example demonstrates the update of the language mapping for an existing APEX application and existing translation mapping.

```

begin
  --
  -- If running from SQL*Plus, we need to set the environment
  -- for the APEX workspace associated with this schema. The
  -- call to apex_util.set_security_group_id is not necessary if
  -- you're running within the context of the Application Builder
  -- or an APEX application.
  --
  for c1 in (select workspace_id
              from apex_workspaces) loop
    apex_util.set_security_group_id( c1.workspace_id );
    exit;
  end loop;

  -- Now, update the language mapping
  apex_lang.update_language_mapping(
    p_application_id      => 63969,
    p_language            => 'ja',
    p_new_trans_application_id => 881188 );

  commit;

  --
  -- Print what we just updated to confirm
  --
  for c1 in (select *
              from apex_application_trans_map
              where primary_application_id = 63969) loop
    dbms_output.put_line( 'translated_application_id: ' ||
c1.translated_application_id );
    dbms_output.put_line( 'translated_app_language:   ' ||
c1.translated_app_language );
  end loop;

```

```
end;  
/
```

## APEX\_LANG.DELETE\_LANGUAGE\_MAPPING Procedure

Use this procedure to delete the language mapping for the translation of an application. This procedure will also delete all translated strings in the translation repository for the specified language and mapping. Translated applications are published as new applications but are not directly editable in the Application Builder.

### Syntax

```
APEX_LANG.DELETE_LANGUAGE_MAPPING (  
    p_application_id IN NUMBER,  
    p_language IN VARCHAR2 )
```

### Parameters

Parameter	Description
p_application_id	The ID of the application for which we wish to delete the language mapping. This is the ID of the primary language application.
p_language	IANA language code for the existing mapping. Examples include en-us, ca-fr, ja, he.

### Example

The following example demonstrates the deletion of the language mapping for an existing APEX application and existing translation mapping.

```
begin  
    --  
    -- If running from SQL*Plus, we need to set the environment  
    -- for the APEX workspace associated with this schema. The  
    -- call to apex_util.set_security_group_id is not necessary if  
    -- you're running within the context of the Application Builder  
    -- or an APEX application.  
    --  
    for c1 in (select workspace_id  
                from apex_workspaces) loop  
        apex_util.set_security_group_id( c1.workspace_id );  
        exit;  
    end loop;
```

```

-- Now, delete the language mapping
apex_lang.delete_language_mapping(
    p_application_id      => 63969,
    p_language            => 'ja' );

commit;

--
-- Print what we just updated to confirm
--
for c1 in (select count(*) thecount
           from apex_application_trans_map
           where primary_application_id = 63969) loop
    dbms_output.put_line( 'Translation mappings found: ' ||
c1.thecount );
end loop;

end;
/

```

## APEX\_LANG.SEED\_TRANSLATIONS Procedure

Use this procedure to seed the translation repository for the specified application and language. This will populate the translation repository with all of the new, updated and removed translatable strings from your application. A seed and publish process should be performed each time you wish to update the translated version of your application and synchronize it with the primary application.

### Syntax

```

APEX_LANG.SEED_TRANSLATIONS (
    p_application_id IN NUMBER,
    p_language IN VARCHAR2 )

```

### Parameters

Parameter	Description
p_application_id	The ID of the application for which we wish to update the translation repository. This is the ID of the primary language application.
p_language	IANA language code for the existing translation mapping. Examples include en-us, ca-fr, ja, he.

## Example

The following example demonstrates the seeding process of the translation repository for an APEX application and language:

```
begin
  --
  -- If running from SQL*Plus, we need to set the environment
  -- for the APEX workspace associated with this schema. The
  -- call to apex_util.set_security_group_id is not necessary if
  -- you're running within the context of the Application Builder
  -- or an APEX application.
  --
  for c1 in (select workspace_id
              from apex_workspaces) loop
    apex_util.set_security_group_id( c1.workspace_id );
    exit;
  end loop;

  -- Now, seed the translation repository
  apex_lang.seed_translations(
    p_application_id      => 63969,
    p_language            => 'ja' );

  commit;

  --
  -- Print out the total number of potentially translatable strings
  --
  for c1 in (select count(*) thecount
              from apex_application_trans_repos
              where application_id = 63969) loop
    dbms_output.put_line( 'Potentially translatable strings found: ' ||
c1.thecount );
  end loop;

end;
/
```

## APEX\_LANG.PUBLISH\_APPLICATION Procedure

Use this procedure to publish the translated version of an application. This will create an underlying, hidden replica of the primary application and merge the strings from the translation repository in this new application. A seed and publish process should be performed each time you wish to update the translated version of your application and synchronize it with the primary application.

