



Detail View Panels



March 7, 2014

© Guidewire Software, Inc. 2001-2014. 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:
 - Describe the functionality of detail view panels
 - Create a new detail view panel
 - Reference a detail view panel from a parent container

This lesson uses the notes section for additional explanation and information.
To view the notes in PowerPoint, select View → Normal or View → Notes Page.
When printing notes, select Note Pages and Print hidden slides.

© Guidewire Software, Inc. 2001-2014. All rights reserved. Do not distribute without permission.

2

G U I D E W I R E



Lesson outline

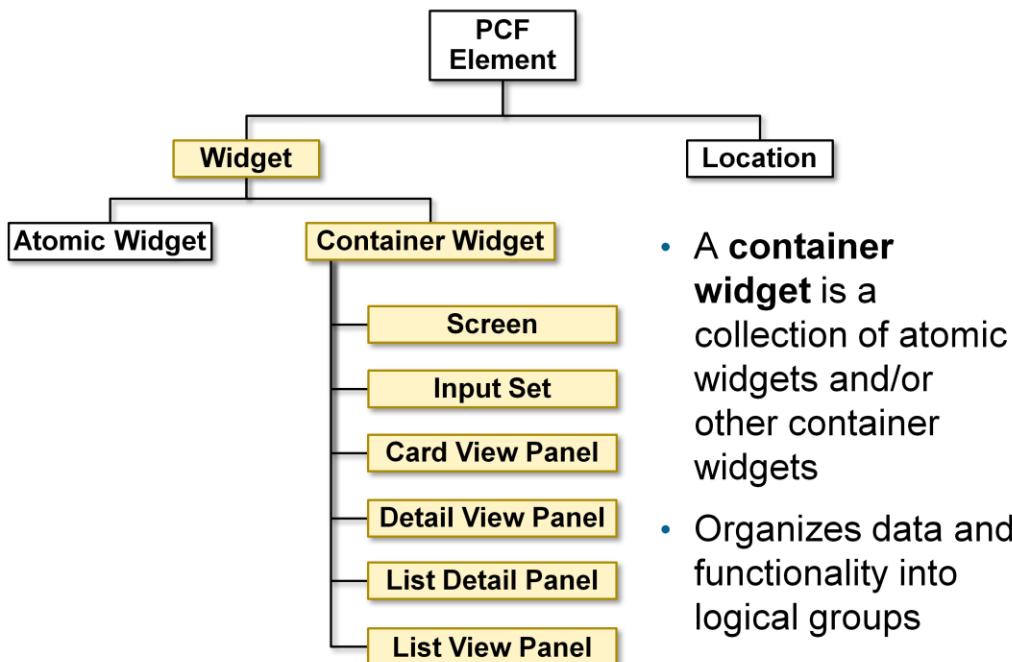
- Detail view panel fundamentals
- Create detail view panels
- Reference detail view panels

© Guidewire Software, Inc. 2001-2014. All rights reserved. Do not distribute without permission.

3

G U I D E W I R E

Container widgets



© Guidewire Software, Inc. 2001-2014. All rights reserved. Do not distribute without permission.

4

G U I D E W I R E

Container widgets hold other widgets. Each one can be defined either in its own file or as a child container within some other PCF element file.

Both Widget and Location are conceptual representations in this diagram. There are no <Widget /> or <Location /> elements. Similarly, both Atomic Widget and Container Widget are conceptual representations. There are no <Atomic Widget /> or <Container Widget /> elements.

The PCF object model is container-based. Each screen element is modeled as an object, which may contain other objects. The hierarchical structure simplifies the task of locating and modifying visual elements. Furthermore, each element can be declared as an independent and therefore reusable element.

Container files are hierarchical

The screenshot shows the Guidewire PCF Editor interface with a hierarchical container structure. On the left, there's a tree view of the container hierarchy. On the right, the details of a selected container are shown in a large panel. The panel has sections for 'Basic Information' and 'Primary Address'. The 'Primary Address' section contains several input fields and dropdown menus. A red bracket on the left side of the panel points to the top-level container, which is a 'Detail View Panel'. A purple bracket on the right side points to a nested 'Input Set' under the 'Primary Address' section, which is further nested within another 'Input Set'. This nested 'Input Set' is shaded dark blue, indicating it is a grandchild container. The panel also includes sections for 'Address Type', 'Description', and 'Valid Until'.

- Top-level container
 - Detail View Panel
- Referenced child container
 - Input Set
- Referenced grandchild container
 - Input Set
- Color darkens with additional nesting of included sections in PCF Editor canvas

When viewing a container file in Studio, if it references a child container and the child container also references a child container, then the "grandchild" container appears in a dark blue shade. In other words, light blue shading denotes a child container and dark blue shading denotes a grandchild container. You can open any referenced file, regardless of whether it's a child or grandchild file, by double-clicking it. Files can be nested as deep as necessary, in which case the color of the shaded area will be progressively deeper. Double-clicking at any level will open the selected file directly. You can toggle the visibility of child files embedded in a parent PCF file. If you disable the representation of the included files, Studio displays the text of the reference expression instead.

