

Editable List View Panels



February 21, 2014

© Guidewire Software, Inc. All rights reserved. Do not distribute without permission.

Guidewire training materials contain Guidewire proprietary information that is subject to confidentiality and non-disclosure agreements. You agree to use the information in this manual solely for the purpose of training to implement Guidewire software solutions. You also agree not to disclose the information in this manual to third parties or copy this manual without prior written consent from Guidewire. Guidewire training may be given only by Guidewire employees or certified Guidewire partners under the appropriate agreement with Guidewire.

Lesson objectives

- By the end of this lesson, you should be able to:
 - Configure a list view panel so that its rows can be modified
 - Configure a list view panel so that rows can be added to and removed from it

This lesson uses the notes section for additional explanation and information.

To view the notes in PowerPoint, choose View→Normal or View→Notes Page.

If you choose to print the notes for the lesson, be sure to select "Print hidden slides."



Lesson outline

- Editable list view panels
- Iterator buttons

Editable list view panels

- List view panels enable end users to manipulate data
 - Adding new rows
 - Removing existing rows
 - Modifying existing rows

<input type="checkbox"/>	* Bank Name	* Routing Number	* Account Number
<input checked="" type="checkbox"/>	Bank of America	155-900	93222-21389
<input type="checkbox"/>	Wells Fargo	991-112	4339214
<input type="checkbox"/>	US Bank	100-112	87321-12121

Editable hierarchy

<input type="checkbox"/> * Bank Name	* Routing Number	* Account Number
<input checked="" type="checkbox"/> Bank of America	155-900	93222-21389
<input type="checkbox"/> Wells Fargo	991-112	4339214
<input type="checkbox"/> US Bank	100-112	87321-12121

list view
panel is
editable
row iterator
is editable
row is
editable
cell is
editable

- List view panels can include editable cells and non-editable cells
 - Routing Number is editable
 - Created On is read-only
- For a cell to be editable, the row, row iterator, and list view panel in which the cell resides must also all be editable

© Guidewire Software, Inc. All rights reserved. Do not distribute without permission.

5

G U I D E W I R E

The following summarizes the default value of the editable property for each element:

- A list view panel's editable property is blank. The property is not required. (A list view panel with no specified editable property is editable.)
- A row iterator's editable property is blank, but this property is required. To have editable cells, it must be set to true, or to a condition that evaluates to true.
- A row's editable property is blank. The property is not required. (A row with no specified editable property is editable.)
- A cell's editable property is false. To have the cell be editable, you must set the property to true, or to a condition that evaluates to true.

In practice, the two elements you must explicitly set to editable to get editable cells are the row iterator and the cells that are to be made editable.

Multiple options for location of toolbars

The screenshot shows a 'Details' form for 'Bank Accounts'. At the top, there are 'Update' and 'Cancel' buttons, with a red arrow pointing to them labeled 'Screen level toolbar'. Below this is a tabbed interface with 'Person Info', 'Phone & Addresses', 'Bank Accounts' (selected), and 'Analysis'. Under the 'Bank Accounts' tab, there are 'Add' and 'Remove' buttons, with a red arrow pointing to them labeled 'List view panel level toolbar'. Below the buttons is a table with three columns: 'Bank Name', 'Routing Number', and 'Account Number'. The table contains three rows: 'Bank of America', 'Wells Fargo', and 'US Bank'. The first row is highlighted in blue.

<input type="checkbox"/> * Bank Name	* Routing Number	* Account Number
<input checked="" type="checkbox"/> Bank of America	155-900	93222-21389
<input type="checkbox"/> Wells Fargo	991-112	4339214
<input type="checkbox"/> US Bank	100-112	87321-12121

- Multiple places where you can place a toolbar
 - Associate the toolbar with the container it acts on
- Edit Buttons go in toolbar at screen level
- Iterator Buttons go with list view panel toolbar

Because the PCF architecture is hierarchical in nature, there is often more than one place that a toolbar could go. For example, the screenshot above shows toolbars at both the screen level and list level. Typically, the best place for a toolbar is the highest place in the hierarchy of containers that you want it to affect. For example, if you need the same toolbar on every card of a card view, then you will provide users with a more intuitive interface if you place a single toolbar at or above the card view level as opposed to putting individual toolbars on every card.

