**Page Configuration Format:**

* Used to create a user interface of all guidewire applications.
* A PCF is an XSD validated XML document containing elements that describe the structure, layout and behavior of the web user interface.
* Named with the extension .pcf .
* PCF elements are defined within tag <PCF>.

**Widgets:**

* A widget is the PCF element that is converted into HTML and displayed.
* Widgets can specify permissions to view or edit data for the logged in user.
* Displayable elements that are rendered into HTML.

**1. Atomic widgets:**

* Smallest building blocks of the user interface.
* Atomic widgets are individual field items such as inputs, cells or buttons.
* They are always defined within the container widgets or locations.
* Executes individual actions.

**2. Container widgets:**

* Collection of atomic widgets and other container widgets.
* Organize data and functionality into logical groups.
* Screen, list detail panel, card, detail view, list view, input set.

1. **Primary container:**

A primary container is a reusable view that organizes atomic widgets.

* **list view**-organize information in a tabular format.
* **detail view**-series of data fields laid out in one or more columns.
* **input set**-set of atomic widgets organized in logical groups, allowing them to be reused.

1. **Secondary container:**

Secondary containers organize primary containers.

* **List detail panel** -list view panel is the top view and bottom view display details about the selected list item from the top panel.
* **Card view** -collection of cards, with each card having detail and list view.

1. **Top-level containers:**

A top-level container organizes secondary and primary containers.

* **Screen –** Screen are the top-level container that are used to connect what is displayed in the user interface.A Screen can directly contain both primary and secondary containers, but it cannot directly contain most atomic widgets.

**Detail view:**

A detail view is a panel that is composed of a series of data fields laid out in one or more columns. It can contain information about a single data object, or it can include data from multiple related objects. Any input widget can appear within a detail view.

**List View:**

A list view is a panel that displays rows of data in a two-dimensional table. The data can be an array of entities, results of a database query, reference table rows, or any other data that can be represented in tabular form.

**List-Detail View:**

* List detail panels contain a [table(list view)](https://docs.guidewire.com/cloud/cc/202411/config/config/topics/ui-design-guide/c_tables.html#c_tables) in the top panel and a [tab](https://docs.guidewire.com/cloud/cc/202411/config/config/topics/ui-design-guide/c_tabs.html#c_tabs) view in the bottom panel. The tab view contents can depend on what is selected in the table.
* Use list detail panels when users need easy and simultaneous access to data summaries and details on the same screen.

**Card View:**

A Card View is a visual UI component that displays information in compact, rectangular "cards" (like digital index cards), instead of traditional tables or forms. Each card represents a single record (e.g., a claim, contact, or document) with key details.

**Locations:**

The location is a PCF element that a user can navigate to.

Locations are used primarily to provide a hierarchical organization of the interface elements and to assist with navigation.

* **Page** - Page is a location. It contains a single screen. Used exclusively with location groups.
* **Location group** - is a collection of pages with each having its own screen. All pages share common info bar, tab bar, side bar, action menu. provide the structure and navigation for a group of related pages.
* **Wizard**- is the collection of screens used. It has multiple screens but only one screen is displayed at a time. It has single info bar, side bar and action menus. Includes tool bar with back and next buttons.
* **Worksheet (entry point):** contains a single screen rendered in the workspace frame. It is the one area of the user interface that is not always visible. It is visible only when a screen is displayed in it.
* **Popup (entry point):**

A popup contains a single screen and returns the user to the previous location once the pop is closed.

* **Exit Point:**

External links can be applied. It will take you out of the guidewire screen.

* **Forward (entry point):**

It will execute the logic before navigating another location.

**Entry point:**

The entry Point is a PCF property that tells Guidewire which screen (PCF file) should be used as the starting point or entry page for a particular functionality or navigation flow in the UI.

Policy\_ExtPopup(aPolicy: Claim)

**Variable:**

A variable is used to define a parameter or context object (like an entity or a value) that the page or popup can access. **name** of the variable which we are going to use (a Claim entity used as a parameter aPolicy) and **type- which entity we are going to use** need to given

**Required Variable:**

Used to **enforce that specific variables must be passed** to this PCF file — otherwise, Guidewire will throw a runtime error. This screen **cannot run** unless you pass me these variables.

**Editable:** requires Boolean expressions

**Label:** returns a string expression

**Value:** return an object expression

**Post on change:**

The post on change functionality enables users to immediately see the impact of a field value change on related fields on the same screen.

**Example,**

When a user creates a vehicle incident in claim center, the collision indicator field has no initial value. When the value is changed, the changes are posted to the server. The server evaluates the remaining fields on the screen and identifies a group of fields that test the collision indicator for a value of true in their visible property. When this condition is satisfied, the additional fields are displayed.

1. differUpdate: Preserves legacy behavior that may be desired after an upgrade. Do not use for new post on change usages.
2. disablePostOnEnter: If evaluated true when page is rendered, this field will not trigger Post On change. Default is false.
3. Onchange: Defines a Gosu expression to invoke when user changes the value of the widget. Cause immediate post back to the server.

**What is atomic and container widgets?**

**Atomic widgets:**

Atomic Widgets are the basic building blocks of the UI. It used to display and interact with individual pieces of data. Atomic widgets are used to create individual fields and controls on a page. Examples., Text input, label, check box etc.

**Container widgets:**

Container Widgets are used to group atomic widgets together and provide the UI components and controlling their appearance and behavior. They help in layout management and organizing multiple fields.

Examples., Row, Column, Detail view etc.

**How available and visible works in PCF?**

**Available:**

* The Available property determines whether a field or widget is enabled and can be interacted with by the user.
* **Behavior:**
  + If **Available is set to true**, the field or widget is enabled, and the user can input data or interact with it.
  + If **Available is set to false**, the field or widget is disabled, and the user cannot interact with it. It may appear grayed out or inactive.

**Example**:

* A field like **Claimant Name** might be disabled (Available = false) if the claim is in a "Closed" status.

**Visible:**

* The Visible property determines whether a field or widget is displayed on the screen.
* **Behavior:**
  + If **Visible is set to true**, the field or widget is shown to the user.
  + If **Visible is set to false**, the field or widget is hidden from the user.

**Example**:

* A field like **Injury Description** might be hidden (Visible = false) if the claim type is "Property Damage" instead of "Personal Injury."