Key features of the hierarchical structure are that you always declare one top-level container, you can declare zero to many child containers, and that you can reference zero to many containers declared in other PCFs.

PCF files are in XML format. XML is a markup that is hierarchical in nature. PCF widgets are XML elements organized into a hierarchy of elements. If you click the XML tab in the PCF Editor, you can view the XML hierarchy of XML elements. The XML view is read-only in the PCF Editor.

Detail view panels

The screenshot shows a 'Summary' detail view panel for a contact. At the top, there are buttons for 'Edit' and 'Suggest Least Busy User'. A red box highlights the 'Flag Entries' section, which contains a table with two rows. The table has columns for Action (View/Edit), Date Flagged, Reason, and Date Unflagged. The first row shows a warning icon, 'View/Edit' button, '01/22/2014', 'No email address for this contact.', and an empty 'Date Unflagged' field. The second row shows a 'View' button, '01/13/2014', 'No email address for this contact.', and '01/16/2014'.

| | Action | Date Flagged | Reason | Date Unflagged |
|--|-----------|--------------|------------------------------------|----------------|
| | View/Edit | 01/22/2014 | No email address for this contact. | |
| | View | 01/13/2014 | No email address for this contact. | 01/16/2014 |

- A **detail view panel** is a container widget that allows user to view, and in some cases edit, data for one object and information related to that object

© Guidewire Software, Inc. 2001-2014. All rights reserved. Do not distribute without permission.

6

G U I D E W I R E

Root objects for detail view panels

The screenshot shows a detail view panel titled "ABContactSummaryDV". It has a "Summary" tab selected. Under "Basic Information", there are fields for Name (William Andy), Public ID (ab:5), Created On (01/13/2014), and Assigned User (Bruce Baker). In the "Flag Entries" section, there is a table:

| | View | Date Flagged | Reason | Data Unflagged |
|-----------|------------|------------------------------------|------------|----------------|
| View/Edit | 01/22/2014 | No email address for this contact. | | |
| View | 01/13/2014 | No email address for this contact. | 01/16/2014 | |

Below the table, there is a "Primary Address" section with fields for Address (345 Fir Lane, La Canada 91352, Algeria, Home), Address Type, Description, and Valid Until. A blue cube icon is overlaid on the panel with the text: "Root object is anABContact of the type ABContact".

- A detail view panel often has a root object
 - Container widgets typically have one root object
 - If referenced, a parent container must pass the root object to the detail view panel
- Displays root object data with atomic widgets

© Guidewire Software, Inc. 2001-2014. All rights reserved. Do not distribute without permission.

7

G U I D E W I R E

There may be multiple root objects, but more commonly a related object is referenced by a foreign key field in the root object.

It is possible to create a detail view panel that has no root objects. This sort of detail view panel could display only static labels, system information, or both, however. These types of detail view panels are therefore uncommon.

Detail view panel structure

The screenshot shows a 'Summary' detail view panel. At the top, there are two buttons: 'Edit' and 'Suggest Least Busy User'. Below these are two sections: 'Basic Information' and 'Flag Entries'. The 'Basic Information' section contains fields for Name (William Andy), Public ID (ab:5), Created On (01/13/2014), and Assigned User (Bruce Baker). The 'Flag Entries' section contains a table with two rows. The first row has a warning icon, a 'View' button, a date (01/22/2014), and a reason ('No email address for this contact'). The second row has a 'View' button, a date (01/13/2014), a reason ('No email address for this contact'), and a date (01/16/2014). Two blue rectangles, labeled 'Input Column', are overlaid on the screenshot: one covering the 'Basic Information' section and another covering the 'Flag Entries' section.

- Input columns organize layout and input widgets
- Must have at least one for input widgets

© Guidewire Software, Inc. 2001-2014. All rights reserved. Do not distribute without permission.

8

G U I D E W I R E

A toolbar cannot be a direct child of a detail view panel. Toolbar placement is discussed in later lessons.

Technically, a detail view panel is not required to have any input columns. A detail view panel with atomic widgets does need at least one input column, however. In most cases, every detail view panel has atomic widgets.

Guidewire applications automatically put a vertical blue line in between input columns. This line can be seen in the slide example between the two blue rectangles.

Detail view panel contents

The screenshot shows a 'Detail view panel contents' interface. At the top left is a 'Summary' section with tabs for 'Edit' and 'Suggest Least Busy User'. Below it is a 'Basic Information' section containing fields for Name (William Andy), Public ID (ab:5), Created On (01/13/2014), and Assigned User (Bruce Baker). A horizontal line separates this from a 'Primary Address' section, which includes fields for Address (345 Fir Lane, La Canada 91352, Algeria) and Address Type (Home). A 'Description' field is present but empty. A 'Valid Until' field is also present but empty. To the right of the address section is a 'Flag Entries' table:

| | View | Date Flagged | Reason | Date Unflagged |
|--|-----------|--------------|------------------------------------|----------------|
| | View/Edit | 01/22/2014 | No email address for this contact. | |
| | View | 01/13/2014 | No email address for thi | 01/16/2014 |

Annotations with red boxes and arrows point to specific elements:

- A box labeled 'Input Set' points to the 'Address' and 'Address Type' fields in the 'Primary Address' section.
- A box labeled 'List View Panel' points to the 'Flag Entries' table.
- A box labeled 'Containers for Input widgets' points to the entire 'Primary Address' section.