Putting lists into edit mode

- To modify cell data, either:
 - Location containing list view panel must always be in edit mode, or
 - List view or one of its parent containers must have Edit Buttons

Details

Person Info Phone & Addresses **Bank Accounts** Analysis

Bank Name	Routing Number	Account Number	Account Type	Verified?	Created On ↑
Bank of America	155-900	93222-21389	Checking	Verified	08/14/2013
Wells Fargo					
US Bank					

Details

Update Cancel

Person Info Phone & Addresses **Bank Accounts** Analysis

Add Remove

<input type="checkbox"/>	* Bank Name	* Routing Number	* Account Number	* Account Type	Verified?	* Created On ↑
<input checked="" type="checkbox"/>	Bank of America	155-900	93222-21389	Checking	Verified	08/14/2013
<input type="checkbox"/>	Wells Fargo	991-112	4339214	Checking	Verified	08/14/2013
<input type="checkbox"/>	US Bank	100-112	87321-12121	Checking	Verified	08/14/2013

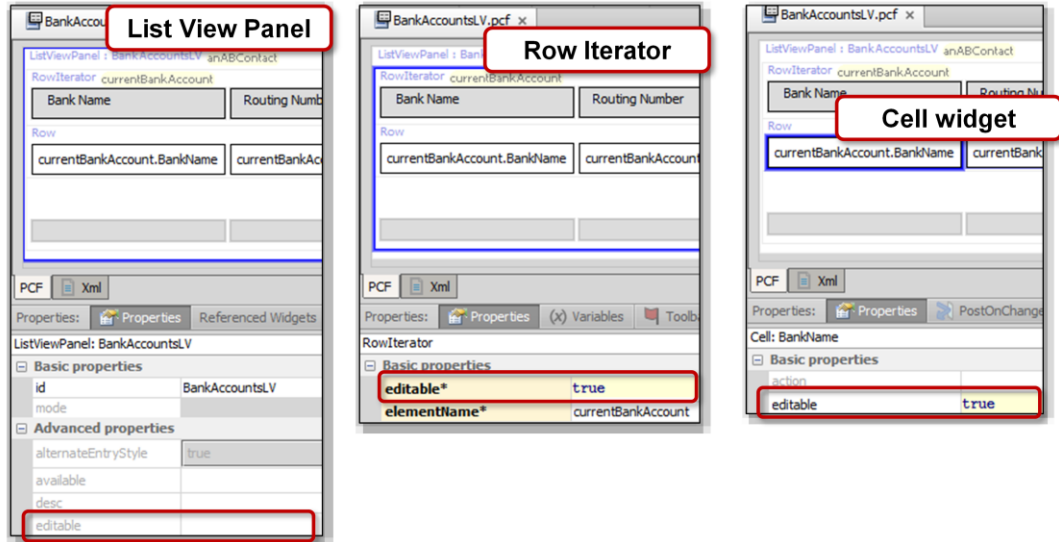
© Guidewire Software, Inc. All rights reserved. Do not distribute without permission.

7

G U I D E W I R E

It is possible to set a location so that it is always in edit mode. To configure this, set the location's `startInEditMode` property to true and `alwaysInEditMode` property to true. A location that is always in edit mode has only Update and Cancel buttons—there is no Edit button. Whenever either of the buttons is clicked, the changes are committed or canceled, and the location immediately reverts to edit mode.

Example of cell with editable data



List view panel
editable property
default is "true"

Row iterator must
specify editable =
true

Cell must specify
editable = true

© Guidewire Software, Inc. All rights reserved. Do not distribute without permission.

8

G U I D E W I R E

The screenshots above show the configuration within a list view panel needed to make the data in the Bank Name cells editable. Note the following about the BankAccountsLV list view panel:

- The list view panel itself is editable. Editable is either blank or set to true.
- The row iterator is editable.
- The cell widget itself is editable.

This presumes that all container widgets that contain BankAccountsLV are also editable.

