



# Introduction to Guidewire Configuration



May 1, 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 product architecture for Guidewire products
  - Recall the primary components used to configure Guidewire products
  - Identify the relationship between the Guidewire platform and the Guidewire applications
  - Explain the basic functionality of TrainingApp
  - Start a development instance of a Guidewire application
  - Describe the purpose of Guidewire Studio

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

- Guidewire product architecture
- Guidewire configuration technology
- The Guidewire platform
- TrainingApp
- Starting Guidewire applications
- Guidewire Studio

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

3

G U I D E W I R E

# Guidewire 8.0 product landscape

- Insurance carriers worldwide use Guidewire core systems as operational systems of record



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

4

GUIDEWIRE

Guidewire builds software products that help property and casualty (P&C) insurers replace their legacy core systems and transform their business. Guidewire provides core systems and data management/business intelligence tools used by insurers as operational systems of record. Additional products provide support for anytime/anywhere access and guidance and monitoring.

**Operational Systems of Record:** In the area of Core Operations Support, Guidewire InsuranceSuite provides a complete set of applications to support a carrier's core operations—underwriting, policy administration, billing, and claim management. The slide provides a high-level view of the core Guidewire products. It is not an exhaustive list of all applications and licensed features. Therefore, some features and products, such as ContactManager, Client Data Management, PolicyCenter Rating, PolicyCenter Reinsurance, and the ISO Standards-Based templates, are not shown.

In the area of Data Management and Business Intelligence, Guidewire DataHub is an operational data store that unifies, standardizes, and stores data from a carrier's own systems as well as external sources. Guidewire InfoCenter is a business intelligence warehouse that is purpose-built for P&C insurance. It provides easy-to-use reporting formats for business intelligence, analysis, and enhanced decision making. Using InfoCenter, carriers gain operational insight across the enterprise.

**Monitoring and Guidance:** Guidewire Live is a network that connects peer insurers, core systems data, external sources of information (such as weather services), and expert tools. Users access Guidewire Live through applications hosted by Guidewire and designed for specific challenges faced by P&C insurance professionals.

**External Access: Mobile & Portals** is a new set of applications that feature add-on products to the Guidewire core suite, providing self-service transactions through various mobile channels.

# Guidewire InsuranceSuite



**InsuranceSuite**



PolicyCenter

Create, modify, renew,  
and cancel policies



BillingCenter

Bill policy holders for  
policies, and pay  
commissions for policies  
to producers



ClaimCenter

Process claims to provide  
payments to claimants  
when covered losses  
occur

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

5

GUIDEWIRE

Guidewire InsuranceSuite 8.0 is a set of three core products that meet the core needs of property and casualty insurance carriers. Guidewire offers a set of applications designed to help property and casualty insurance companies expedite their policies, billing, and claim requirements:

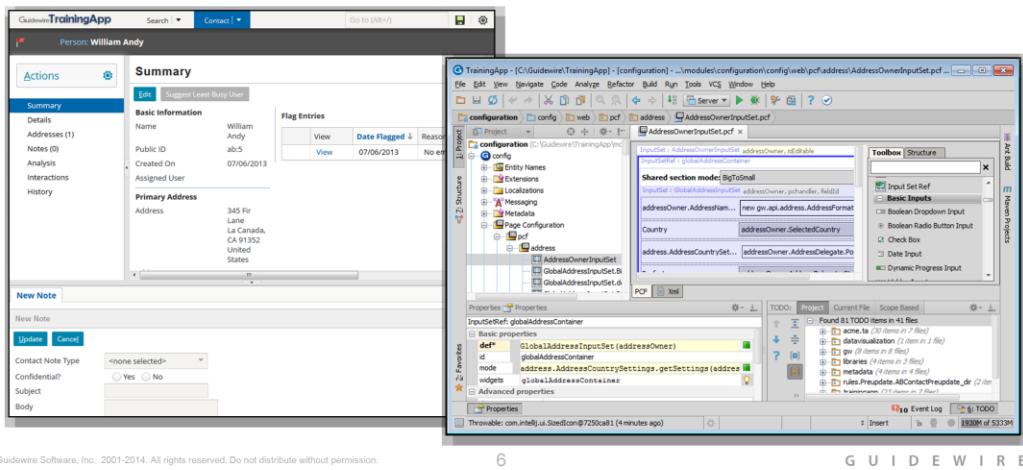
- ClaimCenter is an application designed to manage the process of reporting, verifying, and making payments on claims against a policy.
- PolicyCenter is an application designed to issue, modify, and maintain data about policies.
- BillingCenter is an application designed to issue and track the premium payments for policies and associated account charges.

Each application can function by itself, and each can be integrated with the other applications. PolicyCenter can be integrated with BillingCenter, so that the two applications can share information about the payment expected, the payment schedule, and the possible cancellation of a policy if payments are not received. PolicyCenter can be integrated with ClaimCenter, as ClaimCenter needs information about what was on the policy to determine if a loss is covered, and PolicyCenter needs to know what claims were filed to determine if the policy should be renewed and at what rate. Although less common, BillingCenter can be integrated with ClaimCenter. This is typically done for business situations like subrogation in which the insured suffers a loss that a third party's insurance should pay for. ClaimCenter identifies that subrogation is expected, and BillingCenter manages the invoices sent to the third party's insurance provider.

Depending on the configuration, InsuranceSuite can contain other modules: ContactManage, Client DataManagement, PolicyCenterRating, PolicyCenter Reinsurance, and Standard-Based templates.

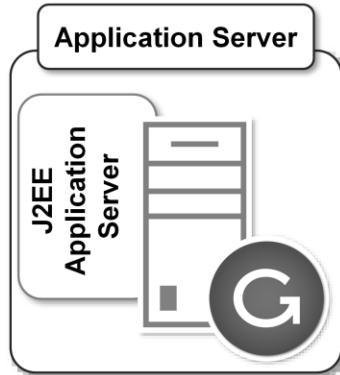
# Guidewire 8.0 platform

- The Guidewire 8.0 platform includes some significant improvements including:
  - Enterprise user interface (UI) framework
  - Full-featured, industry-standard, integrated development environment (IDE)



All lessons covering platform functionality have been developed on TrainingApp, the platform application developed by Guidewire Education to simplify platform functionality discussions by separating the functionality from the core applications.

# Guidewire application tier



- Contains the functional process and business logic
- Guidewire supports the following application servers:
  - WebSphere, WebLogic, Apache Tomcat, JBoss EAP
  - Jetty (for development only)

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

7

G U I D E W I R E

All Guidewire applications are run within an application server. An application server runs enterprise-wide applications. There are several different types of application servers. The type of application server used by Guidewire applications is a Java Enterprise Edition (Java EE) server. The platform was known as Java 2 Platform, Enterprise Edition or J2EE until the name was changed to Java EE in version 5.

Each Guidewire application is built and commonly deployed as an Web Application Archive (WAR) or Enterprise Application Archive (EAR) file to the application server. It contains all the configuration, operational data and data definition files necessary to execute the application. The application server is typically installed on a dedicated machine that does not host any other aspects of the architecture, such as the database.

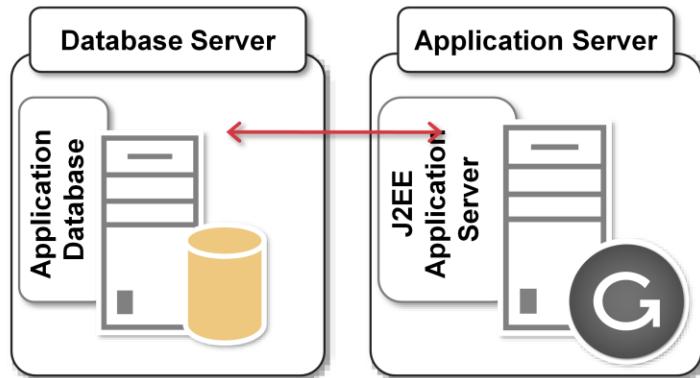
This machine must use an operating system supported by your third party application server vendor (or Guidewire, in the case of Apache Tomcat).

Guidewire supports the operating systems for which application servers provide customer support. For details, review the customer-viewable document Platform Matrix.

The Guidewire application (the center box) consists of a set of business rules, a user interface, a set of integration APIs, and a data model. The technology for configuring these application elements is platform-level (and common to all Guidewire applications), but the specific rules, UI, APIs, and data model for each application are distinct.

For the latest and most complete information about the software Guidewire requires and supports, refer to the Guidewire Platform Matrix page found at <https://guidewire.custhelp.com/app/resources/products/platform>.

## The data tier



- Data tier contains the business and operational database
- Guidewire supports the following RDBMs:
  - Oracle Enterprise
  - Microsoft SQL Server
  - H2 for development only

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

8

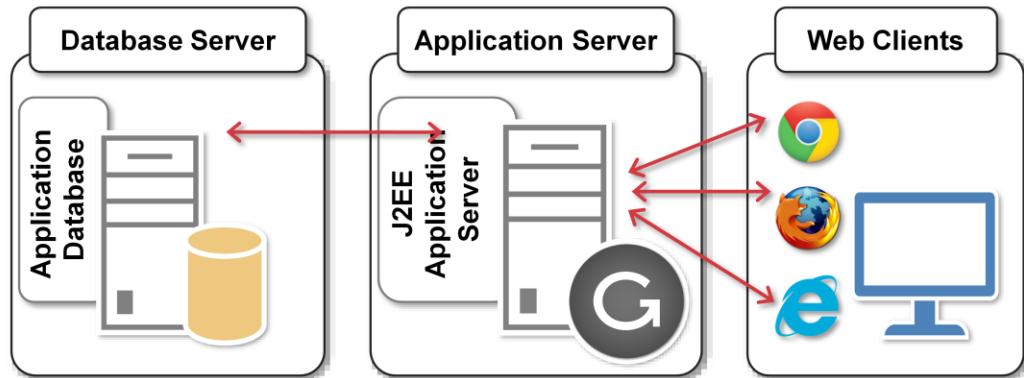
GUIDEWIRE

A relational database is a collection of data structures and objects that relate to each other. Databases store data in database tables. The operational data for the Guidewire application is stored inside a relational database. This database is typically hosted on a machine other than the machine hosting the Java EE application server.

The war file contains a file named config.xml that specifies which database to connect to and how to establish the connection.

For the latest and most complete information about the software Guidewire requires and supports, refer to the Guidewire Platform Matrix page found at <https://guidewire.custhelp.com/app/resources/products/platform>.

# The Presentation tier



- Presentation tier contains the user interface
- Modern web browser support the Guidewire user interface
  - Chrome 28+, Firefox 19+, and IE10+
  - See Guidewire Platform Matrix

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

9

G U I D E W I R E

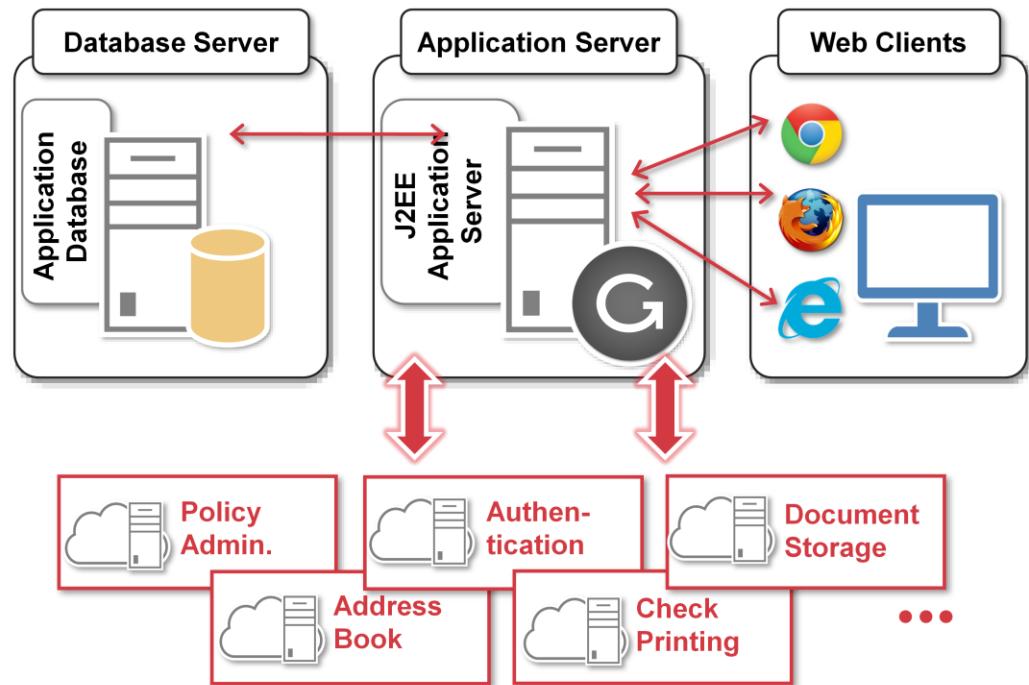
End-users connect to the application using a web browser. Chrome is the preferred browser. Each application generates a collection of standard HTML pages that are rendered by the browser. Each application is a server-side application. It dynamically generates the HTML pages and data within those pages. It makes only a minimal use of functionality in the client (the web browser). Because there are no static HTML files, you cannot use the browser's Back button. All navigation is controlled on the server side.

In a development environment, you deploy configuration changes in a variety of ways using Guidewire Studio and/or using the administrative tools in the application user interface. Configuration changes can cover several areas: data model, user interface, and Gosu code. In certain instances, you must stop and start the server. In other situations, you do not need to restart the server. This course will emphasize how and when to deploy new and changed resources.

In a production environment, configuration changes are typically made by rebuilding and redeploying the application's war file. This always involves stopping and restarting the application.