- Layout widgets make the UI readable and user friendly
- Container widgets organize

© Guidewire Software, Inc. 2001-2014. All rights reserved. Do not distribute without permission.

9

G U I D E W I R E

Input widgets are widgets that display (and in some cases let end users edit) one field of data.

Layout widgets do not display data, but instead are used to make a detail view more readable and/or user-friendly. The two most commonly used widgets of this sort are label widgets, which display label text, and input divider widgets, which are rendered as horizontal lines.

Input Sets are containers for a group of input widgets.

A List View Panel is a container widget that often displays a set of rows that are related to one object or one query.

File and widget

Detail View Panel PCF file

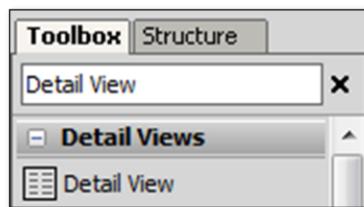
- <DetailViewPanel/> is a top-level PCF element
- File name ends with DV
- Can define root object



Detail View Panel
PCF

DetailViewPanel widget

- Widget is defined in a Screen, Card View Panel, or a List Detail Panel



© Guidewire Software, Inc. 2001-2014. All rights reserved. Do not distribute without permission.

10

G U I D E W I R E

It is possible to define a variable for an DetailViewPanel widget. In many cases, however, a DetailViewPanel widget inherits the root object associated with its parent.

Reusability and inline

Reusability

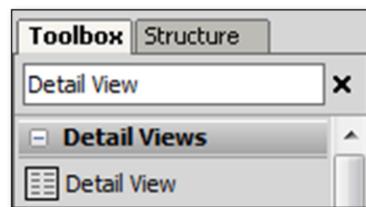
- Detail View Panel is PCF file
- Ideal for multiple references
 - Other PCF files can reference the reusable container using a reference widget



Detail View Panel
PCF

Inline

- Defined as widget
- Single instance usage
 - Not possible to reference in another container



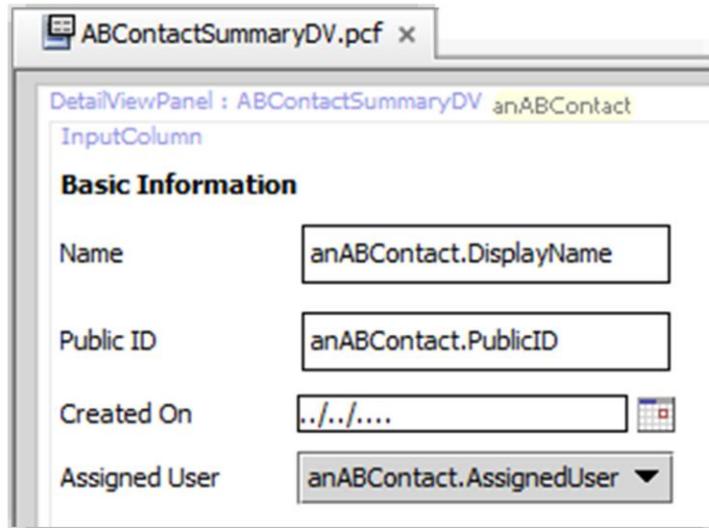
© Guidewire Software, Inc. 2001-2014. All rights reserved. Do not distribute without permission.

11

G U I D E W I R E

When container such as a detail view panel is a top-level container, it is reusable. A top-level container is a PCF file. When a container such as detail view panel is declared as an inline child container, it is not reusable.

Reusable containers



- Top-level container is a PCF file
- Other containers can reference the reusable container

- If the container is likely to be needed in multiple places, create PCF file for container!

© Guidewire Software, Inc. 2001-2014. All rights reserved. Do not distribute without permission.

12

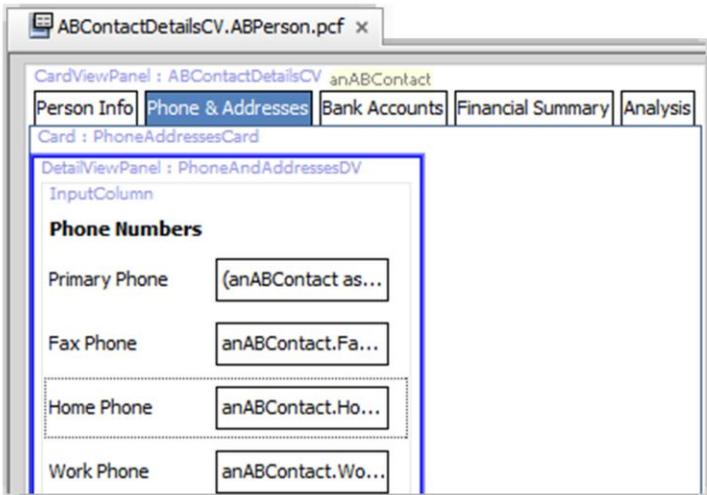
G U I D E W I R E

An inline container is not reusable. In the slide example, ABContactSummaryDV is a top-level container. Other containers can therefore reference it.