Lesson outline

- Editable list view panels
- Iterator buttons

Iterator buttons and row iterators

- Properties of row iterator govern functionality of iterator buttons (Add|Remove)
 - toAdd - action to take when Add is clicked
 - toRemove - action to take when Remove is clicked

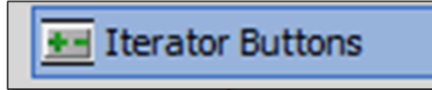
The diagram illustrates the connection between the properties of a `RowIterator` and the actions of the `Add` and `Remove` buttons in a table. On the left, a 'Properties' panel for `RowIterator` lists 'Basic properties': `editable*`, `elementName*`, `toAdd`, `toRemove`, and `value*`. A green gear icon with two circular arrows is positioned between the properties and the table. Red arrows point from the `toAdd` and `toRemove` properties to the corresponding buttons in the table. The table, titled 'Bank Accounts', has columns for 'Bank Name', 'Routing Number', and 'Account Number'. The first row is selected.

	Bank Name	Routing Number	Account Number
<input checked="" type="checkbox"/>	Bank of America	155-900	93222-21389
<input type="checkbox"/>	Wells Fargo	991-112	4339214
<input type="checkbox"/>	US Bank	100-112	87321-12121

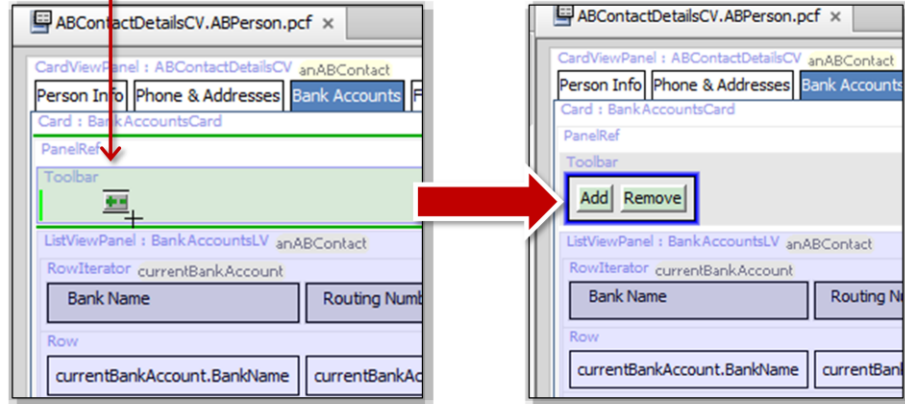
Steps to adding iterator buttons

1. Add iterator buttons to toolbar
2. Link iterator buttons to iterator
3. Specify toAdd property
4. Specify toRemove property
5. Configure checkbox behavior
6. Deploy PCFs

Step 1: Add iterator buttons to toolbar

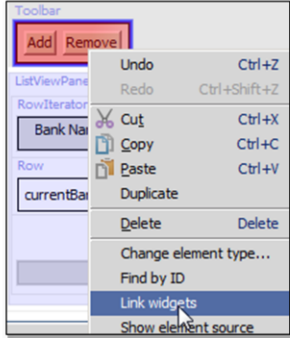


**Includes both
Add and Remove buttons**

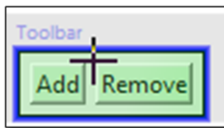


The Iterator Buttons widget contains both buttons needed to add and remove rows. The buttons also already have the appropriate visibility logic. They are visible only when the location is in edit mode.

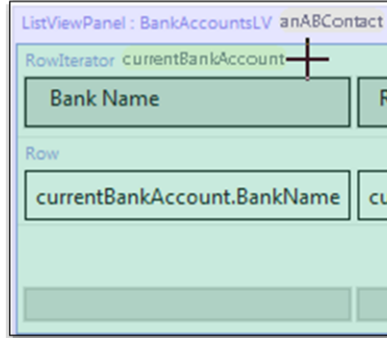
Step 2: Link iterator buttons to iterator



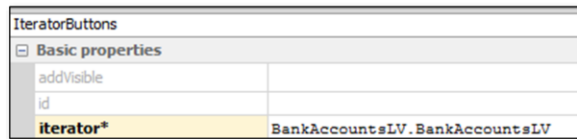
**(1) Context menu
→ Link widgets**



**(2) Click iterator
buttons**



(3) Click row iterator



(4) Set iterator property

© Guidewire Software, Inc. All rights reserved. Do not distribute without permission.

