SkillBuilders Super LOV

An Item Plug-in for Oracle Application Express (APEX)

Table of Contents

Overview

Intro

The built-in Popup LOV item in APEX can be a powerful tool when the number of possible selections is too high for other item types that use LOVs, such as a select list. However, people often wish to display more information when making a selection than this item type allows. In the past, developers worked around this issue by either concatenating values in the display column or by creating a custom solution.

Custom solutions provided the ultimate flexibility but required a lot of knowledge and were not always easily transferable to other items, pages, or applications. The SkillBuilders Super LOV plug-in is designed to make these issues a thing of the past while providing a rich, web 2.0 experience.

Features at a Glance

- Multiple column LOV support
  - Displays multiple columns when LOV is open
  - Column level selection and searching when LOV open
  - Allows for selection of a display column and a return column
  - Allows for additional column to item value mappings
- Rich client-side user interface
  - Skinned with jQuery UI themes (currently 25 themes to choose from)
  - Transition effects with adjustable speeds
  - Uses a modal dialog over a traditional popup window
  - Select from a variety of loading images, or use your own

Version History

- 1.0.0 – 12/17/2010
  - Initial release
- 1.0.1 – 12/17/2010
  - Fixed bug that prevented viewing modal on Webkit browsers (Safari, Chrome, etc.)
- 1.1.0 – 12/23/2010
  - Improved Dialog sizing and resizing algorithms
  - Added “Validate Value” attribute
  - Added “Display Null Values as” attribute
  - Added “Read Only” attributed
  - Added support for several display attributes
- 1.2.0 – 01/13/2011
  - Implemented integrated icons in the main input
  - Added “Effects Speed” attribute
  - Added “Default Search Column Number” attribute

    - Fixed bug that prevented Cascading LOV attribute from working
- 1.2.1 – 01/20/2011
    - Fixed bug that prevented Cascading LOV from working with Popup Keys
- 1.3.0 – 03/6/2011
    - Fixed bug that prevented searches with whitespace or special characters from working
    - Fixed CSS issues related to icons in input
    - Reorganized many settings to allow for more functionality
    - Added ability to map value from columns to additional items
    - Added ability to enable/disable Clear Confirm feature
- 2.0.0 – 1/12/2012
    - Added ability to make item "enterable"
    - Added ability to control item and make selections via keyboard
    - Added methods for hide/show and enable/disable
    - Added optional performance upgrades
- 2.0.1 – 3/6/2012
    - Fixed bug where mapped values were being inadvertently escaped (thanks to Deepthi Bommera for identifying the issue)
    - Fixed bug where LOV would not load if APEX was run in languages other than English (thanks to Dennis Verdaasdonk and Yuri Arts for identifying the issue and to Patrick Wolf for helping me see the error of my ways)
    - Fixed bug that prevented keyboard actions from working when report regions existed on the page (thanks to Dennis Verdaasdonk for identifying the issue)
    - Fixed but that prevented modal dialog from opening in IE when browser was in quirks mode (thanks to Martin D'Souza for identifying the issue and providing a solution)
    - Added support for native Dynamic Actions including Set Value, Get Value, Hide, Show, Enable, and Disable (thanks to Martin D'Souza for the suggestion)
    - Renamed fieldset id to work with $x_ItemRow (thanks to Martin D'Souza for the suggestion)
    - Fixed bug where change event fired when twice when in enterable mode. Now native change event is suppressed and a change event is triggered when a change is made according to standard use of the plug-in (thanks to Martin D'Souza for identifying the issue)
    - Set input value to be escaped while rendering input to prevent XSS vulnerabilities (thanks to Patrick Wolf for identifying this issue)
    - Fixed bug that generated checksum error when readonly items were on the page (thanks to Paul for identifying the issue and to Patrick Wolf for providing a solution)
- 2.0.2 – 3/7/2012
    - Fixed bug that required two presses of the enter key to open the modal while in enterable mode.
    - Fixed a bug that loaded full JS file over minified version.

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 Super LOV 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

Installation

Standard

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

- item_type_plugin_com_skillbuilders_super_lov.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 the plug-in has been installed successfully you will be redirected to the plug-in edit screen.

Optional Performance Upgrades

After installing the plug-in you can make it more performant by either using packaged code, files on the file server, or both.

If you wish to test the package based code, simply compile the package spec and body and add the package name and a period to the beginning of each of the function names in the Callbacks region. At that point you could even safely remove the contents of the PL/SQL Code in the Source region.

If you wish to test the files on the apps server, simply copy the files in the "server" directory to a directory on the apps server and change the File Prefix attribute of the plug-in to point to that directory.

Example:
If you copy the contents of "server" to **/images/plugins/apex_super_lov_2_0**
Then set the "File Prefix" attribute of the plug-in to **#IMAGE_PREFIX#plugins/apex_super_lov_2_0/**

Configuration

Once installed, this plug-in can be used as a native APEX component. When creating a new item, select the “Plug-ins” option and then choose the plug-in on the next page.

Here’s a sample query that could be used for the LOV. Notice how column aliases are used to generate formatted column headings.

```
SELECT empno AS "Emp No.",
   ename AS "Employee",
   job AS "Job",
   sal AS "Salary"
FROM emp
```

See Configuration Settings for details on how the application and component settings affect the plug-in.

Configuration Settings

Application Settings

Application settings are used to configure all instances of a plug-in within an application. These settings are accessed by editing the plug-in within the Shared Components. This plug-in has the following application settings:

Search Type

Use this setting to control how search strings are used to filter the LOV result set.

Theme

Use this setting to control the basic color scheme of the calendar. If no theme is selected then the default theme will be used.