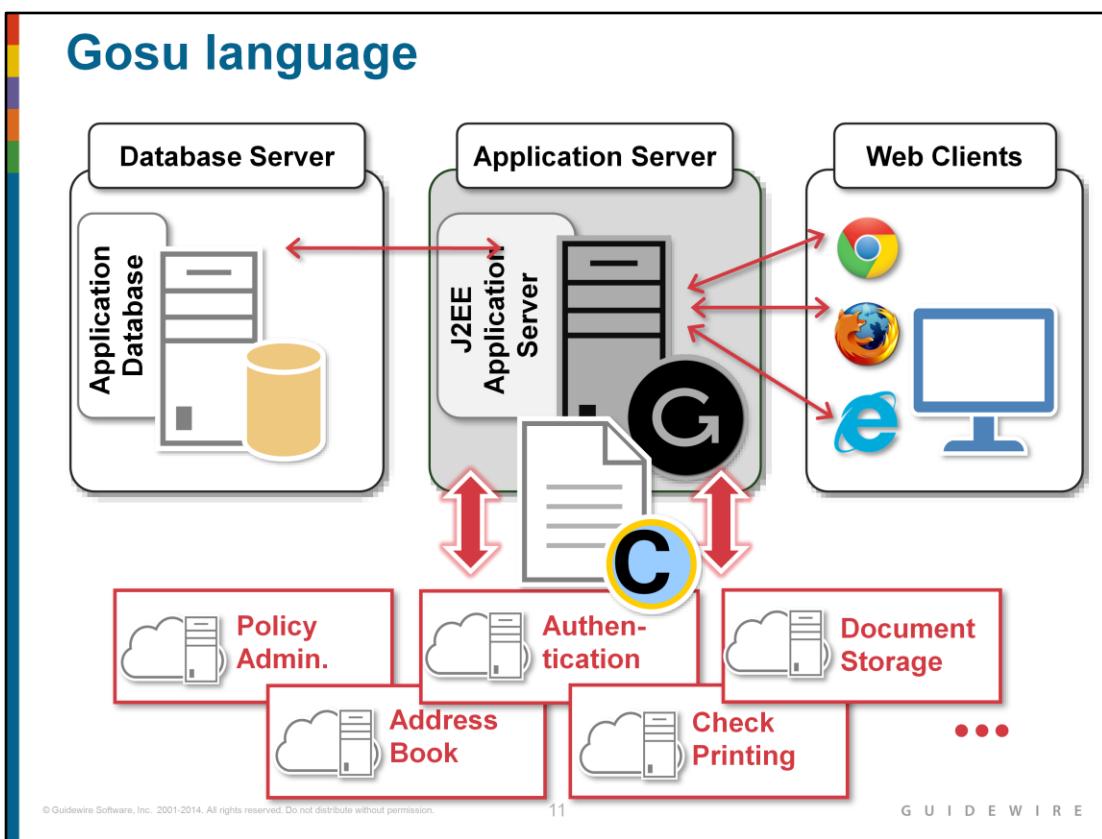
For the latest and most complete information about the software Guidewire requires and supports, refer to the Guidewire Platform Matrix page found at <https://guidewire.custhelp.com/app/resources/products/platform>.

## External systems



Each application is typically integrated with a number of external systems. The connections to these systems are configured through the application APIs.

# Gosu language



Gosu is an open-source, publicly-available programming language created by Guidewire. It is similar to Java. Gosu is used to specify runtime business logic that:

- Executes fundamental application behavior
- Manages complex business processes
- Specifies dynamic client-side behavior

Gosu can be used to affect behavior throughout the product architecture. This includes:

- Application behavior (such as specifying business rule behavior, methods, and entity enhancements)
- Client-side behavior in the web browser (such as reflecting the change of the value in one field in some other field)
- Data model behavior (such as specifying which fields to concatenate for the "display name" of a given object)
- Integration behavior (such as defining the behavior of a web service that responds to requests from external applications)

Guidewire developed Gosu for several reasons. First, there was a desire to have a single syntax that could be used to work with all of the elements relevant to Guidewire products (such as entities, display keys, classes, class enhancements, Java classes, permissions, and script parameters) even though these items have fundamentally distinct internal implementations.

There was no existing language that provided this ability. Second, there was a desire to have a code auto-complete feature in Guidewire Studio, the primary tool used to complete Guidewire configuration tasks. This is possible only with a statically-typed language. Most scripting languages, such as JavaScript, Perl, Python, and Ruby, are dynamically-typed.

## Lesson outline

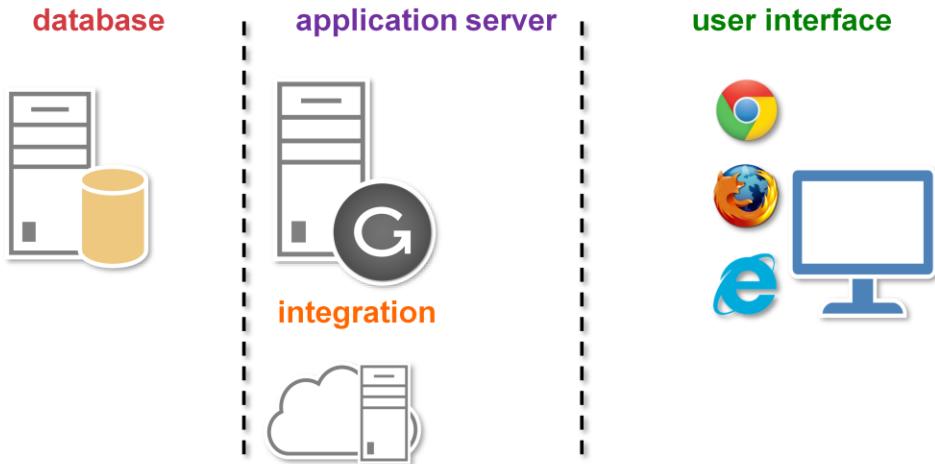
- Guidewire product architecture
- Guidewire configuration technology
- The Guidewire platform
- TrainingApp
- Starting Guidewire applications
- Guidewire Studio

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

12

G U I D E W I R E

# Guidewire configuration technology



- Guidewire has four core areas of configuration technology
  - One area of technology for each tier of the architecture
  - Fourth area of technology for the development of integration points to external systems

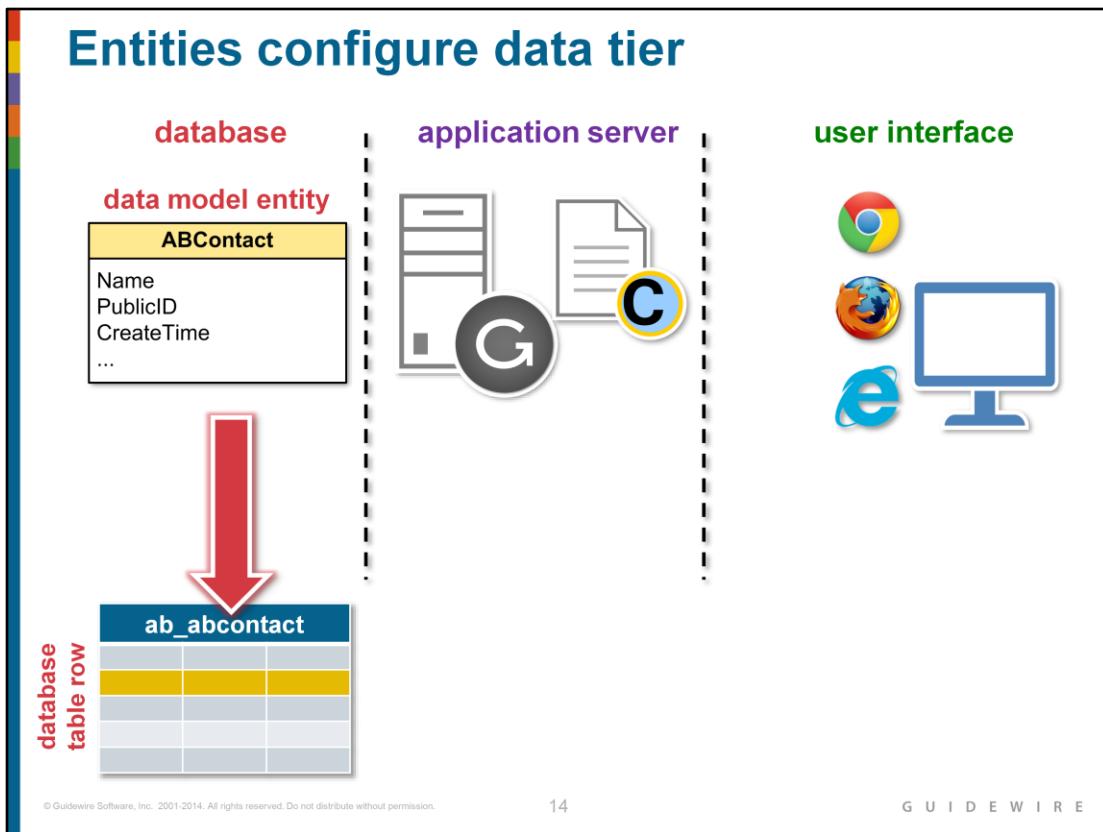
© Guidewire Software, Inc. 2001-2014. All rights reserved. Do not distribute without permission.

13

G U I D E W I R E

Three-tier architecture is a client-server architecture in which the functional process logic, data storage and user interface are developed and maintained as independent modules on separate platforms. Three-tier architecture is one of the industry-standard software architectures.

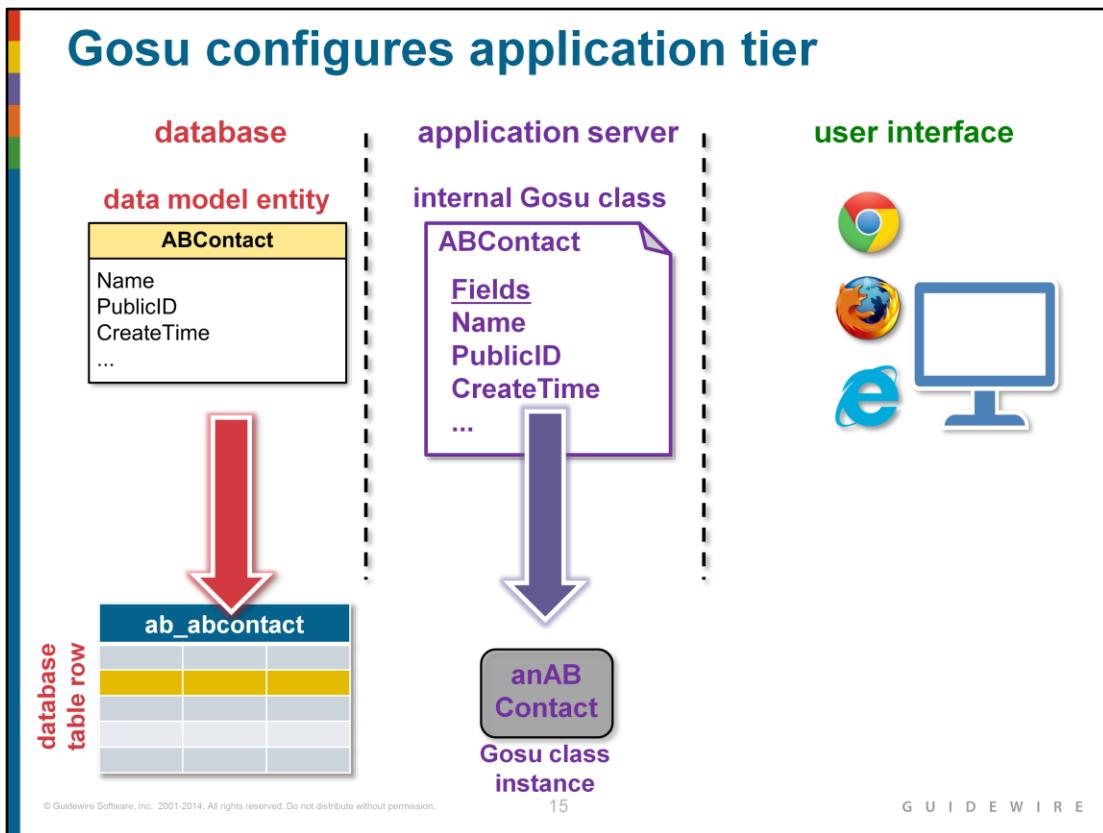
# Entities configure data tier



A data model entity is an abstract definition of a group of objects used by the data tier, such as Address Book contacts, or ABContacts. It defines the information about the objects that must be stored in the database, such as Name, PublicID, and CreateTime. It is defined in a set of one or more XML files.

In most cases, each data model entity corresponds to a table in the database. The data model entity definition defines the table structure. Each instance of the data model entity is stored as one row in the database table.

# Gosu configures application tier

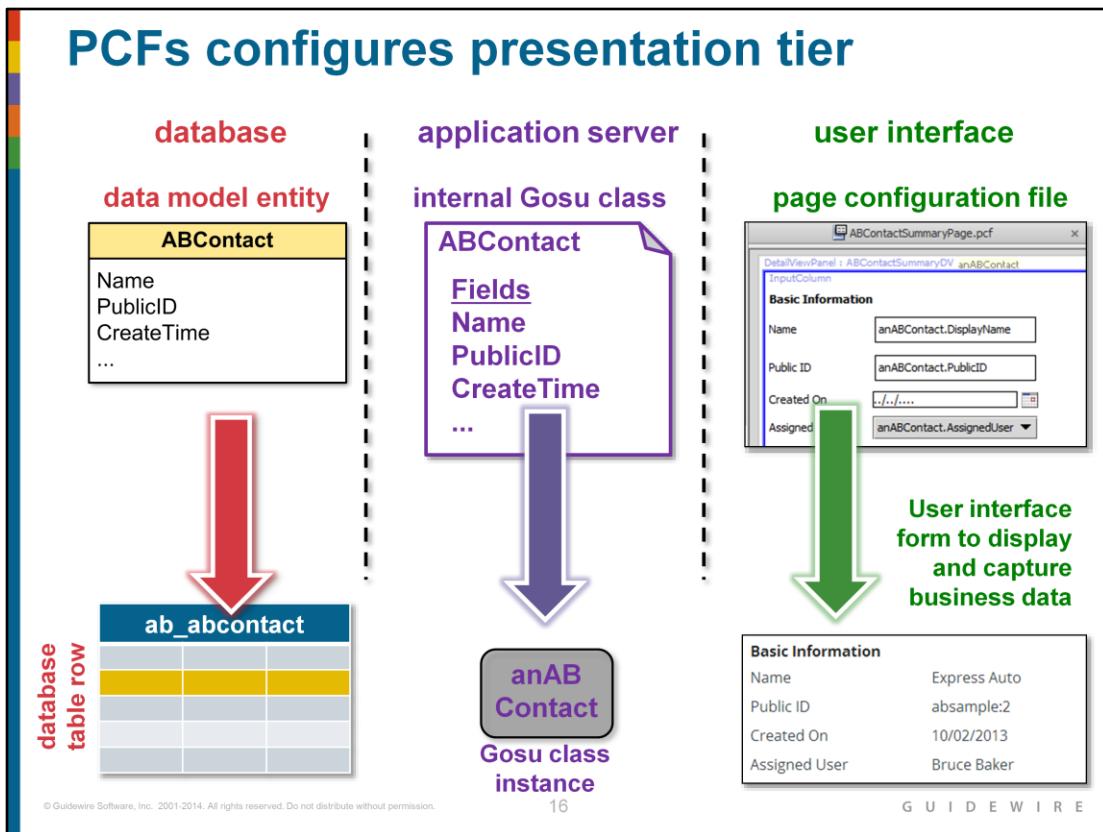


A Gosu class is an abstract definition of a group of objects used by the application tier, such as Address Book contacts, or ABContacts. It defines the information about the objects that must be maintained in the application server's run-time environment, such as Name, PublicID, and CreateTime. It is defined in a set of one or more Gosu files that have a proprietary format and a ".gs" extension.

For every data model entity, the Guidewire application automatically creates an internal Gosu class with the same name. For every field in the data model entity, there is a field in the corresponding internal Gosu class. For example, the ABContact data model entity has a "Name" field, and the internal ABContact Gosu class also has a "Name" field.