13

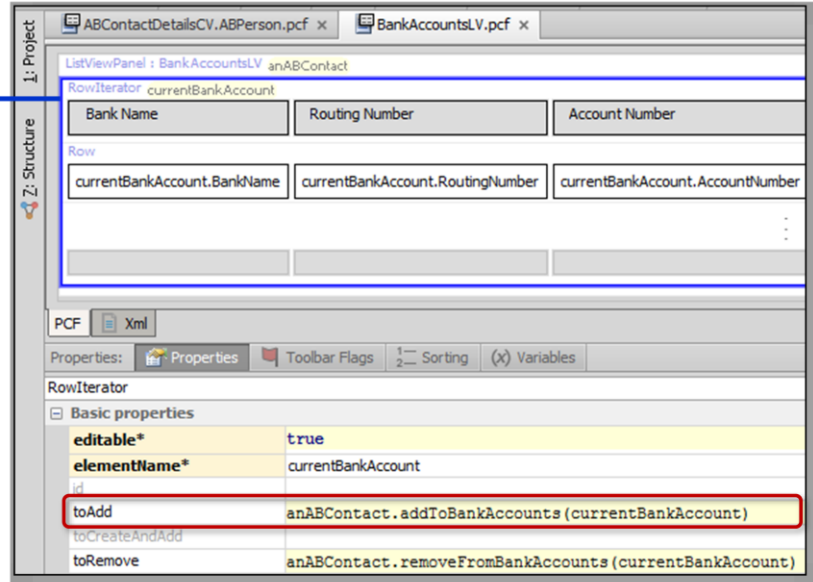
G U I D E W I R E

Add the iterator buttons to the Toolbar first. The mouse cursor changes to a crosshair, and iterator buttons on the PCF turn green to indicate that they are available for linking. If you save the PCF without linking the buttons or give focus to another application before doing so, Studio changes the buttons to red to indicate that they have not been linked.

After you click the iterator buttons, Studio renders any row iterators as green to indicate that they are available for linking. You must click the row iterator itself; even though any contained row (as well as any widget contained by any row) also appears green, they cannot be clicked.

After linking the widgets, configure the iterator property of the iterator buttons to reference the exposed iterator. Click the SmartHelp button to display a list of exposed iterators, and select the appropriate one. Although Studio identifies iterator buttons that have not been linked to a row iterator as an error, the PCF still functions at runtime. Iterator buttons that have not been configured to explicitly reference a particular iterator will implicitly reference and affect whichever row iterator appears after them physically in the page configuration.

Step 3: Specify toAdd property



- Syntax:

parentObj.addToarrayName(elementName)

© Guidewire Software, Inc. All rights reserved. Do not distribute without permission.

14

G U I D E W I R E

For every array on an entity, there are inherently two methods added to that entity:

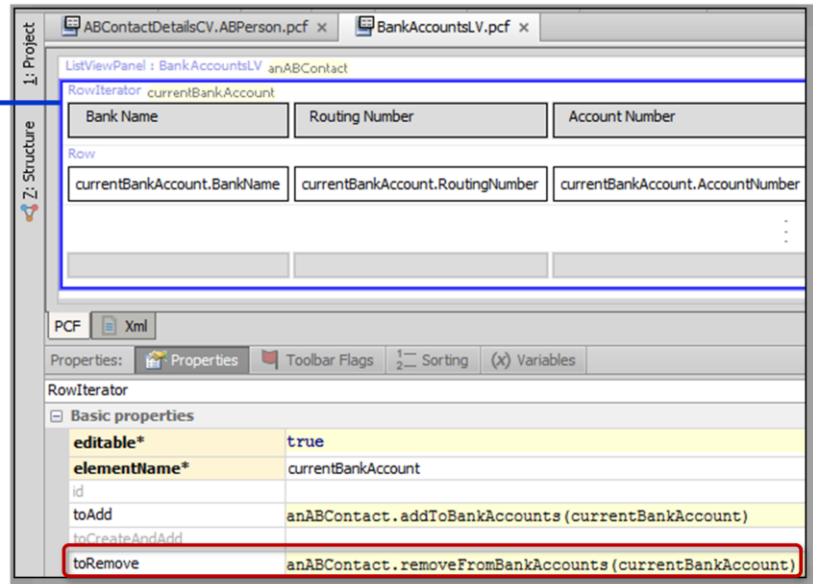
- An addTo<arrayName>() method, which adds a specified object to the named array.
- A removeFrom<arrayName>() method, which removes the specified object from the named array.