In TrainingApp, the ABContactSummaryPage contains a Screen with a Panel Ref that references ABContactSummaryDV.

PanelRef widgets are discussed later in this lesson.

Inline containers



- Inline child containers are not reusable
- Other containers cannot reference an inline container

- If container is unlikely to be needed in multiple places, you can create it as an inline container

© Guidewire Software, Inc. 2001-2014. All rights reserved. Do not distribute without permission.

13

G U I D E W I R E

In the example above, PhoneAndAddressesDV is an inline container. It appears in only in the PhoneAddressesCard of ABContactDetailsCV.ABPerson.pcf.



Lesson outline

- Detail view panel fundamentals
- Create detail view panels
- Referencing detail view panels

Steps to create a Detail View Panel PCF

1. Create the Detail View Panel PCF file
2. Specify the required variables
3. Optionally specify additional properties
4. Add input columns
5. Add input widgets
6. Deploy PCFs

* Slides do not cover the details of creating an inline Detail View Panel. See notes.

© Guidewire Software, Inc. 2001-2014. All rights reserved. Do not distribute without permission.

15

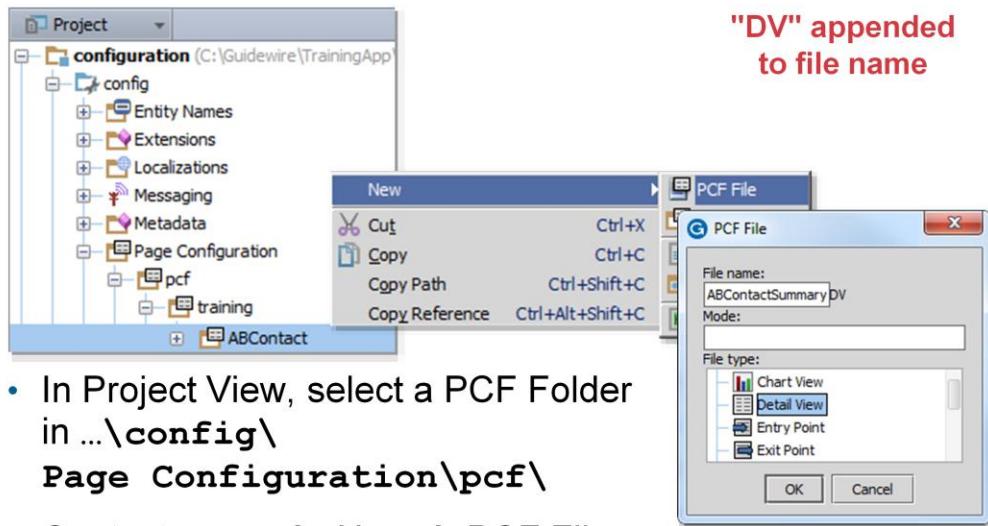
GUIDEWIRE

The steps to create an inline detail view panel are the same as for a reusable detail view (PCF File), except at the very beginning.

Inline detail view panel

1. Add the DetailViewPanel widget to the parent container
2. Optionally specify additional properties
3. Add input columns
4. Add input widgets

Step 1: Create a detail view panel PCF



© Guidewire Software, Inc. 2001-2014. All rights reserved. Do not distribute without permission.

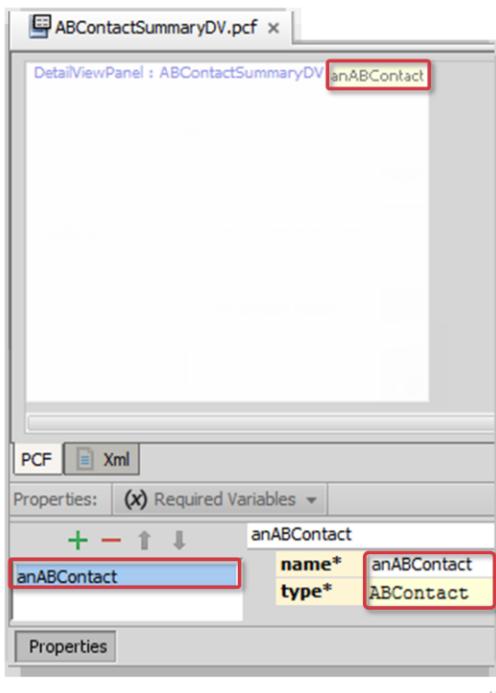
16

G U I D E W I R E

Inline detail view panels

To create an inline detail view, find the DetailViewPanel widget in the PCF Editor toolbox. Drag the list view widget onto an existing screen, list detail panel, or card view panel.