This application will not be visible in the Application Builder. It can only be published and exported but not directly edited.

## Syntax

```
APEX_LANG.PUBLISH_APPLICATION (  
    p_application_id IN NUMBER,  
    p_language IN VARCHAR2 )
```

## Parameters

Parameter	Description
p_application_id	The ID of the application for which we wish to publish and create the translated version. This is the ID of the primary language application.
p_language	IANA language code for the existing translation mapping. Examples include en-us, ca-fr, ja, he.

## Example

The following example demonstrates the publish process for an APEX application and language:

```
begin  
    --  
    -- If running from SQL*Plus, we need to set the environment  
    -- for the APEX workspace associated with this schema. The  
    -- call to apex_util.set_security_group_id is not necessary if  
    -- you're running within the context of the Application Builder  
    -- or an APEX application.  
    --  
    for c1 in (select workspace_id  
                from apex_workspaces) loop  
        apex_util.set_security_group_id( c1.workspace_id );  
        exit;  
    end loop;  
  
    -- Now, publish the translated version of the application  
    apex_lang.publish_application(  
        p_application_id    => 63969,  
        p_language           => 'ja' );  
  
    commit;  
  
end;  
/
```

## APEX\_LANG.UPDATE\_MESSAGE Procedure

Use this procedure to update a translatable text message for the specified application.

### Syntax

```
APEX_LANG.UPDATE_MESSAGE (  
    p_id IN NUMBER,  
    p_message_text IN VARCHAR2 )
```

### Parameters

Parameter	Description
p_id	The ID of the text message.
p_message_text	The new text for the translatable text message.

### Example

The following example demonstrates an update of an existing translatable text message.

```
begin  
    --  
    -- If running from SQL*Plus, we need to set the environment  
    -- for the APEX workspace associated with this schema. The  
    -- call to apex_util.set_security_group_id is not necessary if  
    -- you're running within the context of the Application Builder  
    -- or an APEX application.  
    --  
    for c1 in (select workspace_id  
                from apex_workspaces) loop  
        apex_util.set_security_group_id( c1.workspace_id );  
        exit;  
    end loop;  
  
    -- Locate the ID of the specific message and update it with the new text  
    for c1 in (select translation_entry_id  
                from apex_application_translations  
                where application_id = 63969  
                  and translatable_message = 'TOTAL_COST'  
                  and language_code = 'ja') loop  
  
        apex_lang.update_message(  
            p_id => c1.translation_entry_id,  
            p_message_text => 'The total cost is: %0');  
        commit;
```



```

        exit;
    end loop;

end;
/

```

## APEX\_LANG.UPDATE\_TRANSLATED\_STRING Procedure

Use this procedure to update a translated string in the seeded translation repository.

### Syntax

```

APEX_LANG.UPDATE_TRANSLATED_STRING (
    p_id IN NUMBER,
    p_language IN VARCHAR2
    p_string IN VARCHAR2 )

```

### Parameters

Parameter	Description
p_id	The ID of the string in the translation repository.
p_language	IANA language code for the existing translation mapping. Examples include en-us, ca-fr, ja, he. The language of the mapping cannot be updated with this procedure, only the new translation application ID.
p_string	The new value for the string in the translation repository.

### Example

The following example demonstrates an update of an existing string in the translation repository.

```

begin
    --
    -- If running from SQL*Plus, we need to set the environment
    -- for the APEX workspace associated with this schema. The
    -- call to apex_util.set_security_group_id is not necessary if
    -- you're running within the context of the Application Builder
    -- or an APEX application.

```

```

--
for c1 in (select workspace_id
            from apex_workspaces) loop
    apex_util.set_security_group_id( c1.workspace_id );
    exit;
end loop;

-- Locate all strings in the repository for the specified application
which are 'Search' and change to 'Find'
for c1 in (select id
            from apex_application_trans_repos
            where application_id = 63969
              and dbms_lob.compare(from_string, to_nclob('Search')) = 0
              and language_code = 'ja') loop

    apex_lang.update_translated_string(
        p_id      => c1.id,
        p_language => 'ja',
        p_string => 'Find');
    commit;
    exit;
end loop;

end;
/

```