Whenever the application needs to work with an instance of a data model entity (which is stored as a row in the corresponding database table), the application creates an instance of the corresponding Gosu class. The information from the database is then read into that instance. For example, if a user searches for the ABContact whose name is "Express Auto", then the application finds the row in the database table for ABContact, creates an instance of the ABContact Gosu class, and reads the data from that row into that instance.

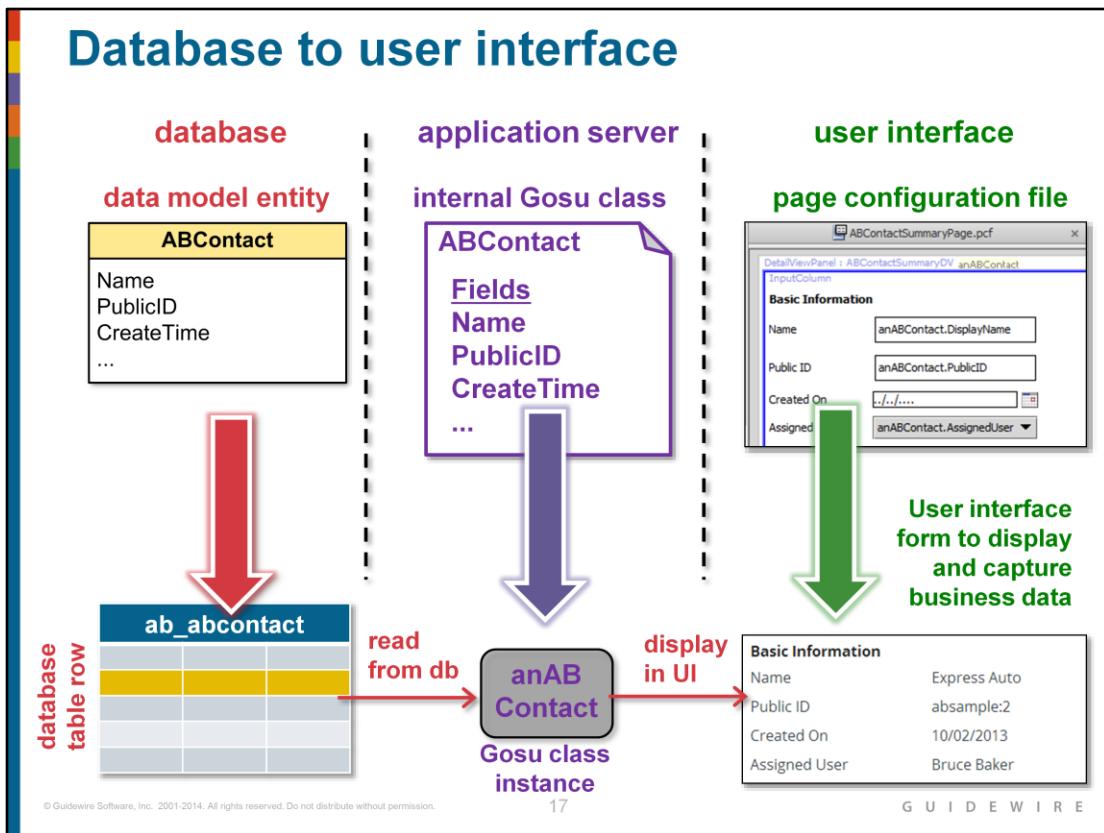
# PCFs configures presentation tier



A PCF (Page Configuration File) is an abstract definition of a form or location used by the user interface tier, such as an "ABContactSummaryDV" detail view used to display information about ABContacts. It defines the information about the objects to be displayed in the user interface, such as a Name field, a PublicID field, and a CreateTime field. It is defined in a set of one or more XML files using the proprietary PCF XML schema definition.

There is a strong correspondence between data model entities and internal Gosu classes. Every data model entity has one internal Gosu class. There isn't necessarily a strong correspondence between internal Gosu classes and PCFs. The data for one Gosu class could be displayed in a single PCF, or it could be displayed across multiple PCFs. The separation of the user interface tier and the application server tier gives developers the freedom to display data in whatever way makes sense to end users without being constrained by how that data is maintained in the application server or how the data is stored in the database.

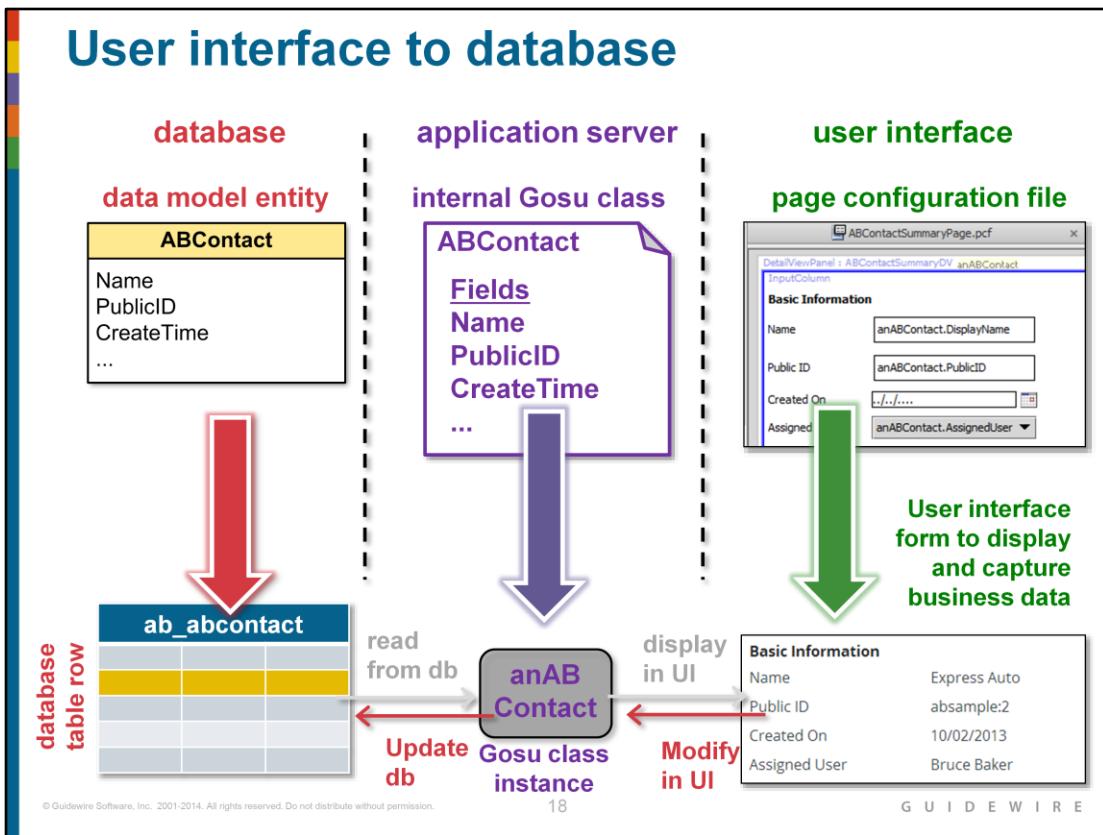
# Database to user interface



Whenever a user needs to work with an instance of a data model entity, such as the ABContact named "Express Auto":

- The application queries for the data row from the database table.
- The application creates an instance of the corresponding Gosu class and reads the information into that instance.
- The application sends the data in that instance to the web browser. The web browser displays the form as defined in the given PCF and populates the form with data from the instance.

# User interface to database



Whenever a user needs to save data, such as changing the name of the ABContact named "Express Auto":

- The information is modified in the user interface and then posted to the application server.
- The application server inserts data into the database table (if the object being saved in the UI is a new object) or updates data in the database table (if the object being saved in the UI is an existing object that has been modified).

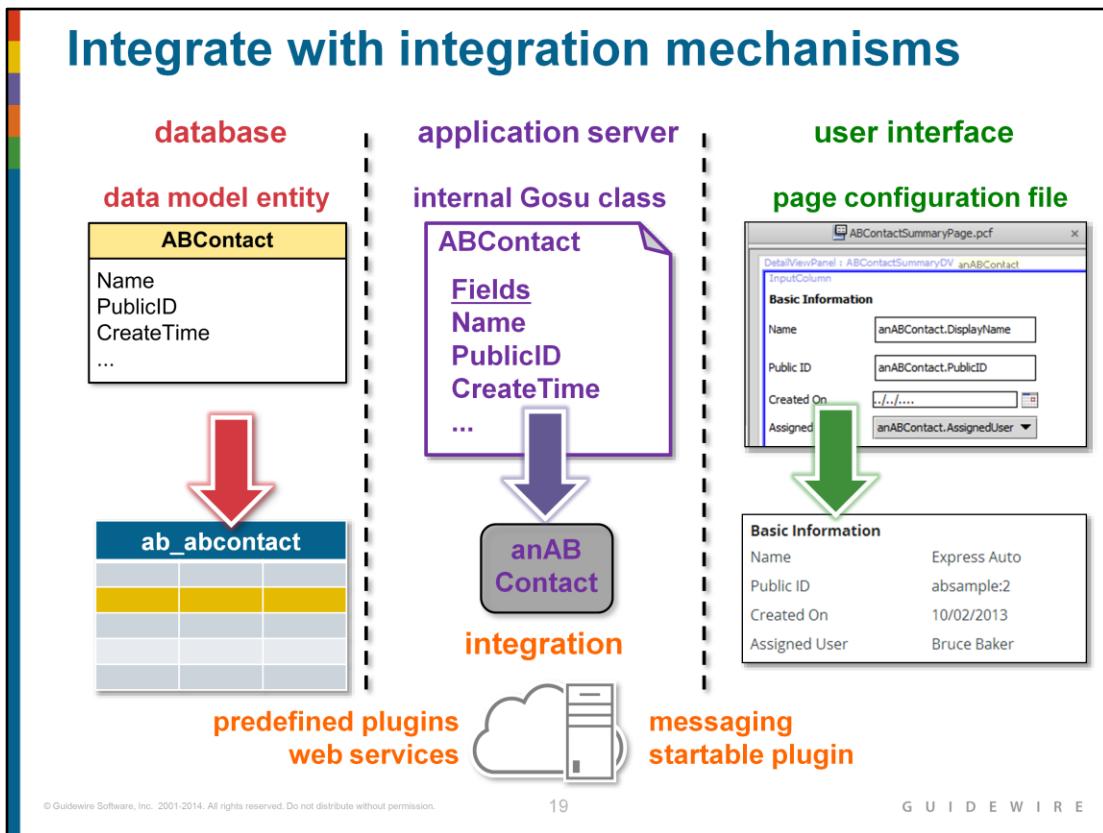
Be aware that there is variation in when data modifications made in the user interface get sent to the application server and/or the database.

In some situations, data in the user interface is ...

- not immediately sent to the application server (or saved to the database).
- sent to the application server but not immediately saved to the database.
- sent to the application server and also saved to the database at the same time.

These issues are discussed in more detail later in the course.

# Integrate with integration mechanisms



Guidewire applications share information with external systems using a variety of industry-standard technologies. This can include:

- Writing information to or reading information from an operating system file
- Writing information to or reading information from an external database
- Using remote procedure calls (such as web services) to share data with other applications directly
- Using message queues to request data from external systems asynchronously

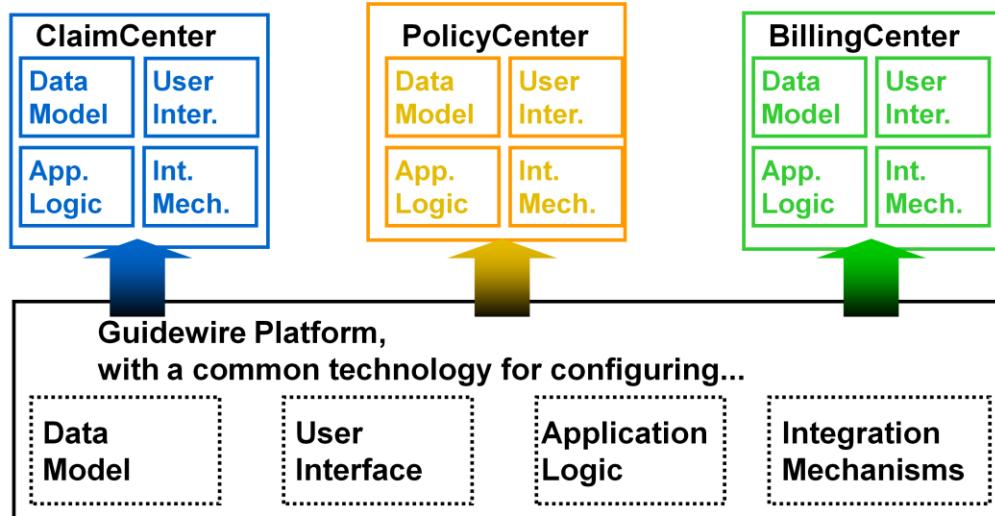
Guidewire applications have a set of integration mechanisms that make use of the industry-standard technologies listed above. This can include:

- Predefined plugins - A predefined plugin is a Gosu or Java class which implements a set of methods called by internal code and is related to fundamental application behavior. For example, the authentication plugin implements methods that define how authentication is executed against an external authentication system, such as an LDAP server.
- Web services - A Guidewire web service is written in Gosu. External applications can use it to make synchronous calls to Guidewire. For example, an ABContactAPI web service might allow external systems to retrieve information about a given contact, such as its phone number.
- Messaging - Messaging is a Guidewire integration mechanism in which messages are asynchronously sent to external systems in response to the creation or change of business data. For example, you may want to verify a newly created bank account exists with the external financial system. If this verification is not needed immediately, you could use messaging.
- Startable plugins - A startable plugin listens for requests from an external system that communicates with Guidewire via a mechanism such as JMS Messaging or TCP/IP. It is triggered by an incoming external request, and it typically processes the message asynchronously. For example, an external system may want to submit payments for a given ABContact. If the payments do not need to be processed immediately, you could process them with a startable plugin.