The addTo... method is used to enable the functionality of the Add button.

The syntax for the toAdd property is *parentObj.addToArrayName(elementName)*, where:

- *parentObj* is the parent object to which the array belong. In the example above, the BankAccounts array belongs to the anABContact object.
- *ArrayName* is the name of the array.
- *elementName* is the row iterator's element name. For an Add button, this creates a new object of the appropriate type and associates it with the array.

Step 4: Specify toRemove property



- Syntax:
parentObj.removeFromarrayName(elementName)

© Guidewire Software, Inc. All rights reserved. Do not distribute without permission.

15

G U I D E W I R E

For every array on an entity, there are inherently two methods added to that entity:

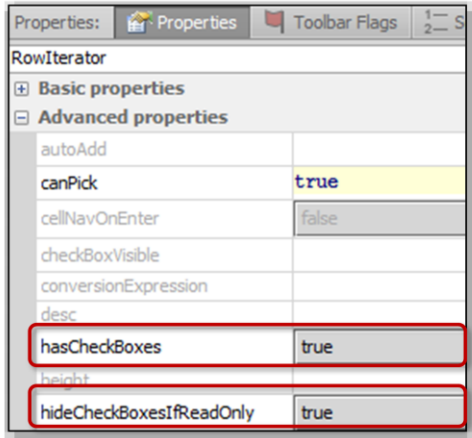
- An `addTo<arrayName>()` method, which adds a specified object to the named array.
- A `removeFrom<arrayName>()` method, which removes the specified object from the named array.

The `removeFrom...` method is used to enable the functionality of the Remove button.

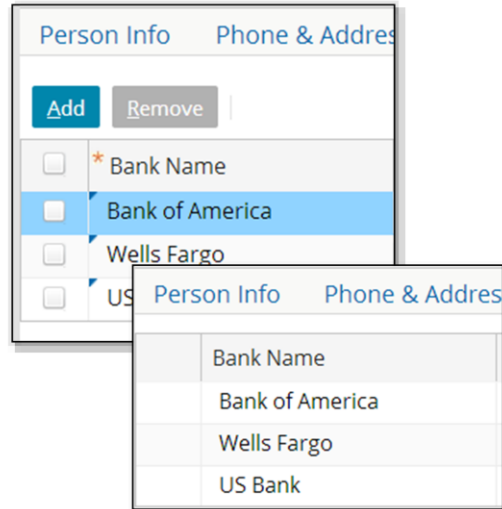
The syntax for the `toRemove` property is *parentObj.removeFromArrayName(elementName)*, where:

- *parentObj* is the parent object to which the array belongs. In the example above, the BankAccounts array belongs to the anABContact object.
- *ArrayName* is the name of the array.
- *elementName* is the row iterator's element name. For a Remove button, this removes the selected row or rows from the array.

Step 5: Configure check box behavior



**check boxes in
edit mode**



**check boxes in
read-only mode**

- hasCheckBoxes
 - Displays check boxes
- hideCheckBoxesIfReadOnly
 - Hides check boxes for read-only

© Guidewire Software, Inc. All rights reserved. Do not distribute without permission.

16

G U I D E W I R E

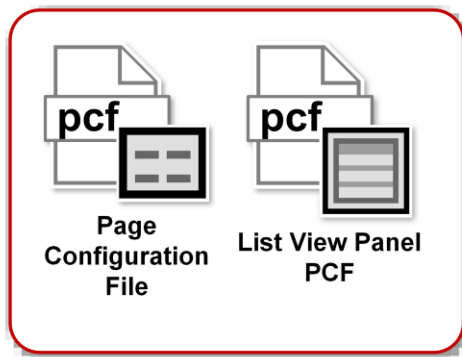
Check boxes enable the Remove button by identifying which rows to remove.

- hasCheckBoxes displays check boxes.
- hideCheckBoxesIfReadOnly hides check boxes in read-only mode.

