



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 how modes are used to create different versions of a single container
 - Implement modal PCFs

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

- Mode fundamentals
- Implement modal PCFs

Widgets that need to vary

Details

Person Info | Phone & Addresses | Bank Accounts | Analysis

Name

Full Name	William Andy
Prefix	
First Name	William
Middle Name	
Last Name	Andy
Suffix	

Employment Info

Occupation	
Employer	

if contact is a person

License Info

Driver's License	1213145
State	Alaska

if contact is an adjudicator

License Info

Adjudicator License	21-11-14141313
Domain	Federal

if contact is an attorney

License Info

Law License	FH1-HG2-113924
Specialty	General Liability

if contact is a doctor

Medical Specialty

Category	Surgery
Specialty	Hospitalist

- In some cases, widgets need to vary significantly based on business scenario

Addressing all scenarios within one PCF

Details

[Edit](#)

Person Info Phone & Addresses Bank Accounts Analysis

Name

Full Name Rebecca Stevens

Prefix

First Name Rebecca

Middle Name

Last Name Stevens

Suffix

Tax Info

Tax ID *****

Tax Filing Status

Date of Birth

Gender

Marital Status

Employment Info

Occupation

Employer

License Info

Driver's License 1213145

State Alaska

Adjudicator License 21-11-14141313

Domain Federal

Law License FH1-HG2-113924

Specialty General Liability

Medical Specialty

Category Surgery

Specialty Hospitalist

if contact is attorney, adjudicator, or person

if contact is person

if contact is adjudicator

if contact is attorney

if contact is doctor

- All fields in one container makes container difficult to maintain

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

5

G U I D E W I R E

For user interfaces that need different structures, you could include all the possible structures in a single PCF file. Including all the possible variants in a single PCF file adds to the complexity of the PCF and eliminates the possibility of reusable containers. In addition, much of the conditional logic becomes overly repetitive. Developers of applications prefer working with smaller components: they are easy to create, debug, modify, and maintain.

Addressing scenarios with multiple PCFs

Details	
Edit	
Person Info	Phone & Addresses Bank Accounts Analysis
Name	Employment Info
Full Name	William Andy
Prefix	
First Name	William
Middle Name	
Last Name	Andy
Suffix	
Occupation	
Employer	

Get InputSet for <contact's subtype>

InputSet for ABPerson

License Info	
Driver's License	1213145
State	Alaska

InputSet for ABAdjudicator

License Info	
Adjudicator License	21-11-14141313
Domain	Federal

InputSet for ABAttorney

License Info	
Law License	FH1-HG2-113924
Specialty	General Liability

InputSet for ABDirector

Medical Specialty	
Category	Surgery
Specialty	Hospitalist

- Sometimes, best approach is to use a series of separate PCFs, each for a specific scenario
 - The application can reference the PCF appropriate to a given scenario

When a single PCF cannot easily accommodate a variety of business cases, it is often easier to create a set of PCFs, one for each use case. Each version needs a value to identify the use case it is designed for. When the PCF is called, it is called by both its name and its "use case". Modes exist in the PCF architecture to implement this "use case" versioning.

Modes

- **Mode** is a property used for PCFs that are appropriate for a given business scenario
 - Mode identifies which scenario(s) the PCF is for
 - Modal PCFs always come in sets

SubtypeInfoInputSet

mode = ABPerson, ABPolicyPerson

License Info	
Driver's License	2WED458
State	Washington

SubtypeInfoInputSet

mode = ABAttorney

License Info	
Law License	JX3-AK7-309921
Specialty	Workers' Compensation

SubtypeInfoInputSet

mode = ABAdjudicator

License Info	
Adjudicator License	23-44-231223
Domain	Municipal

SubtypeInfoInputSet

mode = ABDDoctor

Medical Specialty	
Category	General Care
Specialty	Internal Medicine

There is no requirement for modal PCFs to come in sets. It is possible to create a single PCF with a mode that had no counterparts. This sort of PCF would behave in exactly the same way as a PCF that didn't use modes, which would mean there was no benefit to having made the PCF modal. Therefore, in practice, modal PCFs always come in sets of PCFs.