Step 2: Specify required variable(s)



- Required Variables tab
 - Defines data object variable name and type
 - Example:
anABContact is of type ABContact
- Object data can be
 - Data backed (database)
 - Virtual property
- Container data comes from defined variable object(s)
 - Typically, at least one variable
 - Not required



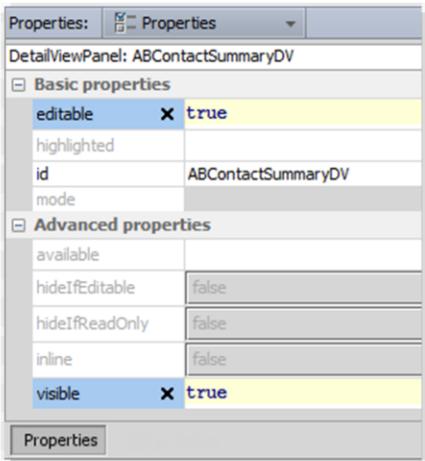
Most container widgets have at least one required object that contains data fields. One way to think of this is that there is at least one root object for a given container.

It is possible to have more than one defined object as it is also possible to not have a required object at all.

Inline detail view panels

You do not need to specify root objects for inline detail view panel. Because a detail view panel can have only one parent container, it automatically inherits the root objects of its parent.

Step 3: Specify additional properties



- **Editable**
 - Makes container and children widget editable
 - Not all container widgets have an explicit editable property
- **Visible**
 - Shows container and all children
 - If false, then hidden

- Blank is default and means that the property inherits the value from parent container or location
- If not defined in hierarchy, then true

© Guidewire Software, Inc. 2001-2014. All rights reserved. Do not distribute without permission.

18

G U I D E W I R E

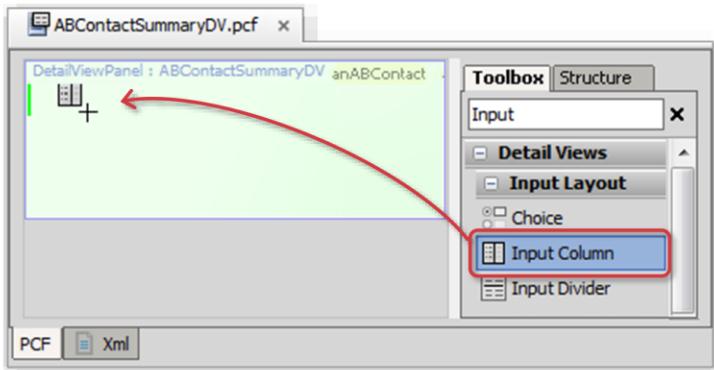
There are two states that a detail view panel can be in: edit mode and read-only mode. If the editable property for a detail view panel is true (or blank, which defaults to true), then the detail view panel can be put into either read-only mode or edit mode. If the editable property for a detail view panel is false, then the detail view panel cannot be put into edit mode and it is always in read-only mode.

By default, there is no way to put detail view panel into edit mode. The "Editable Detail Views" lesson discusses how to add Edit | Update | Cancel buttons so that data can be modified.

Inline detail view panels

This step is the same for standalone and inline detail view panels.

Step 4: Add input column



- Light green line - current place where new widget will go

- An Input Column is a single vertical column in a detail view panel that organizes atomic widgets

© Guidewire Software, Inc. 2001-2014. All rights reserved. Do not distribute without permission.

19

GUIDEWIRE

In most cases, you will want add atomic widgets to a detail view panel in an organized layout. Use an input column widget to organize the atomic widgets.

To add a widget, click its name in the Toolbox and hold the mouse cursor down. As you begin to drag the widget, Studio changes the mouse cursor so that it includes the icon for that widget. Studio places a green line on the canvas at every location on the canvas that it is possible to place the widget. Studio highlights the green line that is nearest on the canvas to the cursor. Studio also overlays in green the element containing the highlighted green line.

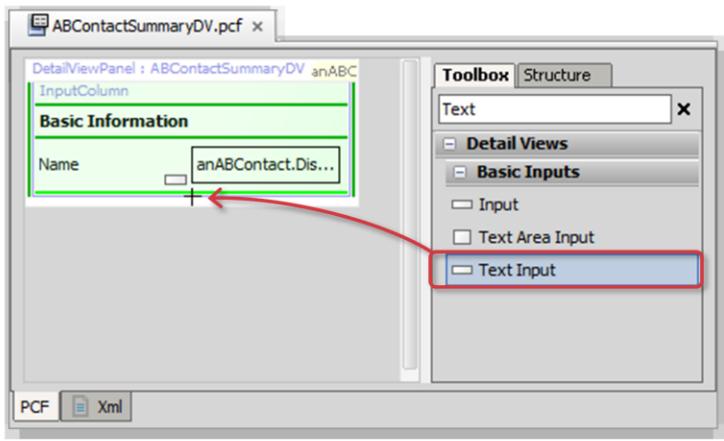
To add an input column, locate the input column tool in the toolbox and drag it onto the canvas. If there are already input columns, dark green bars identify where the new input column could be placed, and a light green bar indicates the current place the new input column will be located.

You can skip this step by dragging an input widget onto the canvas. Studio automatically adds an input column around it if it is the first widget in the stack. It may be easier for developers who are new to PCF configuration to manually drag input columns onto the canvas to ensure that the structure of the detail view panel is correct, however.