Step 6: Deploy PCFs

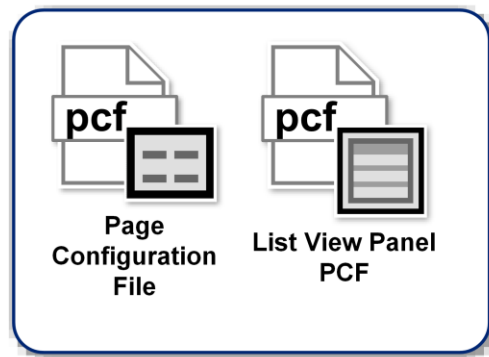
Restart Server

- PCFs read at server startup



Reload PCFs

- ALT+SHIFT+L
 - Internal debug tools enabled
- Internal Tools
 - Reload → Reload PCF Files



© Guidewire Software, Inc. All rights reserved. Do not distribute without permission.

17

G U I D E W I R E

It is also possible to reload PCF files using the Guidewire API and/or internal server tools. The Reload PCF command can be found on the Reload page in Internal Tools. To access Internal Tools, you must log in as an administrator user, e.g., su/gw. Then, use ALT+SHIFT+T. In the tab bar, select Internal Tools → Reload. On the Reload page, click the Reload PCF Files button. The Reload PCF Files button calls the static method `gw.api.tools.InternalToolsUtil.reloadPCFs()`.

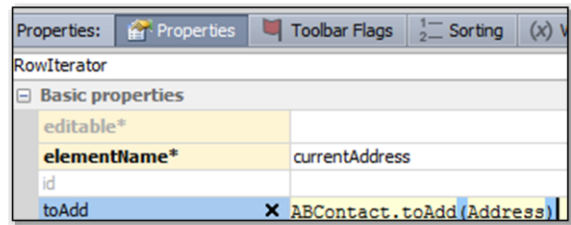
Lesson objectives review

You should now be able to:

- Configure a list view panel so that its rows can be modified
- Configure a list view panel so that rows can be added to and removed from it

Review questions

1. For a cell to be editable, the cell's Editable property must be set to true. What three other elements must also be editable to make the cell editable in the user interface?
2. When you drag an IteratorButtons widget onto a toolbar, how many buttons are added?
3. Assume that an object called anABContact has an Addresses array and an Addresses list view panel whose row iterator is configured as shown. Identify the errors in how the property is configured.



Answers

- 1) The list view panel itself, the list view panel's row iterator, and the row.
- 2) Two: an Add and a Remove button.
- 3) There are four errors: (a) The object is anABContact, not ABContact. (b) The method name starts with "addTo...", not "toAdd...". (c) The name of the array is missing from the end of the method name. (The method should be addToAddresses.) (d) The argument must reference the row iterator's element name, which is currentAddress.

Notices

Copyright © 2001-2014 Guidewire Software, Inc. All rights reserved.

Guidewire, Guidewire Software, Guidewire ClaimCenter, Guidewire PolicyCenter, Guidewire BillingCenter, Guidewire Reinsurance Management, Guidewire ContactManager, Guidewire Vendor Data Management, Guidewire Client Data Management, Guidewire Rating Management, Guidewire InsuranceSuite, Guidewire ContactCenter, Guidewire Studio, Guidewire Product Designer, Guidewire Live, Guidewire DataHub, Guidewire InfoCenter, Guidewire Standard Reporting, Guidewire ExampleCenter, Guidewire Account Manager Portal, Guidewire Claim Portal, Guidewire Policyholder Portal, ClaimCenter, BillingCenter, PolicyCenter, InsuranceSuite, Gosu, Deliver Insurance Your Way, and the Guidewire logo are trademarks, service marks, or registered trademarks of Guidewire Software, Inc. in the United States and/or other countries.

All other trademarks are the property of their respective owners.

This material is confidential and proprietary to Guidewire and subject to the confidentiality terms in the applicable license agreement and/or separate nondisclosure agreement.

This file and the contents herein are the property of Guidewire Software, Inc. Use of this course material is restricted to students officially registered in this specific Guidewire-instructed course, or for other use expressly authorized by Guidewire. Replication or distribution of this course material in electronic, paper, or other format is prohibited without express permission from Guidewire.

Guidewire products are protected by one or more United States patents.