A modal PCF can have multiple modes. In the slide example, the first SubtypeInfoInputSet is used for contacts of subtype ABPerson or ABPolicyPerson.

Default mode

- Modal PCF sets include one PCF with mode of "default", which is used when something references a mode that doesn't otherwise exist

SubtypeInfoInputSet
mode = default

SubtypeInfoInputSet
mode = ABPerson, ABPolicyPerson

License Info	
Driver's License	1213145
State	Alaska

SubtypeInfoInputSet
mode = ABAdjudicator

License Info	
Adjudicator License	21-11-14141313
Domain	Federal

SubtypeInfoInputSet
mode = ABAttorney

License Info	
Law License	FH1-HG2-113924
Specialty	General Liability

SubtypeInfoInputSet
mode = ABDDoctor

Medical Specialty	
Category	Surgery
Specialty	Hospitalist

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

8

G U I D E W I R E

In the slide example, the default PCF is blank. If a SubtypeInfoInputSet is referenced but the mode it references cannot be found (such as, for example, a reference to SubtypeInfoInputSet with the mode ABPropertyInspector), then the default PCF is used.

In some situations, the "default" mode is assigned to a PCF that has other non-default modes. For example, since all person contacts are either of type ABPerson or one of its children, it would be possible to have assigned ABPerson, ABPolicyPerson, and default to the first SubtypeInfoInputSet.

Common uses of modal PCFs

- Modes can be used with:
 - Detail view panels
 - Card view panels
 - Input sets
 - List view panels
 - Info bars
 - Screens
 - Menu item sets
 - ...and more (refer to documentation)
- Modes are often used to accommodate variations in:
 - Contact type (individuals, companies, and so on)
 - Line of business (auto, workers' comp, property, and so on)

In PolicyCenter, modes are also used to accommodate variations in policy transaction type (submission, change, renewal, cancelation, and so on).



Lesson outline

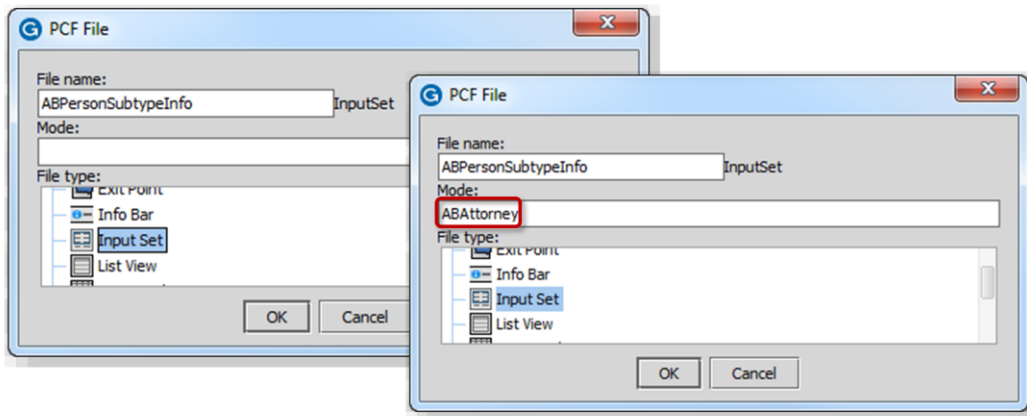
- Mode basics
- Implement modal PCFs

Steps to implement modal PCFs

1. Create modal PCF set
 - a) For each PCF, specify its mode(s)
 - b) Ensure there is one PCF with "default" mode
 - c) Ensure each PCF has identical required variables (in number, type, and order)
2. Create reference to modal PCF set
3. Deploy PCFs

Step 1a: Create modal PCF set

- Every PCF must have:
 - Same ID
 - Same number, type, and order of required variables
 - One or more unique modes (runtime error occurs if referenced mode belongs to multiple PCFs in same set)



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

12

G U I D E W I R E

Except for the mode of "default", mode values are not case-sensitive. Guidewire recommends that mode values use the same casing as the values to be passed to the reference tag, however.

