SkillBuilders Save Before Exit

A Dynamic Action Plug-in for Oracle Application Express (APEX)

Table of Contents

Overview

Intro

Have you ever filled out a form and accidently left the page without saving your work? It is actually quite easy to do. Thankfully, browsers have features to help prevent this from happening. The problem, however, is that taking advantage of those features can be quite difficult - until now.

The SkillBuilders Save Before Exit plug-in allows you to notify your users if they leave a page without saving their work. Simply install the plug-in and add it to your page!

Features at a Glance

1. Detects modifications made to items on the page
2. Warning message displayed if user tries to leave page without clicking a button
   a. Browser is closed
   b. Back button is pressed
   c. Link is clicked
   d. Other...
3. Setting to highlight modified items
4. Setting to customize the warning displayed (ignored in recent versions of Firefox)
5. Setting to select elements that allow a user to leave the page without any warnings
6. Setting to select elements that should not trigger a warning when modified
7. Methods to control whether or not a user is warned via JavaScript

Version History

- 1.0.0 – 7/16/2010
    - Initial release
- 2.0.0 – 11/12/2010
    - Implemented true “modification detection” to avoid false-positives
    - Exposed main selector used to prevent warnings via No Warning Selector setting
    - Added Ignore Change Selector setting to allow modifications to certain elements without generating a warning
    - Added enableWarning and disableWarning methods for warning control via JavaScript
    - Compressed JavaScript for faster loading
- 2.0.1 – 1/19/2011
    - Updated documentation (Tabular Form Integration and more)
- 2.1.0 – 2/22/2011
    - Added support for FCKEditor instances (used before CKEditor)
    - Added “fix” for IE issue that caused users to be prompted two times rather than just once (Thanks to Bill Morris for both identifying the problem as well as a solution)
    - Switched logic for “No Warning” elements to only temporarily disable warning (allows for selecting help labels in IE without disabling overall)

  - Added Disable Milliseconds attribute to configure how long the warning message logic should be suppressed
  - Renamed the No Warning Selector attribute to Disable Warning Selector to fit better with the new Disable Milliseconds attribute
- 2.2.0 – 7/20/2011
  - Instrumented plug-in with debug output (writes to browser console)
  - Fixed bug related to false positive on rich text editors
  - Removed support for older FCKEditor instances (used before CKEditor)
- 2.2.1 – 7/27/2011
  - Changed “$” reference in PL/SQL code to “apex.jQuery”
- 2.3.0 – 9/2/2011
  - Added changeDetected, enableWarningForce, disableWarningForce, and warningForceEnabled methods to allow more programmatic control
  - Added support for APEX shuttles
- 3.0.0 – 1/14/2012
  - Added setting to highlight items that were modified (thanks to Alex Kluev and the APEX development team for the idea)
  - Implemented “fix” for IE issue that caused users to be prompted two times rather than just once (this was removed at some point but has been added back in)
  - Renamed Ignore Change Selector setting to Ignore Modifications Selector. Also renamed changeDetected method to modificationDetected (similar naming convention now used throughout plug-in)
  - Added modifiedItems method which returns a jQuery object of the APEX items that were modified
- 3.0.1 – 1/25/2012
  - Fixed bug that created a false positive for a change on shuttles if the shuttle was initialized without a value (thanks to Stefan-B Schick for identifying the problem)

Feature Requests and Bugs

If you would like to see additional functionality added to the plug-in, or if you have found a bug, please let us know by emailing support@skillbuilders.com.

License

The SkillBuilders Save Before Exit plug-in is currently available for use in all personal or commercial projects under both MIT and GPL licenses. This means that you can choose the license that best suits your project and use it accordingly. Both licenses have been included with this software.

Legal Disclaimer

The program(s) and/or file(s) are supplied as is. The author disclaims all warranties, expressed or implied, including, without limitation, the warranties of merchantability and of fitness for any purpose. The author assumes no liability for damages, direct or consequential, which may result from the use of these program(s) and/or file(s).