## Lesson outline

- Guidewire product architecture
- Guidewire configuration technology
- **The Guidewire platform**
- TrainingApp
- Starting Guidewire applications
- Guidewire Studio

# The Guidewire platform



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

21

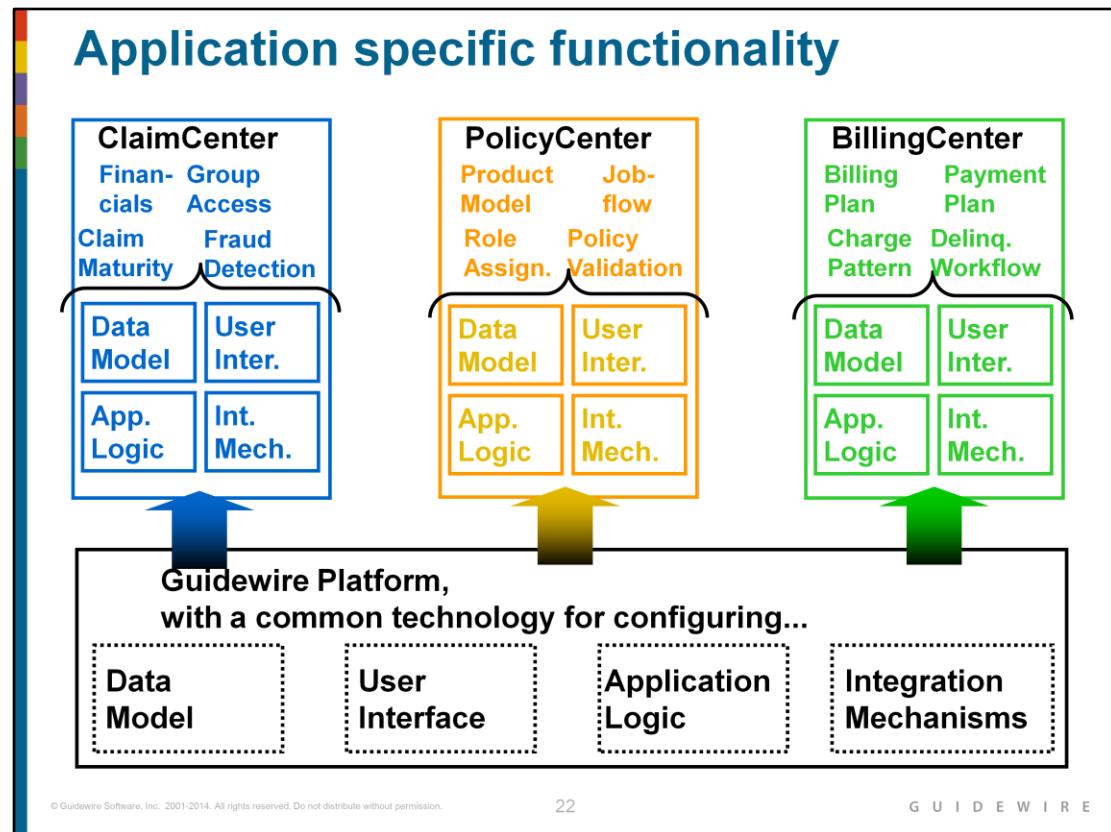
G U I D E W I R E

The Guidewire platform is a layer of configuration technology that includes the functionality needed to define a Guidewire application. For example, it includes:

- The technology to define a data model
- The technology to define a user interface
- The technology to define application logic
- The technology to define integration points

Every application uses this common technology to define its own data model, user interface, business logic, and integration points. Each application is distinct, but every application shares common abilities and configuration techniques with all the other applications.

# Application specific functionality



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

22

G U I D E W I R E

Each application also has application-specific functionality. For example:

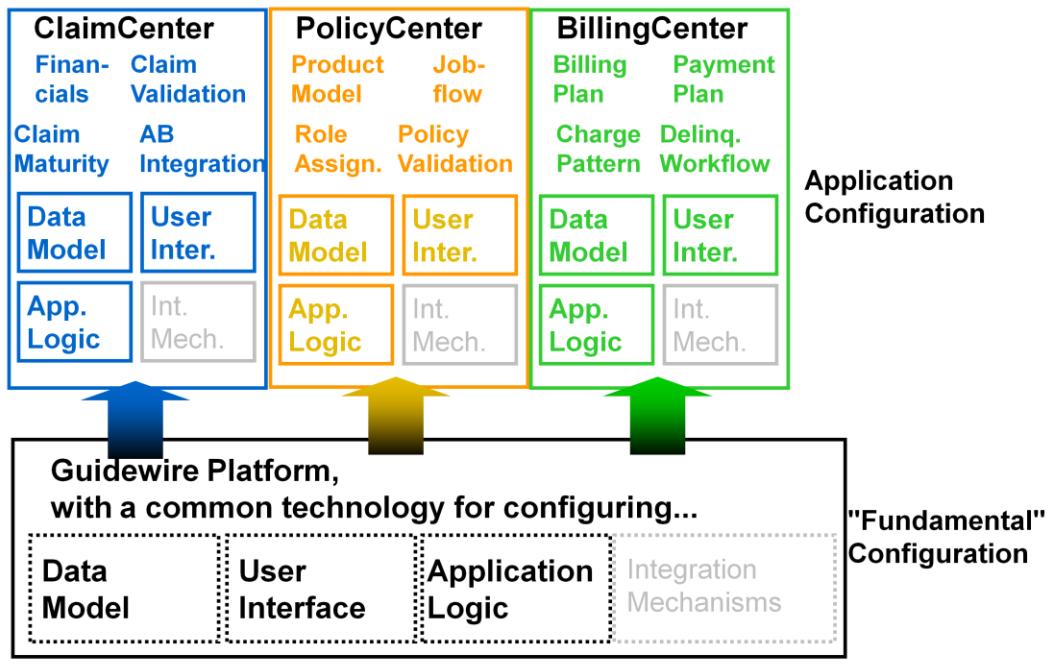
The PolicyCenter-specific functionality includes:

- The product model, which defines products, the policy lines assigned to each product, the coverages and coverables for the product, and the coverage terms for those coverages.
- Jobflow, which defines how policy transactions (such as submissions, renewals, changes, and cancellations) are executed.
- Role assignment, which defines how users are assigned responsibility for a given account, policy, or policy transaction.
- Policy validation, which validates that a given policy is valid, quotable, bindable, or issuable.

The BillingCenter-specific functionality includes:

- The management of invoices over the billing cycle from planned to billed to due and ultimately to paid.
- The production of provider commission statements for business collected directly from the client and the production and reconciliation of account statements where the provider collects monies due.
- The management of processing exceptions and events such as account statement queries. These are known as trouble tickets within BillingCenter.
- The management of the escalation cycle invoked when a payment becomes overdue and the account becomes delinquent.

# Configuration courses



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

23

G U I D E W I R E

Guidewire Education divides configuration technology into two categories: platform (also known as "fundamental") and "application-specific".

The "fundamental" configuration technology is the technology common to all Guidewire applications. One could think of it as the technology embedded within the Guidewire platform. This course focuses on the fundamental configuration technology.

The "application-specific" technology is the technology added to each Guidewire application to enable it to do policy-specific, billing-specific, or claim-specific processing.

The integration technology is covered in the "Application Integration" courses. This course does not focus on integration technology.

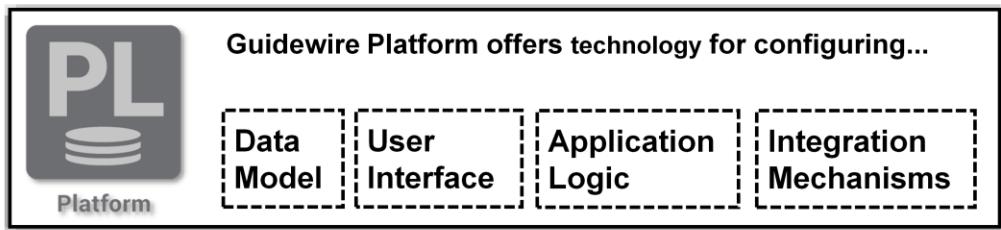
## Lesson outline

- Guidewire product architecture
- Guidewire configuration technology
- The Guidewire platform
- **TrainingApp**
- Starting Guidewire applications
- Guidewire Studio

# TrainingApp



- Developed exclusively for training
- Examples of each fundamental configuration feature
- Small number of entities and screens
- Excludes complex functionality



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

25

G U I D E W I R E

TrainingApp is not sold as genuine Guidewire product. It is used and maintained exclusively by Guidewire Education. The training application also has fundamental integration examples. These examples are discussed in the various Application Integration courses. TrainingApp excludes complex functionality designed for policy, billing, and claims processing.

# TrainingApp as a "business solution"

The screenshot shows a web-based application window titled 'Guidewire TrainingApp'. At the top, there are navigation tabs: 'Contacts' (with a dropdown arrow), 'Contact' (with a dropdown arrow), and 'Administration'. Below the tabs, it says 'Policy Person: Eric Andy'. On the left, a sidebar menu under 'Actions' lists: 'Summary' (which is selected and highlighted in blue), 'Details', 'Addresses (1)', 'Notes (0)', 'Analysis', 'Interactions', and 'History'. The main content area is titled 'Summary'. It contains sections for 'Basic Information' (Name: Eric Andy, Public ID: ab98, Created On: 10/02/2013, Assigned User) and 'Primary Address' (Address: 345 Fir Lane, La Canada, CA 91352, United States, Home). A 'Flag Entries' section shows one entry: 'View' (with a warning icon), 'View/Edit', 'Data Flagged' (10/02/2013), and 'Reason' (No email and contacts).

- "Mock" business solution for insurance carriers
- Designed to function like a contact management application
- Stores and manages contacts and related information

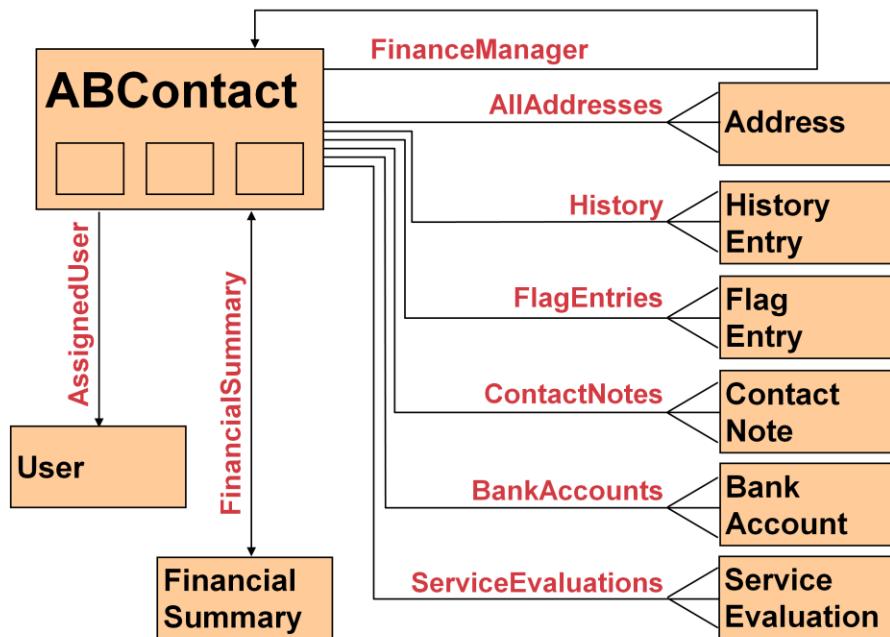
© Guidewire Software, Inc. 2001-2014. All rights reserved. Do not distribute without permission.

26

G U I D E W I R E

TrainingApp is based on Guidewire's ContactManager application. However, much of the ContactManager functionality has been removed or simplified to promote the learning of configuration fundamentals.

# TrainingApp data model



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

27

G U I D E W I R E

The core of the TrainingApp data model consists of nine entities and a small number of relationships which all center around ABContact.

The ABContact entity stores contacts. ("AB" stands for "Address Book". Many insurance carriers refer to their contact application as an Address Book.) The ABContact entity is subtyped, which is represented in the diagram above by the three small rectangles inside the ABContact rectangle. This is discussed in more detail on the following slide.

The Address entity stores addresses. An ABContact has one to many addresses.

Each contact has a history, which is a list of important occurrences in the life of that contact. The HistoryEntry entity stores a single entry in the ABContact's history. Given that a history entry is created to note the creation of the contact, every ABContact has one to many history entries.

The FlagEntry entity stores a flag entry. A flag entry is an issue pertaining to the ABContact which someone should attend to. An ABContact is "flagged" when it has one or more open flag entries. An ABContact can have zero to many flag entries.

The ContactNote entity stores notes about the ABContact. A contact note is a free-form text entry used to capture miscellaneous information about the contact. An ABContact can have zero to many contact notes.

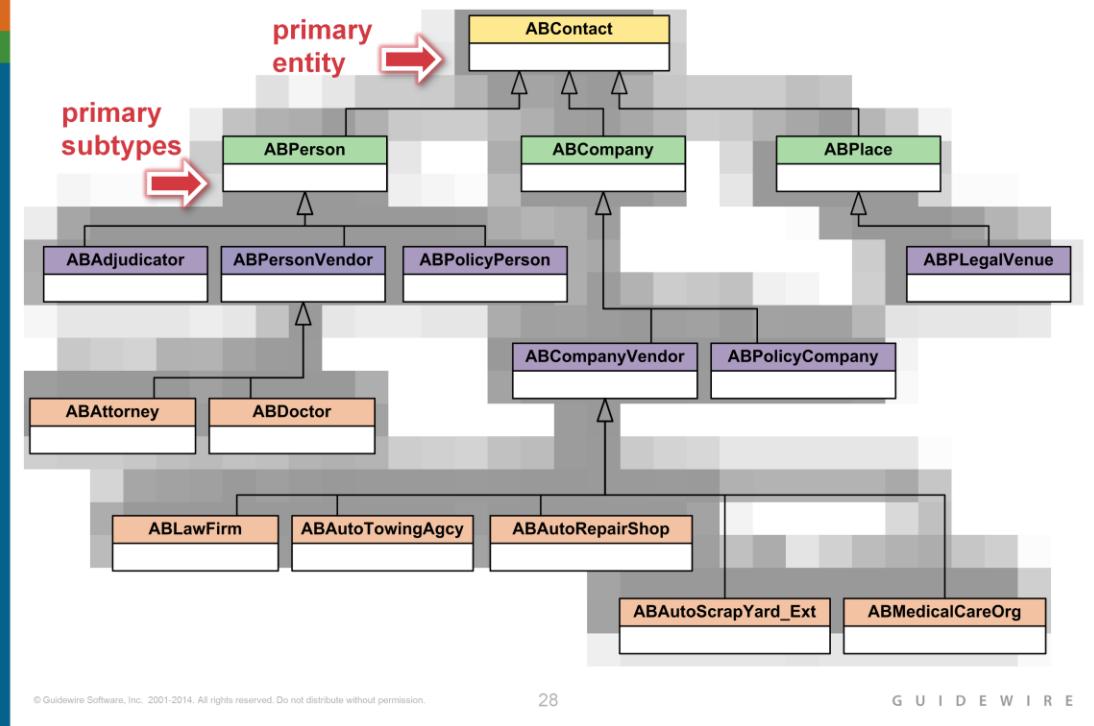
The BankAccount entity stores bank account information. An ABContact can have zero to many bank accounts.

The Service Evaluation entity stores information about a service evaluation, which evaluates the quality of the service provided by the contact and is most appropriate for vendor contacts such as doctors, lawyers, and auto repair shops. An ABContact can have zero to many service evaluations.

The User entity stores information about TrainingApp users. An ABContact may have zero or one assigned users.

The FinancialSummary entity stores information about an ABContact's financial summary. An ABContact may have zero or one financial summaries.

# Subtype hierarchy of ABContact



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

28

G U I D E W I R E

ABContacts are organized into a set of subtypes. The organization of the hierarchy helps to model information about contacts. Information common to all contacts can be established at the ABContact level. It is automatically inherited by all of its subtypes. Information specific to an ABPerson can be put at the ABPerson level. It will be inherited by all of its subtypes, but information on the ABPerson subtype is not available to ABCompany or ABPlace.