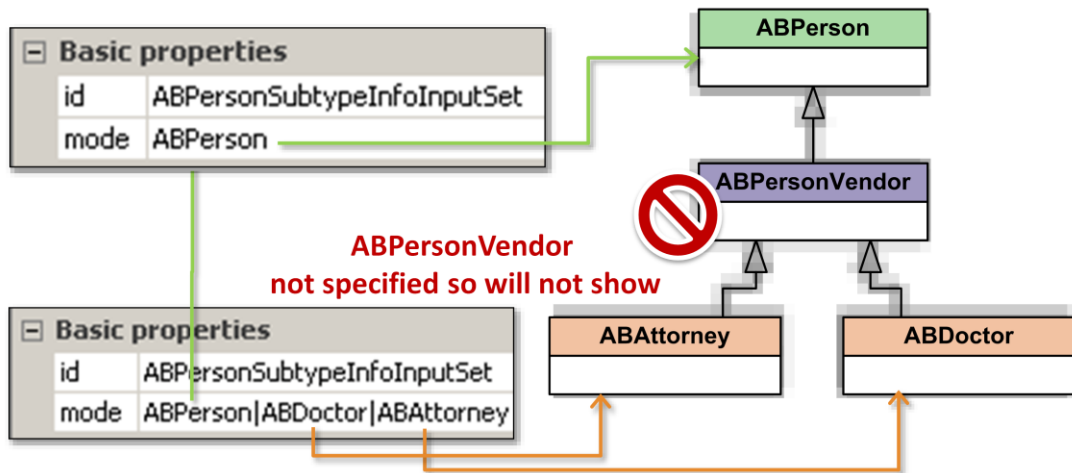
If a PCF has more than one mode, then each mode should be listed in the mode property as a pipe-delimited list (such as "ABPerson|ABPolicyPerson").

Every mode must uniquely belong to one and only one PCF in the set. If a mode is listed with two or more PCFs, there will be a runtime error when that mode is referenced because the application will be unable to determine which PCF to use.

After you have created a modal PCF, you cannot modify the mode through the mode property of the PCF. This is because the file name includes the mode or modes, and the two values must remain in sync. If you need to change the mode of a modal PCF (for example, to add an additional mode to it), right-click the PCF in the resources tree and select "Change mode". This allows you change the mode, which is then reflected in both the mode property and the file name.

Modes and subtypes

- Mode property recognizes explicitly specified subtypes
 - Not aware of a subtype's hierarchy
 - Unspecified subtypes will not render
 - Use pipe delimiter (|) to specify more than one subtype