Inline detail view panels

It is the same for reusable and inline detail view panels.

Step 5: Add atomic widgets



- Light green line - current place where new widget will go
- Dark green line - places where new widget can go

- Select the best input for the data type from the Toolbox
- Specify required and optional widget properties

© Guidewire Software, Inc. 2001-2014. All rights reserved. Do not distribute without permission.

20

G U I D E W I R E

Notice that the input widgets in the slide example reference the ABContact object as anABContact. The variable name of the root object is anABContact. Input widgets must reference object as named on Required Variables tab.

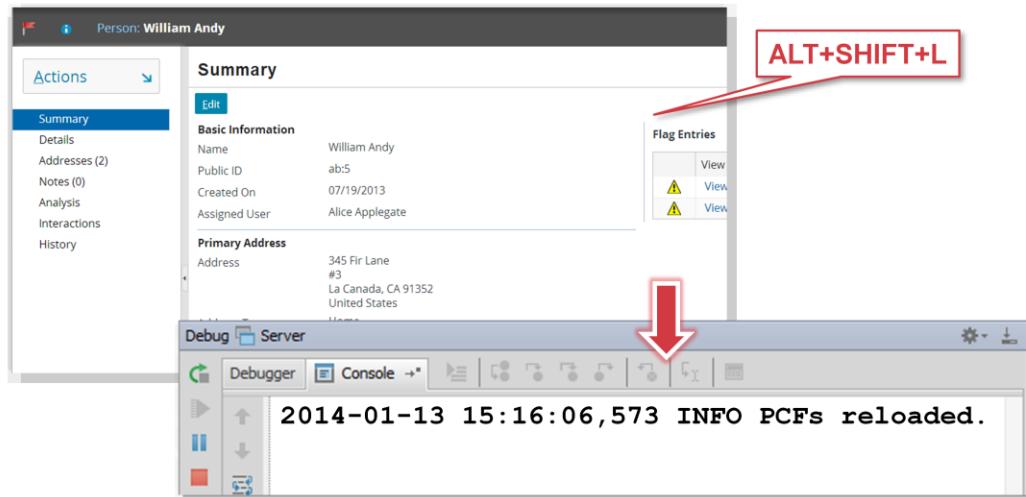
The "Atomic Widgets" lesson discuss adding input widgets and defining the properties on atomic widgets.

Inline detail view panels

It is the same for reusable and inline detail view panels.

Internal debug tools: Reload PCFs

ALT+SHIFT+L



- Reloads all Page Configuration Files
- Command window or Guidewire Studio Console window (Debug / Run) details output

© Guidewire Software, Inc. 2001-2014. All rights reserved. Do not distribute without permission.

21

G U I D E W I R E

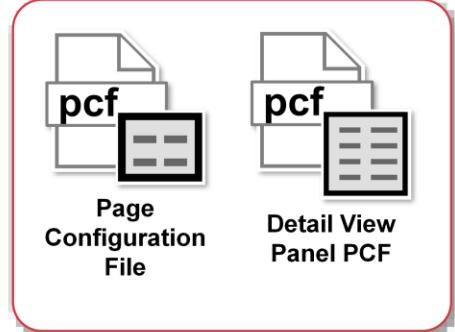
If you are running an open application project in Guidewire Studio and if internal tools are enabled, you can reload all the page configuration files for the server.

If you reload PCF files while in edit mode, you may experience unpredictable results. For the current location, where there is a data modification in progress, the new PCFs may not be reloaded. Therefore, Guidewire recommends reloading PCF files while in read-only mode as it provides for more predictable results.

Step 6: Deploy PCFs

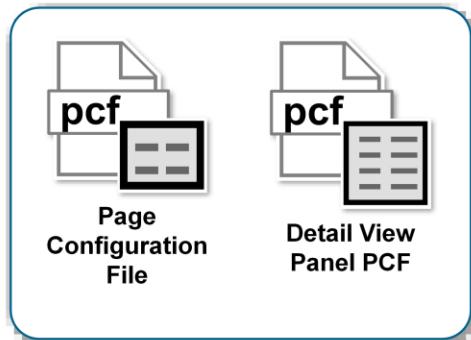
Restart Server

- PCFs read at server startup



Reload PCFs

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



© Guidewire Software, Inc. 2001-2014. All rights reserved. Do not distribute without permission.

22

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()`.

Inline detail view panels

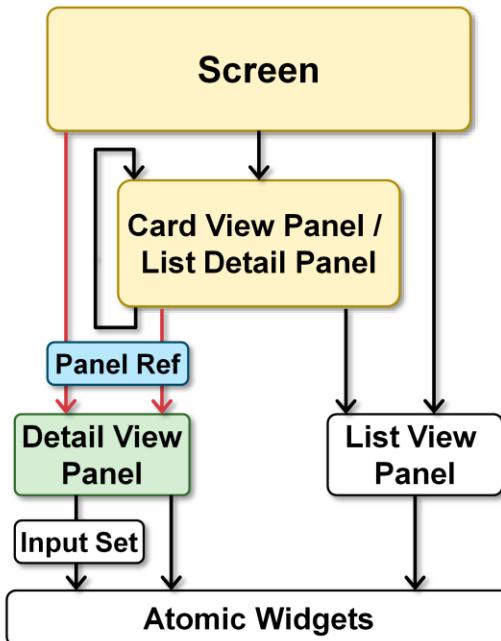
This step is the same for standalone and inline detail view panels. However, for inline detail view panels, only the location PCF files get deployed.