In the ABContact hierarchy, an ABPolicyPerson is a person who owns a policy issued by the carrier (such as an individual with a personal auto policy). An ABPolicyCompany is a company that owns a policy issued by the carrier (such as a construction company with a workers' compensation policy).

There is one entity that participates in the hierarchy that does not appear in the diagram above: ABUserContact. This type of contact represents a TrainingApp user and is used to store his or her contact information (such as home phone numbers). It has been excluded because it is not relevant to the configuration work to be done with TrainingApp.

Subtyping is also used within the three primary applications. For example:

- BillingCenter's Activity, Plan, and Contact entities
- ClaimCenter's Transaction, Incident, and Contact entities
- PolicyCenter's Job and Contact entities

# TrainingApp user interface

The screenshot shows the Guidewire TrainingApp interface. At the top, there's a navigation bar with tabs for 'Contacts' (selected), 'Contact' (dropdown), and 'Administration'. A 'Go to (Alt+/)' search bar is on the right. A context menu is open over a contact named 'Eric Andy', with options like 'International', 'Example List' (which is highlighted with a cursor icon), 'About', 'Preferences', 'Clear Layout Preference', and 'Log Out Super User'. The main content area is titled 'Summary' for 'Policy Person: Eric Andy'. It includes sections for 'Basic Information' (Name: Eric Andy, Public ID: ab:98, Created On: 10/02/2013) and 'Flag Entries' (a table with one row showing a warning flag icon, 'View' link, 'Date Flagged' (10/02/2013), and 'Reason' (No email address contact)). On the left, a sidebar has a 'Summary' tab selected, with links for 'Details', 'Addresses (1)', 'Notes (0)', 'Analysis', 'Interactions', and 'History'.

- Examples of all fundamental UI configurations
- List of where examples can be found

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

29

G U I D E W I R E

The TrainingApp user interface is simple and straightforward.

Users can search for ABContacts on the Search tab. Once an ABContact is selected, it is displayed on the Contact tab.

When you select the Contact from the tab, you see seven pages for the high-level areas of ABContact information (summary, details, addresses, notes, interactions, and history).

The Details tab contains a set of cards (such as "Company Info", "Phone & Addresses", "Bank Accounts", "Vendor Info", and "Analysis" shown above.) Some of the cards appear for all contacts (such as "Phone & Addresses"). Others are conditional and appear only when it is appropriate for the type of ABContact. (For example, "Company Info" is displayed only for ABContacts that are ABCompanies. "Vendor Info" is displayed only for ABContacts that are ABPersonVendors or ABCompanyVendors.)

TrainingApp is incomplete by design. Certain portions of the user interface have incomplete functionality. Students complete the functionality during one of the Guidewire training courses. TrainingApp also has at least one example of virtually every objective in the Configuration Fundamentals course. To find an example of any fundamental configuration technology, click the Example List link. This displays a list of examples arranged alphabetically by primary area (Data Model, User Interface, Gosu, and Cross-Section Functionality).

# TrainingApp application logic

The screenshot shows a contact editing interface. At the top, there are navigation tabs: 'Contacts' (selected), 'Contact' (selected), and 'Administration'. Below the tabs, the contact name 'Eric Andy' is displayed. The main area is titled 'Details'. A validation message 'Date of Birth : The date of birth cannot be in the future date.' is shown above the 'Date of Birth' field. The 'Person Info' tab is active, showing fields for Full Name (Eric Andy), Prefix (<none>), First Name (Eric), Middle Name ( ), Last Name (Andy), and Suffix (<none>). The 'Tax Info' tab is also visible. The 'Date of Birth' field contains '10/25/2013' and has a red border, indicating it is invalid. At the bottom left, a copyright notice reads: '© Guidewire Software, Inc. 2001-2014. All rights reserved. Do not distribute without permission.' At the bottom right, there are 'Update' and 'Cancel' buttons.

- Gosu language expresses application logic, including:

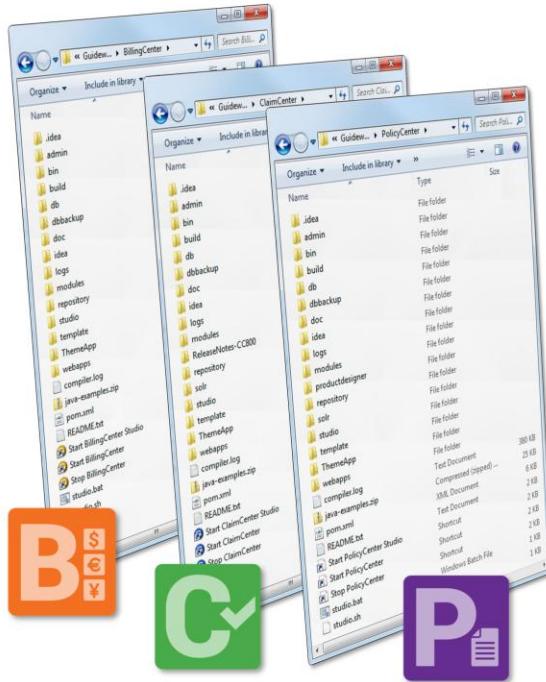
- Business rules
- PCF methods
- Enhancements
- Static methods in Gosu classes
- Script parameters
- Widget attributes

The screenshot above shows an example of application logic specified as a widget attribute. The "Date of Birth" widget has a validation expression that identifies the value provided cannot be greater than the current date. If this validation expression fails, then the update is blocked and the message "The data of birth cannot be in the future" is displayed in the user interface.

## Lesson outline

- Guidewire product architecture
- Guidewire configuration technology
- The Guidewire platform
- TrainingApp
- **Running Guidewire applications**
- Guidewire Studio

# Parallel directory structure



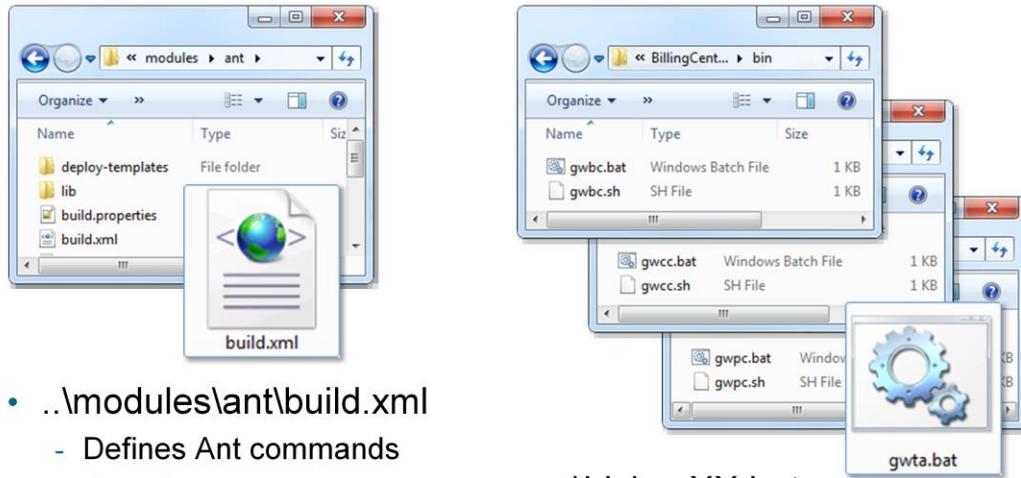
© Guidewire Software, Inc. 2001-2014. All rights reserved. Do not distribute without permission.

G U I D E W I R E

32

- Guidewire applications have a similar directory structure
  - admin
  - bin
  - idea
  - modules
  - studio
  - webapps

## Files used to start application



- ..\modules\ant\build.xml
  - Defines Ant commands
  - Start Guidewire application
  - Building web application archive (not all app servers)
  - Generate data and security dictionaries
- ..\bin\gwXX.bat
  - Batch file
  - Launch tasks defined in build.xml

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

33

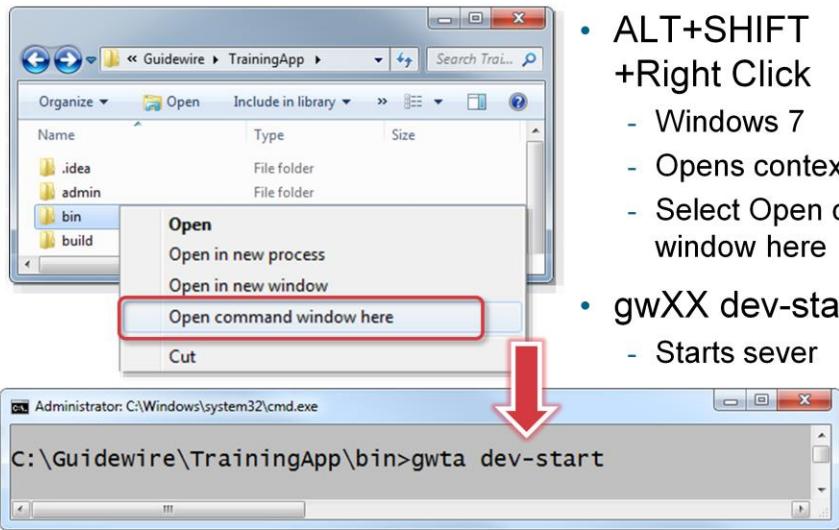
G U I D E W I R E

Every application's build.xml file is in <installDirectory>\modules\ant.

Every application's gw--.bat file is in <installDirectory>\bin. The name of the gw batch files varies from application to application:

- TrainingApp: gwta.bat
- BillingCenter: gwbc.bat
- ClaimCenter: gwcc.bat
- ContactManager (Address Book): gwab.bat
- PolicyCenter: gwpc.bat

## Start Guidewire: gwXX dev-start



- ALT+SHIFT +Right Click
  - Windows 7
  - Opens context menu
  - Select Open command window here
- gwXX dev-start
  - Starts sever

- " \*\*\*\* [ApplicationName] ready \*\*\*\* "
  - When running, all Guidewire applications display to console
  - TrainingApp "ready" message references ContactManager

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

34

GUIDEWIRE

To start any Guidewire end-user application in development mode:

1. At a command prompt, navigate to the bin directory of the installation.
2. Enter gwxx dev-start, where xx is the application's two-letter code.
3. When the message "\*\*\*\* application ready \*\*\*\*" appears, the application has started.

The command prompt window must remain open while the application is running. You can minimize it, however.

For more information on how to start a Guidewire end-user application in production mode, refer to the Installation Guide for the appropriate application.

# Accessing the application

- **http://hostName:port/appCode**
  - URL syntax to access a Guidewire application for training
- Example:
  - **http://localhost:8880/ab/**

Application	Application Name	Port	App Code
TrainingApp	ContactManager	8880	ab
BillingCenter	BillingCenter	8580	bc
ClaimCenter	ClaimCenter	8080	cc
ContactManager	ContactManager	8280	ab
PolicyCenter	PolicyCenter	8180	pc

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

35

G U I D E W I R E

TrainingApp's official application code and application name are "ab" and "ContactManager". This is a hard-coded aspect of Guidewire applications that cannot be customized.

## Run in one of two modes

Behavior	Development mode	Production mode
Available in QuickStart instance?	Yes	No
Browser title bar reads...	[DEV mode + build number] + Application	Application
Internal tools access	No restrictions	Restricted by permissions
QuickJump box commands?	Enabled	Disabled
Database exceptions logged at...	Error level	Warning level

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

36

G U I D E W I R E

For every Guidewire application, there is a QuickStart instance that can be run using a Jetty application server and an H2 database. These instances are useful for learning, demo, and development. A QuickStart instance cannot be run in production mode, however.

The Internal Tools interface contains controls for developing and administering Guidewire applications. It can be accessed only if:

The EnableInternalDebugTools parameter in config.xml is set to true.

The server is running in development mode.

To access Internal Tools, enter ALT + SHIFT + T. (It does not matter which user you are logged in as.)

The QuickJump box appears in the upper right-hand corner of the user interface. By default, it contains the text "Go to (Alt+/)". Typically, end users use this box to quickly navigate to various claims, policies, accounts, producers, or contacts. In development mode, however, more powerful commands can be executed. For example, the "run ImportSampleData" command loads sample data into a Guidewire application.

There is no obvious performance overhead to running in development mode—the performance characteristics of a running development server should closely resemble those of the same server in production mode.

Keep in mind that the information on this slide is not exhaustive. There are additional behavior differences that are either more technical in nature or are application-specific. For more information, refer to the Guidewire documentation.

## Commonly performed tasks

	Start Application in Dev Mode	Regenerate Dictionaries	Stop Application
<b>TrainingApp</b>	gwta dev-start	gwta regen-dictionary	gwta dev-stop
<b>BillingCenter</b>	gwbc dev-start	gwbc regen-dictionary	gwbc dev-stop
<b>ClaimCenter</b>	gwcc dev-start	gwcc regen-dictionary	gwcc dev-stop
<b>ContactManager (AddressBook)</b>	gwab dev-start	gwab regen-dictionary	gwab dev-stop
<b>PolicyCenter</b>	gwpc dev-start	gwpc regen-dictionary	gwpc dev-stop

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

37

G U I D E W I R E

Every gw batch file and build task it references is prefixed by two-letter codes. The code for each product appears above in red. The code for ContactManager is "ab", which represents "Address Book".

### Commonly Performed Tasks:

dev-start starts the given application in development mode.

regen-dictionary regenerates the data dictionary and the security dictionary.

dev-stop stops the application.

Note that—with the exception of the two-letter application code—the method for executing tasks is identical across all applications.

# Log in access accounts

- Super User account:  
su/gw
  - Only in default installation
  - All permissions
  - Special capabilities beyond permissions
- Alice Applegate account:  
aapplegate/gw
  - In sample data for every application
  - Represents basic end user
  - Typical permissions for underwriter, adjuster, billing and contact manager



The screenshot shows a web browser window titled "Guidewire TrainingApp". The URL is "localhost:8880/ab/ContactManager.do". The login form has "User name" set to "su" and "Password" set to "gw". There is a checked checkbox labeled "Keep me logged in" and a "Log in" button.

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



The screenshot shows a web browser window titled "Guidewire TrainingApp". The URL is "localhost:8880/ab/ContactManager.do". The login form has "User name" set to "aapplegate" and "Password" set to "gw". There is an unchecked checkbox labeled "Keep me logged in" and a "Log in" button.

G U I D E W I R E

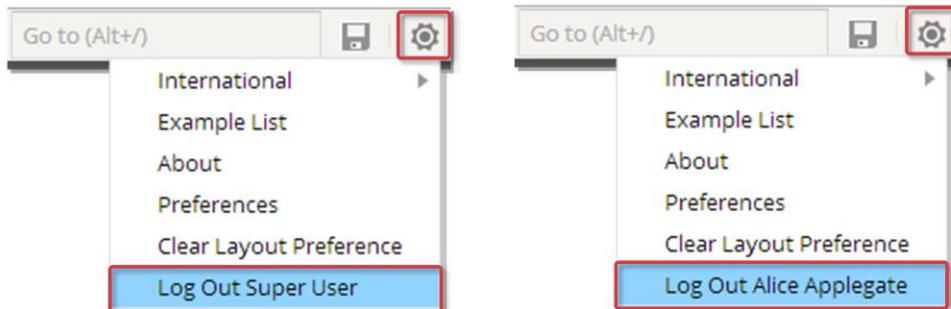
38

Because the Super User account has unique permissions behavior, the account should not be used during testing. All user accounts in Guidewire sample data have an initial password of "gw".