Tip: The Theme setting adds a link to a CSS file on the APEX page where the component is used. If possible, it's best to manually add the link to the CSS file in the page template where the component is used. This can improve performance/caching while decreasing the likelihood of a conflict due to another plug-in linking to a different CSS file. This setting can then be disabled by selecting the null value (- Select Theme-).

Tip: The plug-in is skinned using the jQuery UI CSS Framework . To learn more or create a custom theme visit: http://docs.jquery.com/UI/Theming and http://jqueryui.com/themeroller.

Loading Image Type

Use this setting to choose between a default or custom loading image. The loading image is displayed when the LOV is opened, before the result set appears. There are a number of default loading images that can be used (see Loading Image next) but you can use your own as well.

Loading Image

Use this setting to specify which loading image you would like to use. Based on the Loading Image Type selection, you will either choose from a number of default images or you will specify the path/name to a custom image.

Effects Speed

Use this setting to specify the speed at which the modal dialog should perform certain effects such as sizing, resizing, and fading. Selecting “instant” will essentially disable any effects.

Use Clear Confirm

Use this setting to enable or disable the Clear Confirm feature. Clear Confirm requires the user to click the clear button twice to clear the selected value. This is done to help prevent accidental clearings that would require the LOV to be reopened.

When No Data Found Message

Use this setting to specify what message should be displayed to users when the LOV query fails to retrieve any results.

Component Settings

Component settings are used to configure an individual instance of a plug-in within an application. These settings are accessed by editing the component as you would a native APEX component. This plug-in has the following component settings:

Use Value Validation

Use this setting to enable or disable the Value Validation feature. Value Validation will re-check the submitted value against the LOV. If the value is not found then the validation will fail and the user will see a validation error message.

Dialog Title

Use this setting to explicitly set the title of the dialog. If no value is supplied then the item's label will be used.

Item Display & Return Columns

Use this setting to specify which column should be used for the item's display value and which column should be used for the item's return value. The value should be a comma separated pair of numbers where the numbers refer to the column in the LOV query. The first number should be the display column and the second number should be the return column. The display column will be used as the default search column.

Searchable Columns

Use this setting to specify which columns should be displayed in the select list of columns that allows users to filter the LOV result set. The value should be a comma separated list of numbers where the numbers refer to columns in the LOV query. If no value is supplied then all columns will be searchable. If a value is supplied then only those columns will be searchable. The display column (defined via Item Display & Return Columns) will always be searchable and will be the default search column.

Hidden Columns

Use this setting to specify which columns should be hidden when the LOV is displayed. The value should be a comma separated list of numbers where the numbers refer to columns in the LOV query. If no value is supplied then all columns will be visible. The display column (defined via Item Display & Return Columns) will always be visible.

Map From Columns

Use this setting to specify which columns should be used to map values to other items (see Map To Items). The value should be a comma separated list of numbers where the numbers refer to columns in the LOV query. Both visible and hidden columns can be used when mapping to other items.

Map To Items

Use this setting to specify which items should be used when mapping values from columns (see Map From Columns). The value should be a comma separated list of item names. The order of the items in Map To Items should match the order of the columns in Map To Columns.

Enterable

Use this setting to make the item "enterable". If enterable, users will be able to type in the actual textbox. The display column (defined via Item Display & Return Columns) will be the default search column against which values entered will be validated. If one match is found then the display and return values will be set accordingly. If no match is found or multiple matches are found the modal dialog will open so that the user can make a selection.

Max Rows Per Page

Use this setting to specify the maximum number of records that should be displayed at one time.

Show Null Values As

Use this setting to specify how null values should be displayed in the result set.

Methods

The disableItem and enableItem methods can be used to programmatically disable or enable the item. Example(s):

```
$('#P1_ITEM_NAME').apex_super_lov('enable');

$('#P1_ITEM_NAME').apex_super_lov('disable');
```

The hideItem and showItem methods can be used to programmatically hide or display the item and its label. Example(s):

```
$('#P1_ITEM_NAME').apex_super_lov('hide');

$('#P1_ITEM_NAME').apex_super_lov('show');
```

The hideRow and showRow methods can be used to programmatically hide or display the item, its label, as well as any other elements that exist in the same table row. Example(s):

```
$('#P1_ITEM_NAME').apex_super_lov('hideRow');

$('#P1_ITEM_NAME').apex_super_lov('showRow');
```

The getValuesByReturn and setValuesByReturn methods can be used to programmatically fetch return values and set the item using the return values. getValuesByReturn will return a JSON object with the results from the fetch. setValuesByReturn goes a step further using the return values to set the display and return values for the item. Example(s):

```
$('#P1_ITEM_NAME').apex_super_lov('getValuesByReturn', '101');

$('#P1_ITEM_NAME').apex_super_lov('setValuesByReturn', '999');
```

About SkillBuilders and the Authors

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 Authors

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.

Jason Lyle

Jason graduated in 2006 from Bowling Green State University with a degree in computer science. For him, coding is more of a puzzle to be solved rather than work to be done. He has dabbled in PHP, MySQL, Java, C++, TCL/TK, web technologies, PL/SQL (quite obviously), and anything else he can get his hands on. Right now, Jason is focusing most of his time on Oracle Databases and APEX applications.

Outside of the world of code, Jason enjoys trying almost anything. However, mostly he enjoys snow skiing, snowboarding, white water rafting, canoeing, kayaking, volleyball, cooking, reading sci-fi or fantasy books, and traveling. In his international travels, he has only made it to a couple countries in South America, but hopes to keep increasing his list of places visited. He also hopes to learn to surf, water ski, and wake board.