Lesson outline

- Detail view panel fundamentals
- Creating detail view panels
- Reference detail view panels

Referencing detail view panels



- A reference widget in a parent container references a PCF File as an embedded child container
- A PanelRef widget can reference:
 - Detail View Panel
 - List View Panel
 - Card View Panel
 - List Detail Panel

© Guidewire Software, Inc. 2001-2014. All rights reserved. Do not distribute without permission.

24

G U I D E W I R E

Detail view panels can be referenced by three types of containers: screens, card view panels and list detail panels.

The methods for referencing a list view panel in a screen, card view panel or list detail panels is identical: you use a PanelRef widget.

Input Sets are covered in the "Input Sets" lesson. List views are covered in the "List Views" lessons.

Panel Ref

The screenshot shows a software application window with a title bar. Below the title bar is a toolbar with buttons for 'Edit' and 'Suggest Least Busy User'. The main area is titled 'Summary'. It contains two sections: 'Basic Information' and 'Flag Entries'. The 'Basic Information' section displays details like Name (William Andy), Public ID (ab:5), Created On (01/13/2014), and Assigned User (Bruce Baker). The 'Flag Entries' section is a table with columns for Flag ID, View, Date Flagged, Reason, and Date Unflagged. Two entries are listed: one from 01/22/2014 with reason 'No email address for this contact.' and another from 01/13/2014 with reason 'No email address for this contact.' and date unflagged as 01/16/2014. At the bottom right of the summary panel, there is a label 'Detail View Panel'.

- A **Panel Ref** includes a reference to a "panel" container
 - Card View Panel, Detail View Panel, List Detail Panel, List View Panel, or Panel Set
 - Optionally supplies referenced panel with Title, Toolbar, Help Text

© Guidewire Software, Inc. 2001-2014. All rights reserved. Do not distribute without permission.

25

G U I D E W I R E

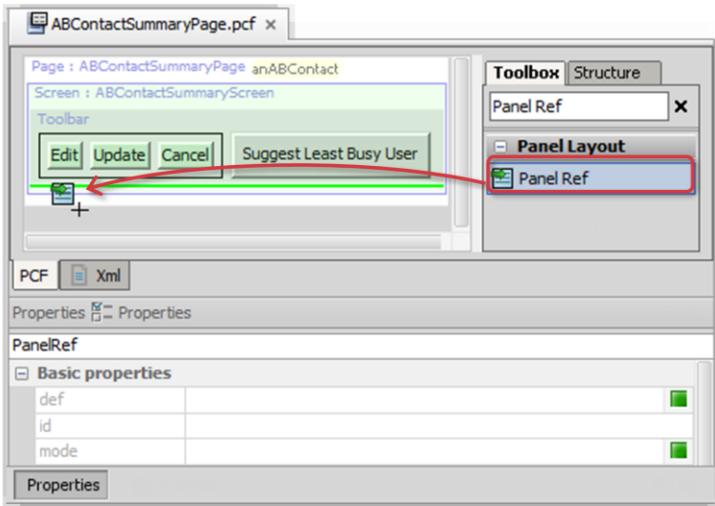
A Panel Ref requires a reference to a panel such as a Detail View Panel, List View Panel, Panel Set, or Card View Panel. A Panel Ref supplies the referenced panel with title, toolbar or instructional text.

Steps to reference a detail view panel

1. Add PanelRef widget
2. Specify widget properties
3. Deploy PCFs

It is not possible to add a reference to an inline detail view panel.

Step 1: Add panel ref



- Add a PanelRef widget to the parent container
- Panel Ref requires a def property value

- def property identifies the name of panel to reference and object to pass as root object

© Guidewire Software, Inc. 2001-2014. All rights reserved. Do not distribute without permission.