If a user checks the "Keep me logged in" field and logs in, then for the next seven days the user is logged in automatically whenever they navigate to the login page. (This works only if the application is hosted by the same application server.) An explicit logout will clear the "keep me logged in" state.

Note that the "Keep me logged in" functionality writes a cookie to the user's machine. As is the case with cookies, this may expose a security risk if other people get access to the cookie. (For example, someone could copy the cookie to another machine and then log in without entering a user name or password.) If there is a concern about browser cookie theft, administrators should remove this checkbox from the login page.

# Log out



- Click Settings menu
  - Menu option to Log Out shows the user name
- Select Log Out <username>

# Stop Guidewire: gwXX dev-stop



```
Administrator: TrainingApp 8.0.0 GA E20(CMD)
Terminate batch job (Y/N)? y
C:\Guidewire\TrainingApp\bin>gwta dev-stop
Buildfile: C:\Guidewire\TrainingApp\modules\ant\build.xml

tasks:
```

- **gwXX dev-stop** is a clean shutdown process that safely releases application resources (such as ports)
- If application is running in development mode:
  - Terminate the batch job (CTRL+C, y)
  - Execute **gwXX dev-stop**
- If application is running in production mode:
  - Always stop app by executing **gwXX dev-stop**

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

40

G U I D E W I R E

## gwXX dev-stop

The gwXX dev-stop command executes a clean shutdown. For example, if the application is in the middle of writing to a file, dev-stop allows the process to complete before the shutdown. gwXX dev-stop also releases all application ports. You can execute a gwXX dev-stop automatically through a script or manually through a second command prompt window. TrainingApp has a pre-built "Stop TrainingApp" shortcut in the Guidewire/TrainingApp directory that executes a "gwta dev-stop". To execute the command, simply double-click the shortcut.

### Terminating the batch job

When you are working in development mode, you can also stop the Guidewire application by terminating the batch job. To do this, press CTRL + C and then respond to the "terminate batch job?" prompt with a "y" (for yes). This has the advantage of keeping the command prompt window open and retaining the history of previously executed commands. This is useful if you need to stop and later restart the application. However, in some cases, the server may not release the port number. When this occurs, you need to execute a gwXX dev-stop before you can restart the application.

You should never terminate the batch job for an instance running in production mode.

### Additional notes

The command prompt window must remain open while the application is running. You can minimize it, however.

## Lesson outline

- Guidewire product architecture
- Guidewire configuration technology
- The Guidewire platform
- TrainingApp
- Starting Guidewire applications
- **Guidewire Studio**

# About Guidewire Studio

- Integrated Developer Environment (IDE)

- Gosu, XML, JAVA \*
- Refactoring
- Plugin extensibility
- Dynamic type support
- Make, run and debug
- Guidewire editors



- IntelliJ IDEA 12.1.x community edition

- No download needed; bundled with application



- Physical file based project representation

- Represents complete software solution
- Support for version control systems, e.g. Git, GitHub, Subversion

\* Within the Guidewire application project scope, limitations apply for Java development

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

42

G U I D E W I R E

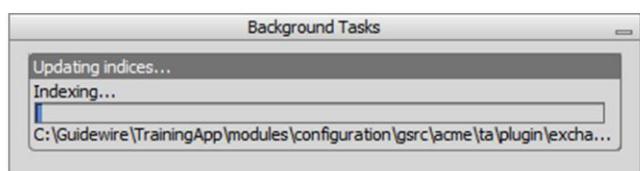
Guidewire Studio 8.0 is an Integrated Developer Environment (IDE) based on IntelliJ IDEA. Guidewire Studio comes bundled with your application as an application project. Guidewire Studio supports the IntelliJ IDEA directory-based format (.iml) that defines the application project. The file, configuration.iml, defines the base project contents.

A project is the highest level of organization in Guidewire Studio and includes project-wide settings as well as collections of modules and libraries. A project is an organizational unit that represents a complete software solution. With Guidewire Studio 8.0, it is much easier to identify, work with, manage, and version the project related resource. In Guidewire Studio 8.0, there is almost always a 1-to-1 relationship between the item being configured and the underlying file that is stored on the file system. In short, the project reflects the physical file locations on the disk.

# Start Studio: gwXX studio

```
C:\Administrator: TrainingApp 8.0 E17
C:\Guidewire\TrainingApp\bin>gwta studio
Buildfile: C:\Guidewire\TrainingApp\modules\ant\build.xml

studio:
=====
= Running main class: com.intellij.idea.Main
=   JVM args: -Xbootclasspath/a:../lib/boot.jar -Didea.proper
C-129.354 -Dfile.encoding=UTF-8 -ea -Didea.platform.prefix=Studio -Xm
studio:
=====
= Running main class: com.intellij.idea.Main
=   JVM args: -Xbootclasspath/a:../lib/boot.jar -Didea.proper
C-129.354 -Dfile.encoding=UTF-8 -ea -Didea.platform.prefix=Studio -Xm
=   Program args:
=     Classpath:
```



- To open the project, run
  - studio.bat
  - **gwXX studio** from the bin command window
- Indexing influences startup times
  - First time longest
  - Subsequent starts faster
- Possible to work while indexing!

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

43

G U I D E W I R E

In the specific Guidewire application folder root, the .idea directory contains a set of configuration files (.xml). The configuration data for projects and their components is in plain text XML files. Each file contains only a portion of configuration data pertaining to a certain functional area which is reflected in the name of a file, for example, compiler.xml, encodings.xml, modules.xml. These files contain information specific to the project itself, such as names and locations of the project component modules and compiler settings. If using version control for your project, you may want to consider including certain project files.

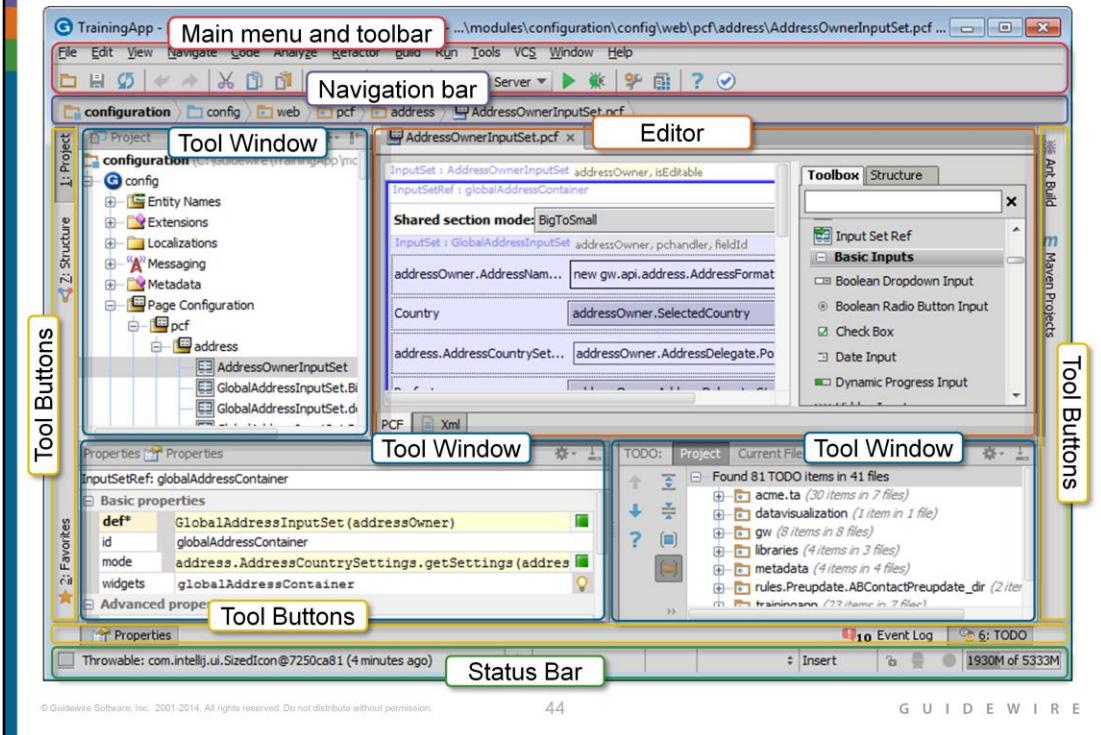
One exception is the workspace.xml file as this is specific to each individual Guidewire Studio user.

First time start up = Time taken to load the project + finish indexing for the first time.

Second time start up = Time taken to load the project + finish indexing for the second and third times.

In certain cases, you may want to rebuild your indexes. To rebuild your project indexes, you need to clean out the system caches. On the main menu, choose File → Invalidate Caches. Do not open a different app project with Guidewire Studio, use "xx studio" to start a different app. Guidewire Studio memory settings are in memory.properties.

# Anatomy of Guidewire Studio



1. Menus and toolbars - the main menu and toolbars let you carry out various commands.
2. Navigation bar that helps navigate through the project and open files for editing.
3. The status bar - indicates the status of your project, the entire IDE, and shows various warning and information messages.
4. The editor - here you create and modify the code.
5. Tool windows - secondary windows that provide access to various specific tasks (project management, source code search and navigation, running and debugging, integration with version control systems, etc.).

# Main menu and Toolbar

The screenshot shows the Guidewire Studio interface with the main menu bar at the top and a toolbar below it. The main menu bar has items: File, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, VCS, Window, and Help. The toolbar contains various icons for file operations like Open, Save, Print, and Find, along with icons for project management, code analysis, and version control.

**Main menu**

- Main menu commands
  - Open, edit and find files
  - View tool windows
  - Navigate class, file, symbol
  - Analyze dependencies
  - Refactor and analyze code
  - Make and rebuild
  - Run and debug
  - Gosu Scratchpad
  - Version control system
- Main toolbar buttons
  - Open, save, and synchronize files
  - Undo and redo
  - Cut, copy and paste
  - Find and replace
  - Make project
  - Edit configurations, settings, and structure
  - Run and debug
  - Refresh PCF
  - Gosu Scratchpad

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

45

G U I D E W I R E

Menus and toolbars let you carry out various commands. The main menu and toolbar contain commands that affect the entire project or large portions of it. Most command have an associated keyboard shortcut to enable quicker access to it.

The main menu contains commands for opening, creating projects, refactoring the code, running and debugging applications, keeping files under version control and more.

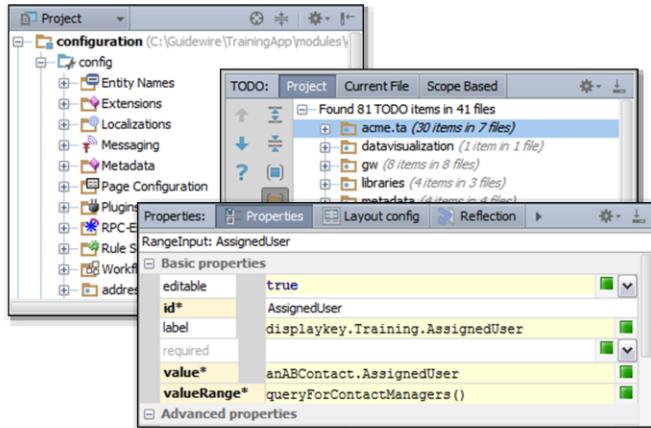
The main toolbar contains buttons that duplicate the essential commands for quicker access. You can hide the main toolbar, using the checked command on the toolbar context menu.

You can show or hide the main elements of the Guidewire Studio using the View menu.

Descriptions of the actions from all the menus and toolbar buttons are displayed in the left side of the Status bar.

If you know which action you want to perform, but do not know where to find the appropriate command, just press Ctrl+Shift+A and select the desired action from the suggestion list.

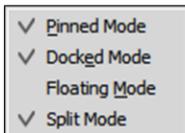
# Tool windows



- **View → Tool Windows**

- Project
- Favorites
- TODO
- Structure
- Properties

- Configure your Guidewire Studio environment
- Configure window modes
  - Pinned, Docked, Floating and Split Modes



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

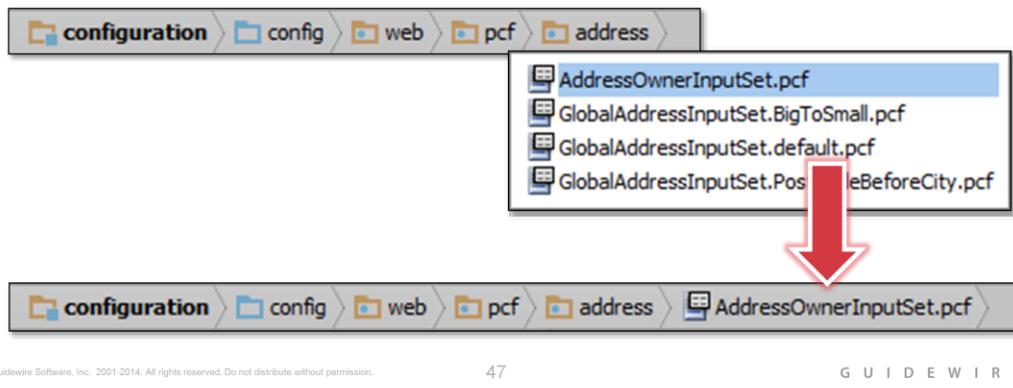
46

G U I D E W I R E

If a given Tool window is in Floating mode, then title bar context menu does not display options for Docked Mode or Split Mode.

# Navigation bar

- Open with
  - View → Navigation Bar
  - ALT+HOME
- Always shows the physical file structure names
  - Alternative to the Project tool window view
- Navigate through a project and open files for editing



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

47

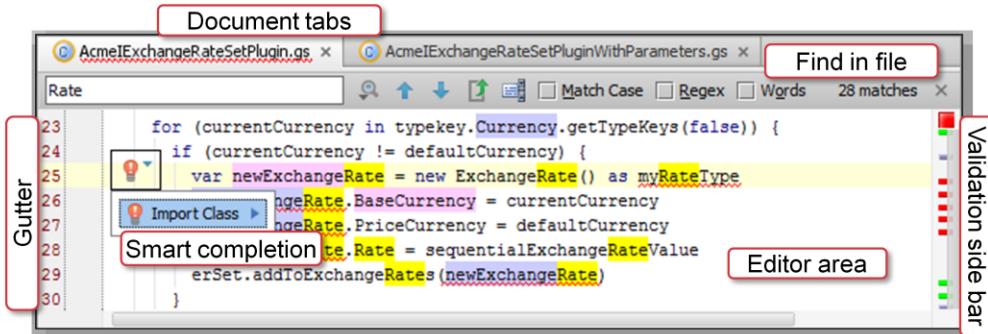
G U I D E W I R E

The Navigation bar always shows the physical file structure names for the file open in the Editor.