Installation and Configuration

Intro

Installation of the SkillBuilders Save Before Exit plug-in can be broken into 2 easy steps:

1. Plug-in Installation
2. Configuration

Installation

With this installation package there is a plug-in installation file named:

- dynamic_action_plugin_com_skillbuilders_save_before_exit.sql

Navigate to "Shared Components > Plug-ins" and click **Import >**. From there you can follow the menu to upload and install the plug-in using the file above. After installing the plug-in you will be redirected to the plug-in edit screen.

Configuration

Navigate to a page, typically a form page, where you would like to warn your users that they should save their work before leaving.

1. Create a new Dynamic Action
   a. Using the Tree View, scroll to the bottom of Page Rendering and right-click **Dynamic Actions**, then left-click **Create**.
   b. Using the Component View, scroll down to the bottom of Page Rendering and left-click the Create button in the Dynamic Actions region (looks like a "+").
2. Select **Advanced**.

3. Click **Next >**.
4. Enter any name you would like to use.
5. Click **Next >**.
6. Set **Event** to **Page Load**.
7. Click **Next >**.
8. Set **Action** to **SkillBuilders Save Before Exit (3.0.0) [Plug-in]**.
9. Change settings if you wish (see Attributes).
10. Click **Next >**.
11. Click **Create**.

Settings

Warning Message

This setting contains the warning message displayed to users if they try to leave the page without using one of the elements selected via No Warning Selector attribute. Note that some browsers will display a default message even when a custom message is supplied.

Disable Warning Selector

This setting allows developers to specify which elements should allow end users to leave the page without being prompted, even if modifications have been made. The default ":button" selector selects all button elements which are typically "valid exits" from a page. In some cases the default selector will not work. Examples include:

- A custom theme is used that does not use standard buttons
- Some buttons on a page are not considered valid exits without warning

In these cases, or any others that may apply, the selector can be modified as needed. The value specified can be any valid jQuery selector. As an example, the following is the HTML used for a button in one theme prior to APEX 4.0:

```
<table cellspacing="0" cellpadding="0" border="0" summary=""
class="t1Button">
<tbody>
   <tr>
      <td class="t1L">
         <a href="javascript:apex.submit('SAVE')">
            <img width="4" height="24" alt=""
src="/i/themes/theme_1/button_left.gif">
         </a>
      </td>
      <td class="t1C">
         <a href="javascript:apex.submit('SAVE')">Apply
Changes</a>
      </td>
      <td class="t1R">
         <a href="javascript:apex.submit('SAVE')">
            <img width="4" height="24" alt=""
src="/i/themes/theme_1/button_right.gif">
         </a>
      </td>
   </tr>
</tbody>
</table>
```

This button actually has three anchor tags that submit the page when clicked. The following jQuery selector could be used to select all three:

```
table.t1Button a
```

Ignore Modifications Selector

By default this plug-in attempts to detect modifications on all visible items. This includes:

1. Standard
   a. Input elements of type text, file, password, radio, checkbox, and hidden
   b. Textarea elements
   c. Select elements
   d. Shuttles
2. Special
   a. CKEditor items (new rich text editors)

The Ignore Modifications Selector setting accepts a jQuery selector used to exclude elements from the modification detection process. The following jQuery selector could be used to exclude a rich text editor:

```
#P3_TICKET_HISTORY
```

Disable Milliseconds

Elements selected via the Disable Warning Selector will temporarily disable the warning message when clicked. Use the Disable Milliseconds setting to control the length of time that the warning is disabled. The default of 250 milliseconds (1/4 second) is typically long enough to prevent issues that can occur in

IE while remaining short enough that the warning will re-enable if the user stays on the page for whatever reason.

Highlight Modified Item(s)

This setting is used to control whether or not items that were modified are highlighted. This can help make the reason for which the warning message was displayed more obvious to the end user.