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

13

G U I D E W I R E

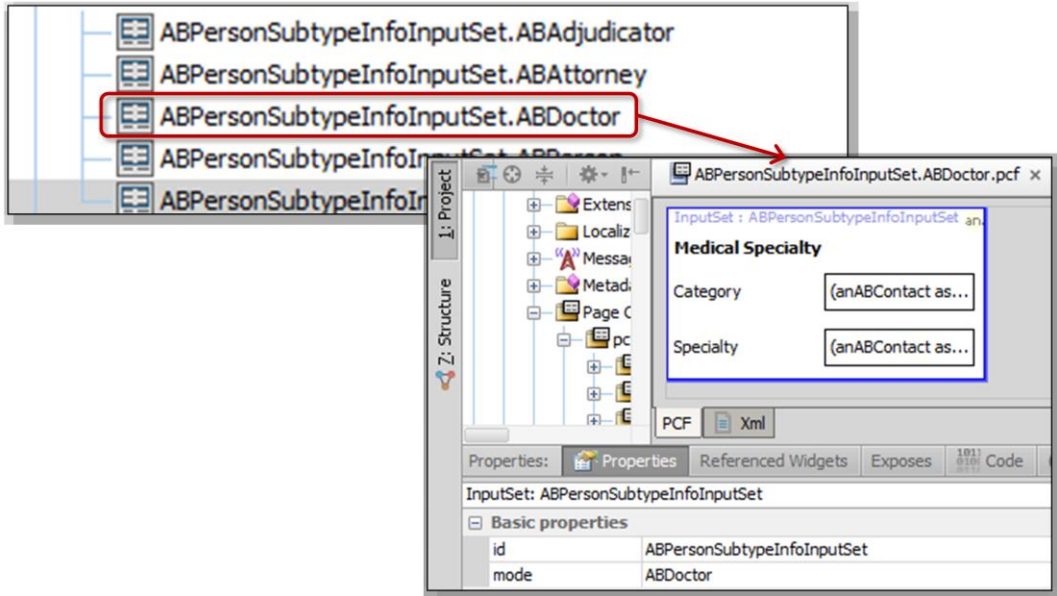
In the slide example, the first modal PCF has only a single mode, ABPerson. It will be used in any circumstance where the mode value is "ABPerson", but it will not be used if the mode value is "ABAttorney" or "ABDoctor", even though they are subtypes of ABPerson.

The second modal PCF has three modes, ABPerson, ABDoctor, and ABAttorney. This modal PCF will be used for any ABPerson object, whether it is ABPerson or one of its subtypes.

In the base application, there is actually an ABPersonVendor subtype in between the ABPerson level and the ABAttorney or ABDirector level. Because ABPersonVendor is not specified, the modal PCF will not be displayed.

Modal widgets in Studio

- In Studio, modal PCFs are listed as "<ID>.<mode>"



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

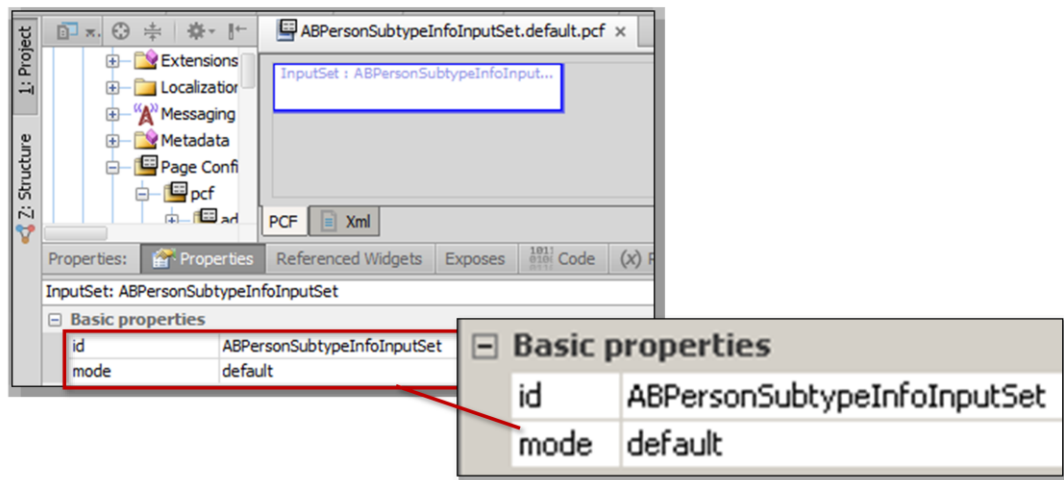
14

G U I D E W I R E

If a PCF has multiple modes, it is listed as "<ID>.<mode1>|<mode2>..."

Step 1b: Create default PCF

- One PCF should have "default" as (one of) its mode(s)
 - "default" must be lower case
 - Runtime error occurs if unknown mode is called and no "default" widget exists



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

15

G U I D E W I R E

Aside from the shared IDs, modes, and required variables, there is no configuration difference between a modal PCF and a non-modal PCF.