To show the navigation bar, in the View menu, select the Navigation bar option or press ALT+HOME. To hide the navigation bar, on the View menu, clear the Navigation bar option or click the close icon in the upper-right corner of the Navigation Bar. When the main toolbar is hidden, the Navigation bar shows the run/debug configurator selector, run server, and debug server buttons. If the Navigation bar is already hidden when you press ALT+HOME, it appears in floating mode.

For a type which is internally managed by the Guidewire application, does not expose a physical file, and is open in the Editor, the Navigation bar is empty.

# Editor



- Document tabs
  - Order tabs, switch between files, and split into windows
- Editor area
  - Syntax checker
  - Smart completion pop-up
  - Find in file (CTRL+F)
- Gutter Area
  - Line numbers, breakpoints, and icons
- Validation side bar
  - Warnings, errors, comments, TODOs, and highlights
  - Click to jump to source

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

48

GUIDEWIRE

Guidewire Studio features various editors. It is possible to enable and disable many of the Guidewire Studio Editors. To learn more about working with Guidewire Studio specific editors, refer to the Guidewire Studio Editors lesson.

**Editor area:** Use this area to type and edit your source code. The editor suggests numerous coding assistance facilities. For example, **smart completion pop-up** is one of the key editing assistance features that suggests method names, functions, tags and other keywords you are typing.

**Gutter area:** The left gutter provides additional information about your code and displays the various icons that identify the code structure, bookmarks, breakpoints, scope indicators, change markers and the code folding lines that let you hide arbitrary code blocks.

**Gutter icons:** Active Gutter Icons you can use to jump between an element's source and usage, for example, collapsing code blocks.

**Document tabs:** Enable quick navigation across the multiple documents you are working on. Clicking a tab brings its contents to front and makes it available for editing in the active editor. Right click the document tab to open the context menu, CTRL+Click a tab in the editor to navigate to any part of the file path. Select the necessary element in the drop-down, and the corresponding file path is opened in an external browser (e.g., in the Explorer, if your OS is Windows).

**Validation side bar / marker bar:** This is the bar to the right from the editing area, showing the green, red or yellow box on its top depending on whether your code is okay, or contains errors or warnings. This bar also displays active red, yellow, white, green and blue navigation stripes that let you jump exactly to the erroneous code, changed lines, search results, or TODO items.

**Keystrokes:** The Escape key (Esc) in any tool window moves the focus to the editor. SHIFT+ESC moves the focus to the editor and also hides the current (or last active) tool window. The F12 key moves the focus from the editor to the last focused tool window.

# Tool buttons

The screenshot shows a software interface with several tool buttons highlighted by red boxes:

- 1: Project (top left)
- 2: Structure (top left)
- Properties (bottom left)
- 2: Favorites (bottom left)
- 6: TODO (bottom left)

On the right side, there are two vertical stacks of tool windows:

- Ant Build
- Maven Projects

A context menu is open under the 'View' menu, with 'Tool Buttons' selected:

- View
- Navigate
- Code
- Analyze
- Refactor
- Build
- Tool Windows
- Quick Definition Ctrl+Shift+I
- ✓ Toolbar
- ✓ Tool Buttons**
- ✓ Status Bar
- ✓ Navigation Bar
- Active Editor
- Enter Full Screen

- Toggles open and close specific tool windows
  - Even if floating and not docked
- View → Tool Buttons
- Show hidden with ALT+ALT
- Toggle hidden tool buttons with Status bar

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

49

G U I D E W I R E

Tool buttons surround the border of the working environment. Tool buttons are easily toggled on and off.

# Status bar

	PluginException: Fai... (5 minutes)		42:72	LF	UTF-8	Insert				885M of 3555M
Icon	Description									
	Toggles between showing or hiding tool window bars. ALT+ALT to show hidden tool windows bars.									
	Shows description of a current command or current exception.									
	Opens Background Tasks manager									
42:72	Line number and caret position of cursor in current editor.									
	View and change line endings of the current editor file.									
	View and change encoding of the current file in the editor.									
	Insert key toggles the Insert and Overwrite modes. In the Overwrite mode, the cursor is a rectangle.									
	Indicates if file in the current editor is read-only or writable.									
	Click to edit the code inspection profile settings.									
	Click to show IDE error.									
885M of 3555M	Shows the current heap level and memory usage.									

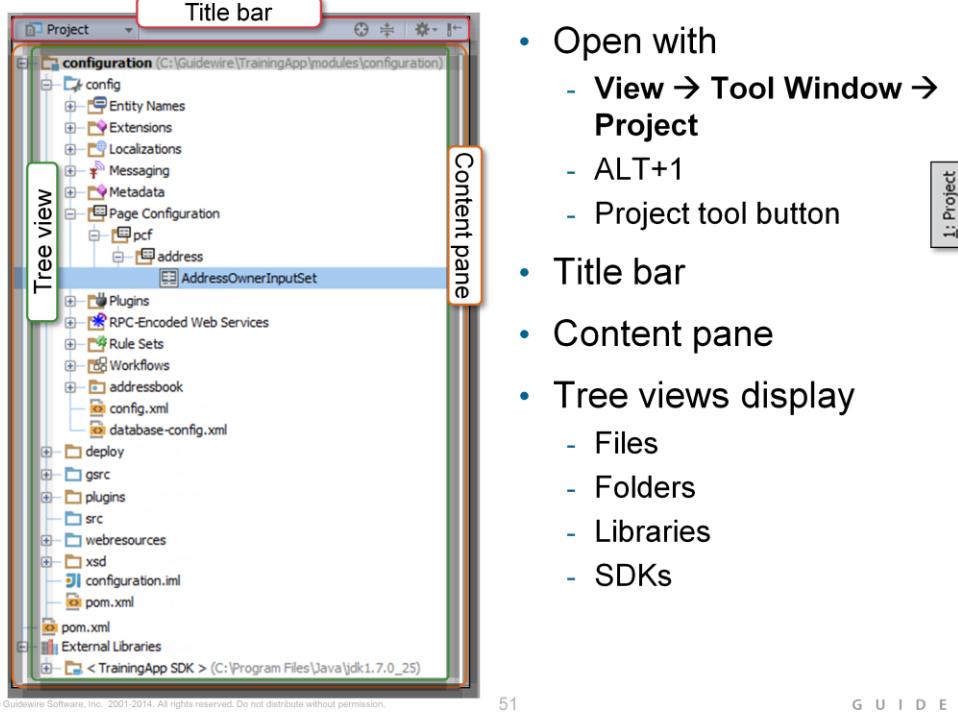
© Guidewire Software, Inc. 2001-2014. All rights reserved. Do not distribute without permission.

50

G U I D E W I R E

The status bar indicates the current IDE state and lets you carry out certain environment maintenance tasks.

# Project Tool Window



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

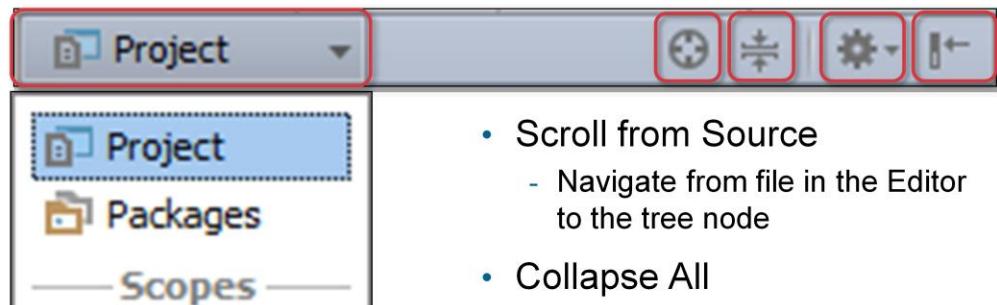
51

G U I D E W I R E

The Project Tool Window displays physical folders, files, packages, modules, and SDKs, as they exist on disk. In certain cases, a node item in the hierarchy may use an alias name instead of the physical name. The Navigation bar always shows the physical name and physical path, if it exists.

Within a specific view, the Project Tool Window allows you to perform specific tasks such as creating new directories, packages, files, and classes as well as opening files in the editor and navigating to code.

# Title bar in Project Tool Window



- Scroll from Source
  - Navigate from file in the Editor to the tree node
- Collapse All
  - Collapses all nodes
- Views
  - Project view (default) shows directory structure and dependencies
  - Packages view shows package structure and NO dependencies
- Current Configuration
  - Configure the current view
  - View modes
- Hide
  - Hide the tool window

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

52

GUIDEWIRE

Title buttons include Scroll from Source, Collapse All, Current Configuration, and Hide.

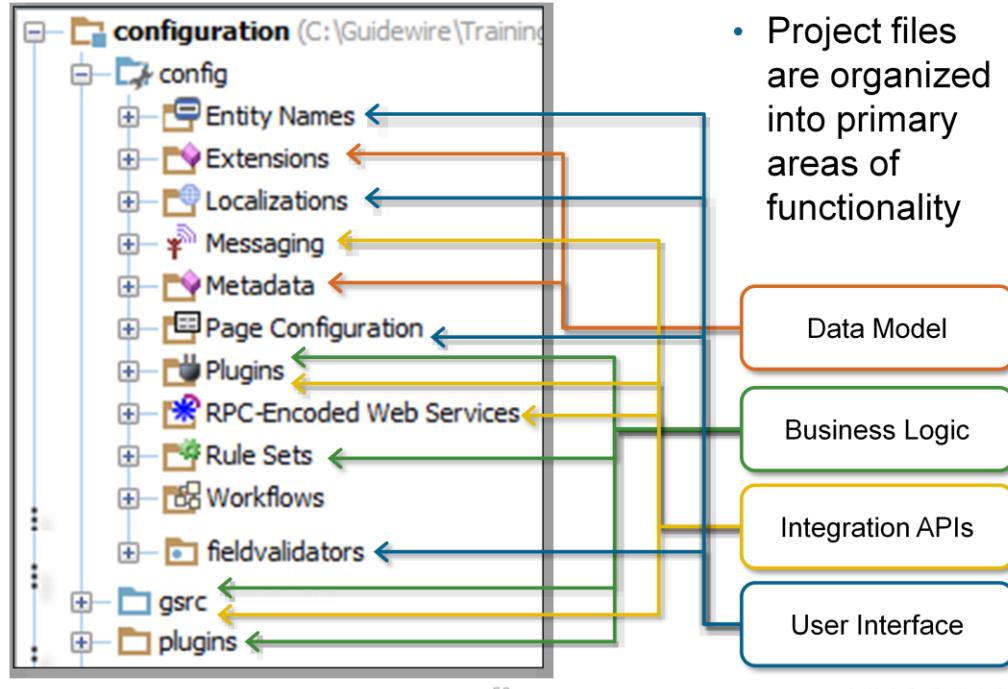
For the open and selected file in the Editor, the Scroll from Source jumps to the specific file in the tree view in the content pane.

Collapse All collapses all folders in the tree view in the content pane.

Use the Current Configuration to open a context menu for configuring the current view and changing the tool window viewing modes. Depending on the viewing mode, the current configuration context menu displays different options than the Title toolbar context menu. Most of the menu items are options that you can enable and disable. Check marked menu options are enabled options.

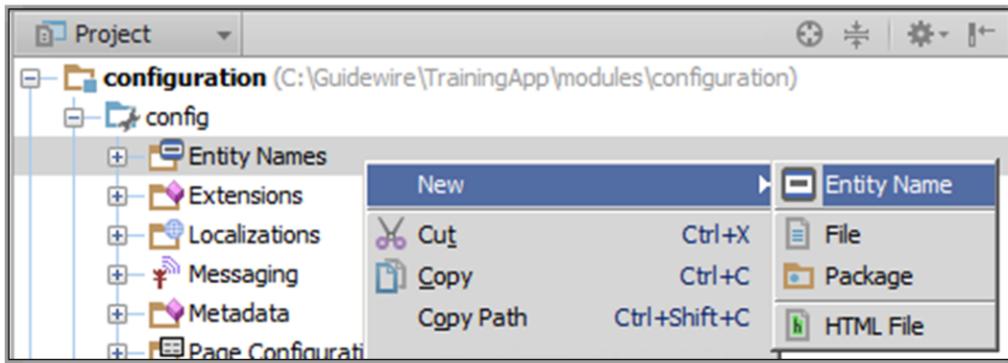
Click the hide icon to close the tool window.

# Project view structure



The Project View is for TrainingApp 8.0. Some folders and files are removed from the Project View screenshot.

# Project view context menu variations



- Folder determines context menu options, for example:
  - \Entity Names\ = Entity Names
  - \Extensions\Typelist\ = Typelist and Typelist Extensions
  - \Rules Sets\ = Rules Set Category

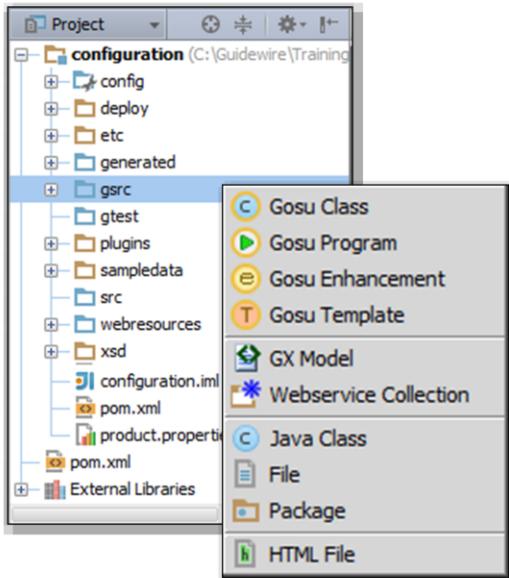
© Guidewire Software, Inc. 2001-2014. All rights reserved. Do not distribute without permission.

54

G U I D E W I R E

The file options opens the Register New File Type Association dialog. In most cases, you do not need to associate a new file type with the core configuration setting for Guidewire application in Guidewire Studio.

## Context menu for \gsrc\



- Create your packages, classes, plugins classes, and web services in the \gsrc\ folder
  - Gosu Class
  - Gosu Program
  - Gosu Enhancement
  - Gosu Template
  - GX Model
  - Webservice Collection
- Creating Java Class files directly is **NOT** supported

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

55

G U I D E W I R E

It is possible to add Java classes and JARS to the gsrc directory using file explorer. However, you should not create a Java Class directly in the /gsrc/ folder as this is not supported.