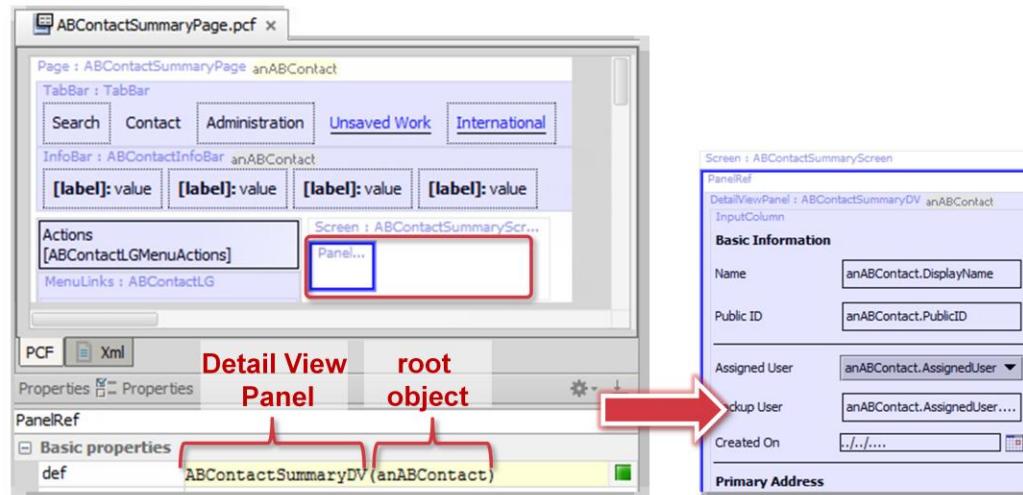
27

G U I D E W I R E

A PanelRef widget can reference a detail view panel. To reference a detail view panel from a parent container, add a Panel Ref in the appropriate place in the parent container.

In the slide example, the Panel Ref has already been placed in the Screen in the PCF editor canvas. The Properties window shows the Properties tab of the Panel Ref.

Step 3: Reference the detail view panel



- Define the def property
 - Specify the detail view panel PCF file
 - Pass the required object type as an argument

© Guidewire Software, Inc. 2001-2014. All rights reserved. Do not distribute without permission.

28

G U I D E W I R E

In the panel ref's def property, specify the detail view panel name. After the name, inside parentheses, specify the required object(s) to pass to the detail view panel.

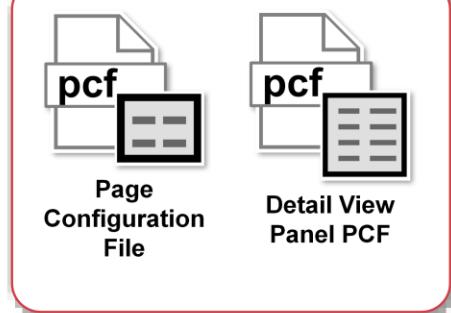
In the slide example, ABContactSummaryPage defines a root object named anABContact. ABContactSummaryPage contains a newly added Panel Ref. The Panel Ref requires a value for the def property. The def property references the Detail View Panel named ABContactSummaryDV. The def property passes the anABContact root object as an argument to ABContactSummaryDV.

The toolbar in the slide example has been removed in the screenshot.

Step 3: Deploy PCFs

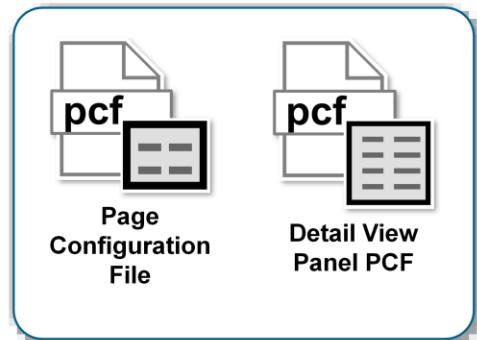
Restart Server

- PCFs read at server startup



Reload PCFs

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



© Guidewire Software, Inc. 2001-2014. All rights reserved. Do not distribute without permission.

29

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()`.

Detail view panel in read-only mode

Summary

← Toolbar and Edit Buttons

Basic Information

| | |
|---------------|--------------|
| Name | William Andy |
| Public ID | ab:5 |
| Created On | 01/13/2014 |
| Assigned User | Bruce Baker |

Flag Entries

| | View | Date Flagged | Reason | Date Unflagged |
|--|---------------------------|--------------|------------------------------------|----------------|
| | View/Edit | 01/22/2014 | No email address for this contact. | |
| | View | 01/13/2014 | No email address for this contact. | 01/16/2014 |

Primary Address

| | |
|--------------|--|
| Address | 345 Fir Lane La Canada 91352 Algeria |
| Address Type | Home |

- By default, there is no way to put detail view into edit mode
- Next lesson discusses how to add Edit|Update buttons so that data can be modified

Lesson objectives review

- You should now be able to:
 - Describe the functionality of detail view panels
 - Create a new detail view panel
 - Reference a detail view panel from a parent container

Review questions

1. A detail view panel that displays data must have at least one root object. Why?
2. A detail view panel that displays data must have at least one input column. Why?
3. What are the two most common types of widgets used to lay out a detail view panel so that it is more usable to end users?
4. What determines if a detail view panel is reusable or not?
5. The def property of a panel ref usually has a value in the format of "x(y)"
 - a) What information comes before the parenthesis? (What is the x?)
 - b) What information is defined within the parenthesis? (What is the y?)

Answers

- 1) The input widgets of a detail view panel must reference some object in order to specify where the data they display comes from. This object must be the root object, or an object related to the root object. Therefore, in order to display data, a root object is needed.
- 2) Input columns are used to organize atomic widgets in a detail view panel. Every input widget must be in an input column. If a detail view panel displays data, then it has at least one input widget, which means it must have at least one input column.
- 3) Label and input divider.
- 4) If a detail view panel is declared as the parent object of a PCF file, then it is reusable. It can be referenced by any number of other containers.
- 5a) The x is the name of the container to reference.
- 5b) The y is a list of objects to pass to the child container to use as root objects.

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.