In order for the default widget to be used by default, the mode must be "default" in lower case. If it is not specified this way, then Guidewire treats the modal widget set as if it has no default widget. If the application encounters an unknown mode (and cannot find the default PCF because of a case-mismatch), then the user will see an error similar to this:

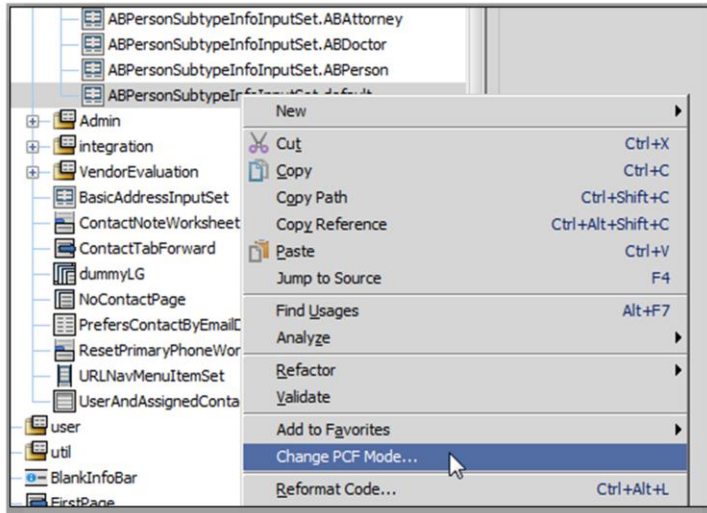
GWConfigurationException: <CONTAINER>/(InputColumnWidget)/(InputSetRefWidget) is looking for a non-existent widget with id="..." and mode="..."

Step 1c: Ensure same required variables

- Ensure each modal PCF has identical required variables
 - Variable definition (type)
 - Number of variables
 - Order of variables

The image displays two side-by-side screenshots of the Guidewire software interface, specifically the PCF (Person Contact Form) configuration for 'ABPersonSubtypeInfoInputSet.ABPerson.pcf' and 'ABPersonSubtypeInfoInputSet.ABAttorney.pcf'. Both forms have a 'License Info' section with input fields for 'Driver's License', 'State', 'Law License', and 'Specialty'. The 'Required Variables' section at the bottom of each form is highlighted with a red box, showing a list of variables: 'anABContact', 'name*', and 'type*'. The 'name*' variable is defined as 'anABContact' and the 'type*' variable is defined as 'ABContact'.

Changing modes



- Context menu -- > Change PCF mode...
 - Changes mode property
 - Updates file name

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

17

G U I D E W I R E

If you need to change a PCF's mode (for example, to add an additional mode to it), right-click the file name in the resources tree and select "Change PCF mode...".

This is not an issue for PCFs created by customers because those files exist only in the configuration directory. There is no need to keep these file names fixed.

Step 2: Create reference to PCF set

The screenshot shows the 'ABContactDetailsPersonDV.pcf' file in the Guidewire software. The main panel displays various input fields for a contact details form. The Properties panel at the bottom shows the 'InputSetRef' section with a table of properties. A red box highlights the 'mode' property, which is set to 'anABContact.Subtype'. A red arrow points from this property to the 'Shared section mode' dropdown in the 'License Info' section of the main panel.