## Helpful context menu commands (1)



- **Mark as ...**
  - Toggle between editing a file in plain text or with a Guidewire Editor
  - Plain text has no validation or smart code completion

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

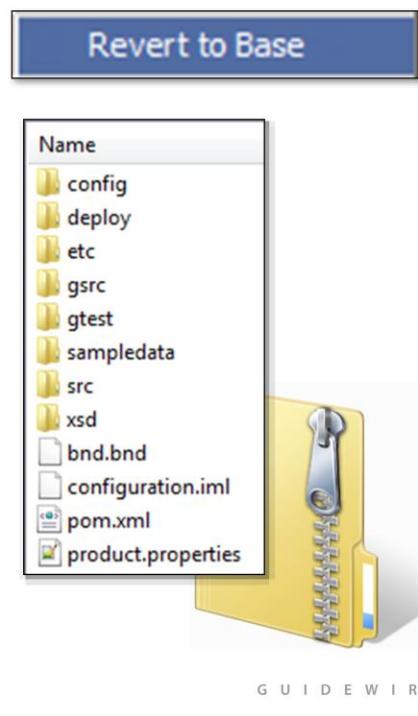
56

G U I D E W I R E

It is possible to switch between using a Guidewire Studio Editor and using the built-in text editor. In certain cases, editing a raw file with a text editor can help fix specific issues, such as highly malformed PCF XML.

## Helpful context menu commands (2)

- Revert to Base
  - Restores file from  
`/modules/base.zip`
- Replaces the selected file with  
the original Guidewire  
application file found in
  - config
  - deploy
  - etc
  - gsrc
  - src
- Restores only files!
- Command not applicable for  
Rule Sets and Rules



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

57

G U I D E W I R E

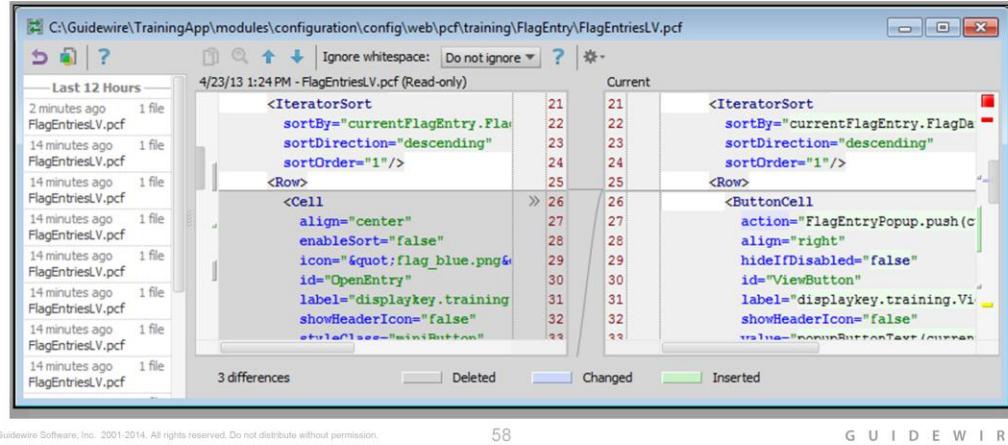
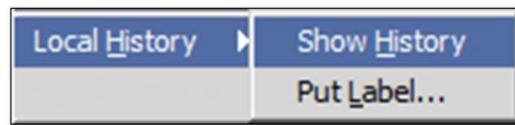
Revert to base only support files and not folders. File and folders both exist, however, in the base.zip file.

It is important to note that rule set categories, rule sets, and rules cannot revert to the original file using the Guidewire Studio revert to base command. However, it is possible to revert to a previous change while using the Show History dialog.

## Helpful context menu commands (3)

- Local History → Show History

- Compare selected version to current
  - Current pane is editable
- Revert to previous change
  - Use labels to mark points



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

58

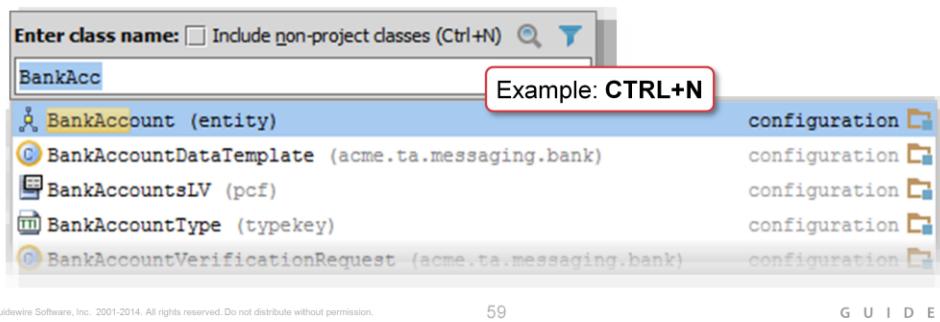
G U I D E W I R E

To see your local history of changes in a file, invoke Local History → Show History in the popup menu. You may navigate through different file versions, see the differences and rollback to any previous version. Use the same popup menu item to see the history of changes on a directory. You will never lose any code with this feature!

You can also select Show History from the VCS Operations popup. The keystroke is ALT+` (Back quote)+5 to show the history.

# Navigation keystrokes

- Navigate
  - CTRL+N
    - Class
  - CTRL+SHIFT+N
    - File
  - CTRL+ALT+SHIFT+N
    - Symbol
- Find
  - CTRL+F
    - Find in editor
  - CTRL+SHIFT+F
    - In path
  - CTRL+H
    - View hierarchy
  - CTRL+E
    - Recent files



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

59

G U I D E W I R E

CTRL+N: Use for finding entity, entity name (displayname), library, typelist, pcf, rule, and classes. Some things found are not files, but rather internal items, such as an entity subtype typelist. One way to determine if an item is internal or virtual is to see if the Navigation bar is empty.

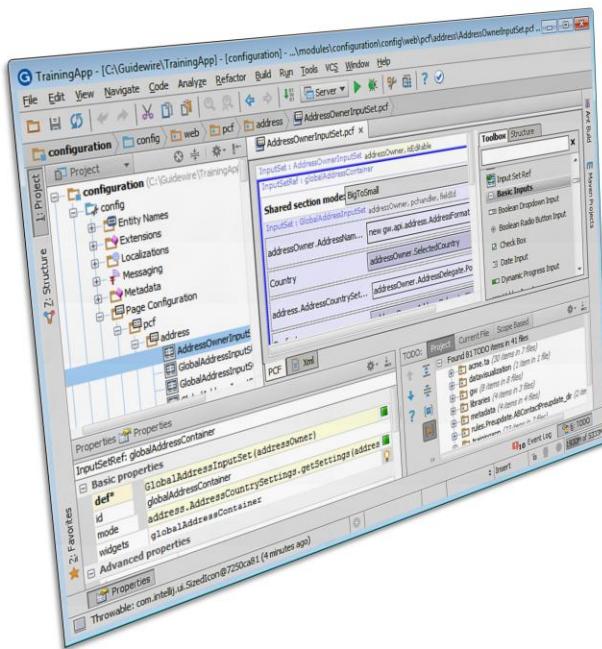
CTRL+SHIFT+N: Great for finding files like config.xml. Using wildcards (\*), e.g., \*.wsc, is supported.

CTRL+ALT+SHIFT+N: Find a symbol in related files. This is great for finding which resources contain an enum, method, property, TC\_name, etc.

You can bring forward the list of all usages of a class, method or variable across the whole project, and quickly jump to the selected usage. To do that, place the caret at the symbol's name or at its usage in code with ALT+F7 or in a popup frame with Ctrl+ALT+F7. Scroll the list and click the desired usage.

To navigate to the implementation(s) of an abstract method, position the caret at its usage or its name in the declaration and press Ctrl+ALT+B.

# Make, Run, and Debug in Studio!



- Make Project
  - Compiles the solution
- Run 'Server'
  - Starts the application server
- Debug 'Server'
  - Starts the application server in debug mode
- Run and Debug have individual console windows
- Stop + Run/Debug = Restart server

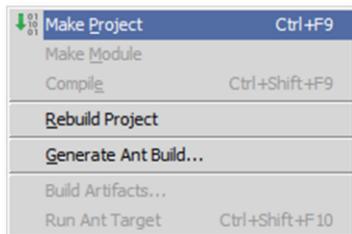
GUIDEWIRE

In Guidewire Studio 7, in order to debug an application server, you first need to run the application server in debug mode, start Studio, and then connect to the instance running in debug mode.

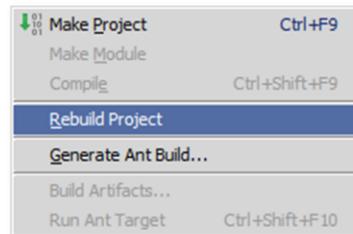
In Guidewire Studio 8, simply start Studio. Then, execute the Debug the Server command ALT+SHIFT+F9. If you need to debug your code in Gosu Scratchpad, open Gosu Scratchpad and click Run in Debug Process.

# Make and Rebuild project

## Make project



## Rebuild project



- Build → Make project
  - CTRL+F9
  - Compiles only modified files since the last compilation
- Run → Reload Changed Classes
  - Works when in Run or Debug

- Build → Rebuild project
- All the source files in the project are recompiled

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

61

G U I D E W I R E

Make Project: All the source files in the entire project that have been modified since the last compilation are compiled. Dependent source files, if appropriate, are also compiled. Additionally, the tasks tied to the compilation or make process on modified sources are performed.

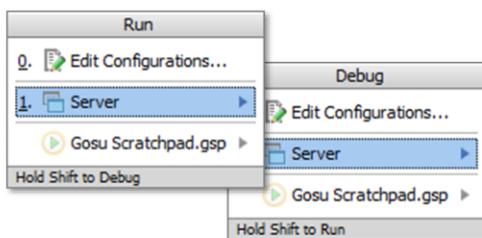
Rebuild Project: All the source files in the project are recompiled. This may be necessary when the classpath entries have changed, for example, SDKs or libraries being used added, removed or altered.

# Run and debug server

## Run Server

ALT+SHIFT+F10

- Errors in Messages window
- Console information and generated output in Run window
- View running instance on Jetty socket and port



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

## Debug Server

ALT+SHIFT+F9

- Errors in Messages window
- Debug window contains
  - Debugger and Console tabs
- Console information and generated output to Console tab in Debug window
- View running instance on Jetty socket and port

G U I D E W I R E

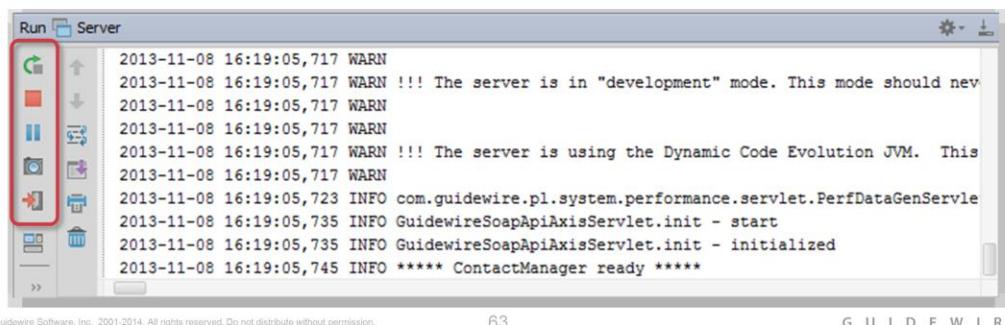
In Guidewire Studio, you can run, debug (and test) your Guidewire application without leaving Guidewire Studio. You can run or debug your project by selecting the Run Server and/or Debug Server menu commands. In certain cases, you can use the SHIFT+F10 keystroke to Run Server and SHIFT+F9 keystroke to Debug Server. However, if you are actively working in Gosu Scratchpad (run/debug/stop), these keystrokes will run and debug, respectively, the code in Gosu Scratchpad! For this reason, you may want to use ALT+SHIFT+F10 and ALT+SHIFT+F9 to open the respective dialogs to run and debug the server instance for your project.

The console information and the generated output are displayed in the Run tool window or the Debug tool window in the Console tab.

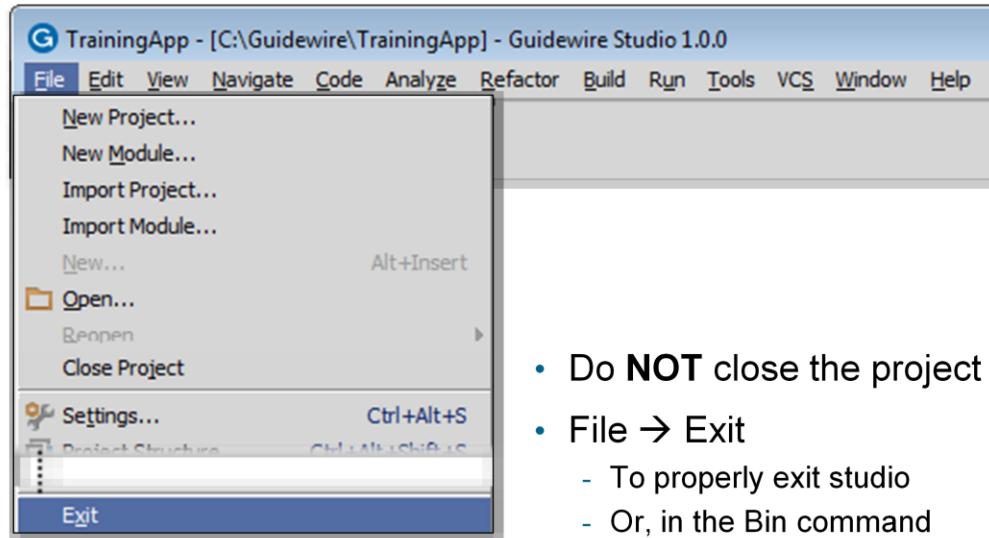
The debugger enables you to execute your application step by step, examine program information related to variables, watches, or threads, and change your program without leaving Guidewire Studio. Prior to launching the debugger session, it makes sense to set breakpoints that cause the debugger to suspend the application or take some other actions when breakpoints are reached in code.

# Run Tool Window

- ALT+4
  - Opens Run window
- View Run 'Server' application log
  - Same as console window output
  - View console output
- Gutter commands
  - Stop server, (Re)run server, Pause Output, Exit run



## Exit Studio: File → Exit



- Do **NOT** close the project
- File → Exit
  - To properly exit studio
  - Or, in the Bin command window, enter the exit command

## Lesson objectives review

- You should now be able to:
  - Describe the product architecture for Guidewire products
  - Recall the primary components used to configure Guidewire products
  - Identify the relationship between the Guidewire platform and the Guidewire applications
  - Explain the basic functionality of TrainingApp
  - Start a development instance of a Guidewire application
  - Describe the purpose of Guidewire Studio

## Review questions

1. What are the three tiers in the Guidewire application architecture?
2. What are the four fundamental areas of Guidewire configuration technology?
3. In which fundamental area of configuration are the following used?
  - a) Page configuration files (PCFs)?
  - b) Web services, messaging, and plugins?
  - c) Gosu?
  - d) Data model entities?
4. How do you start a Guidewire application?
5. How do you stop a Guidewire application?

## Answers

- 1) Data, application, presentation (or user interface)
- 2) Data model, user interface, business logic, integration mechanisms
- 3a) User interface
- 3b) Integration mechanisms
- 3c) Business logic
- 3d) Data model
- 4) Open a command prompt, navigate to the application \bin directory, and execute the start command ("gwXX dev-start" or "gwXX studio").
- 5) Close the command prompt (or, in the case of Studio, the application window)

# 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.