Methods

The modificationDetected method can be used to check elements (not set to ignore) to see if they have modified since the page loaded. It can be called as follows (returns Boolean):

```
$(document).apex_save_before_exit('modificationDetected');
```

The modifiedItems method can be used to get a jQuery object with the APEX items that were modified on the page (returns jQuery). An optional options object can be passed to control highlighting of the items on the page (defaults to false).

```
$(document).apex_save_before_exit('modifiedItems');

$(document).apex_save_before_exit('modifiedItems' , {highlight:
true});
```

enableWarning and disableWarning, can be used to programmatically specify whether modification detection should occur via JavaScript. Even if the warning is enabled with these methods, a modification must be detected for a prompt to occur. They can be called as follows:

```
$(document).apex_save_before_exit('enableWarning');

$(document).apex_save_before_exit('disableWarning');
```

enableWarningForce and disableWarningForce, can be used to programmatically specify whether modification detection should occur via JavaScript. If force is enabled then a user will be prompted to save regardless of any modification actually occurring. They can be called as follows:

```
        $(document).apex_save_before_exit('enableWarningForce');

        $(document).apex_save_before_exit('disableWarningForce');
```

warningForceEnabled can be used to determine the enableWarningForce method has been called. It can be called as follows (returns Boolean):

```
        $(document).apex_save_before_exit('warningForceEnabled');
```

Tabular Form Integration

Tabular Forms in APEX have built in functionality similar to that offered by the SkillBuilders Save Before Exit plug-in. This functionality is needed in Tabular Forms because some built in controls, such as header sorting links and pagination links, could cause a user to loose unsaved modifications to data when used. This built in functionality works great, even highlighting the cells that have modifications when prompting the user to save.

Unfortunately, problems can arise if SkillBuilders Save Before Exit plug-in is not configured correctly to work with Tabular Forms. Here are some guidelines to properly integrate the two:

1. Enable Partial Page Rendering (PPR) for the Tabular Form. This setting can be found under Layout and Pagination in the Report Attributes. PPR allows the tabular form to sort and paginate without causing the entire page to refresh. This prevents the plug-in from doing anything in these cases. The built in functionality will continue to work with PPR enabled.
2. Make sure that the sorting links, pagination links, as well as the “Add” button of the Tabular Form are NOT selected in the No Warning Selector attribute of the plug-in. Again, since PPR is preventing the page from reloading it is not necessary to include these elements. In fact, including them can even cause an issue that prevents the plug-in from working when it should.

   Warning: The default No Warning Selector will include the Add button so make sure to change this. You’ll need to explicitly select the buttons and/or other elements that should allow a user to leave without being warned of unsaved modifications.

About SkillBuilders and the Author

About SkillBuilders

SkillBuilders is known for excellent IT training and consulting. Our instructors are always industry-tested experts and outstanding teachers who have set an unsurpassed standard of excellence. SkillBuilders' roots can be traced to 1985 when our founder, Dave Anderson, embarked on his career as an independent IT consultant, instructor and author. Dave and his colleagues built a small, energetic and growing company, headquartered in South Kingstown, RI.

About the Author

Dan McGhan

Dan is a Senior Developer and Instructor with SkillBuilders. He suffers from Compulsive Programing Disorder which is believed to be linked to his balding. Having started his development career in the land of MySQL and PHP, he was only too happy to have stumbled upon APEX. Since then, he's dedicated his programming efforts to learning more about Oracle and web based technologies in general.

Dan is an Oracle Application Express Certified Expert, an Oracle PL/SQL Developer Certified Associate, as well as an Oracle ACE. In addition to his "day job", he is one of the top contributors to the APEX forum, maintains his own Oracle and APEX blog, and is a regular presenter at various events and user group meetings including ODTUG Kaleidoscope and APEXposed, the New York, New England, and Suncoast Oracle User Groups. His most recent addiction, as you may have guessed, is developing plug-ins for APEX.

When not programming, Dan may be found studying languages other than those used for development, notably Spanish. He's also been sited at various venues dancing Salsa with his wife, Sonia, and even enjoying an occasional cigar, a time when Sonia prefers not to be around.