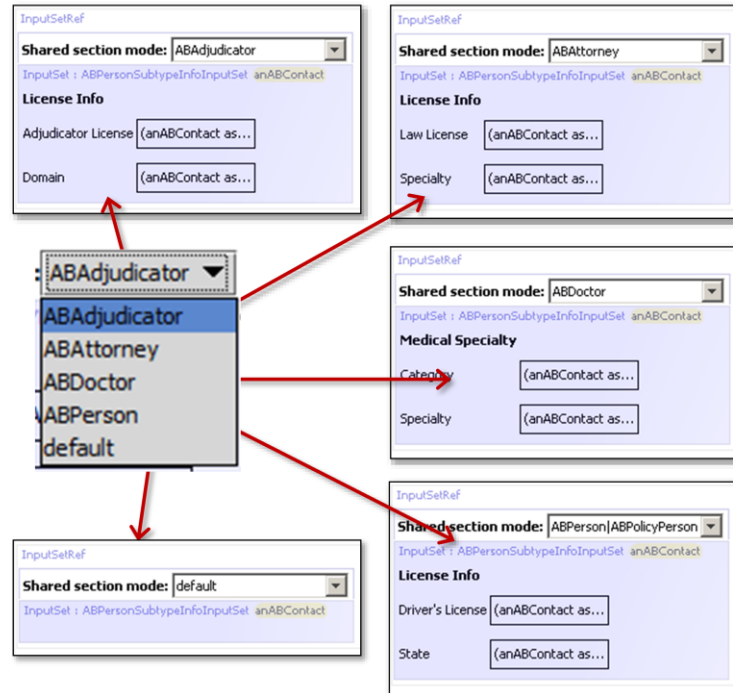
def*	ABPersonSubtypeInfoInputSet(anABContact)
id	
mode	anABContact.Subtype

- Reference must specify
 - PCF set to use
 - Mode within set to use

The value to pass for mode is usually a field of one of the base objects available to the parent container.

Modal references in Studio

- PCF Editor shows modal as dropdown that lists every mode
- The PCF Editor displays currently selected mode in the canvas



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

19

G U I D E W I R E

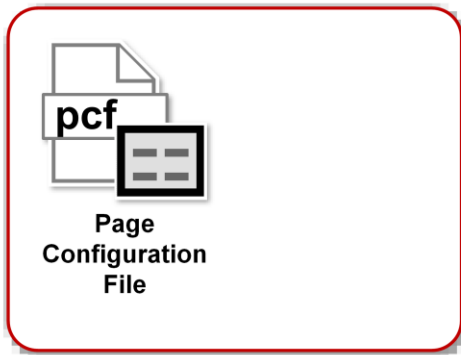
When you open a file with a modal reference, Studio displays the embedded modal PCF for the first mode when listed alphabetically.

Double-clicking the dark-blue area opens the currently selected modal widget.

Step 3: Deploy PCFs

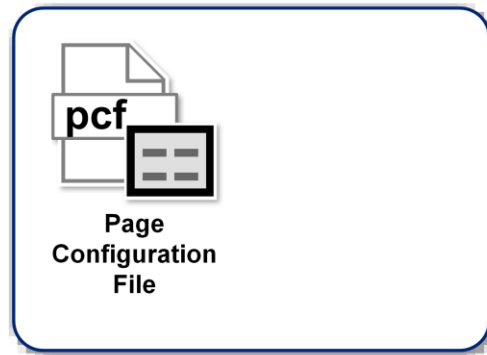
Restart Server

- PCFs read at server startup



Reload PCFs

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



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

20

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:

- Describe how modes are used to create different versions of a single container
- Implement modal PCFs

Review questions

1. Name two business situations in which it might be useful to use modal PCFs.
2. For a given set of modal PCFs, what two things must all of the PCFs have in common?
3. Can a PCF have more than one mode?
4. Can the same mode be used by more than one PCF within the same modal set?
5. What happens if a reference identifies a mode, but there is no PCF with that mode in the set?

Answers

- 1) Possible answers: Different types of contacts (individual vs. business); different lines of business (auto vs. property).
- 2) All of the files must have the same required variables (identical in number, order, and type), as well as the same ID.
- 3) Yes.
- 4) No. If the same mode is used in multiple PCFs in the same set, then the application cannot determine which file to use and a runtime error occurs.
- 5) If a PCF with a mode of "default" exists, then that default is used. Otherwise, a runtime error occurs.

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.