ClaimCenter Reporting Guide

Release 6.0.8

Product Name: Guidewire ClaimCenter
Product Release: 6.0.8
Document Name: *ClaimCenter Reporting Guide*
Document Revision: 05-February-2013

Contents

Part III
Working with Reports

About This Document

This document contains an overview of ClaimCenter Standard Reporting. ClaimCenter integrates with InetSoft Style Report Enterprise Edition to provide a number of ClaimCenter-specific reports. This document details how to install and configure the InetSoft application to work with Guidewire ClaimCenter, as well as how to create and run reports.

This topic includes:

Intended Audience

Guidewire intends this document for the following readers:

- Report writers who actually implement and maintain reports
- Business analysts who define the reports, report types, and the types of information needed in a report
- Implementation team members or IT staff who do not directly write reports, but seek a better understanding of the ClaimCenter reporting functionality

Assumed Knowledge

This document assumes that you are already familiar with the Guidewire ClaimCenter application. Guidewire recommends that you have read *ClaimCenter Application Guide* or that you have experience with the ClaimCenter screens and functionality before you begin working with reports. This document also assumes that you are familiar with the InetSoft reporting documentation. If you need access to the InetSoft documentation, contact Guidewire Support.

Related Documents

See the following documents for further information:

ClaimCenter Application Guide – Introduces the application, explains application concepts, and provides a high-level view of major features and business goals of ClaimCenter. This is your first place to look when learning about a feature. This book is written for all audiences.

ClaimCenter Gosu Generated Documentation – Documents all types visible from the Gosu type system. This includes Guidewire entities, Gosu classes, utility classes, Gosu plugin definitions, and Java types that are available from Gosu. With a local copy of the product, you can regenerate this documentation. From the `ClaimCenter/bin` directory run the `gwcc regen-gosudoc` command.

ClaimCenter Configuration Guide – Describes how to configure ClaimCenter and includes basic steps and examples for implementing such configurations. This guide is intended for IT staff and system integrators who configure ClaimCenter for an initial implementation or create custom enhancements. This guide is intended as a reference, not to be read cover-to-cover.

Gosu Reference Guide – Describes the syntax of expressions and statements within ClaimCenter. This document also provides examples of how the syntax is used when creating rules. This document is intended for rule writers who create and maintain rules in Guidewire Studio.

ClaimCenter Installation Guide – Describes how to install a new copy of ClaimCenter into Windows or UNIX environments. This guide is intended for system administrators and developers who need to install ClaimCenter.

ClaimCenter Integration Guide – Provides an architectural overview and examples of how to integrate ClaimCenter with external systems and custom code. This document is a learning tool for explanations and examples with links to the *Java API Reference Javadoc* and *SOAP API Javadoc* for further details. This document is written for integration programmers and consultants.

ClaimCenter Java API Reference Javadoc – Documents the Java API for integration programmers. It contains API details such as classes, interfaces, method parameters, return values, and behavior of each method. See the *Integration Guide* for more details. The Java API Reference Javadoc includes:

- Java plugin interface definitions (also usable to write Gosu plugins)
- Details of ClaimCenter Java entities, including both the entity data and additional methods called domain methods
- General Java utility classes in a separate Javadoc directory.

ClaimCenter Rules Guide – Describes the business rule methodology, rule categories for ClaimCenter, and rule syntax for Guidewire Studio. This book is intended for programmers who write Gosu business rules and analysts who define the business rule logic.

ClaimCenter SOAP API Reference Javadoc – Documents the SOAP APIs and entities for integration programmers. It includes: (1) Web service (SOAP) API interfaces; (2) ClaimCenter SOAP entities which are simplified versions of ClaimCenter entities; (3) SOAP-specific Java utility classes. See the *Integration Guide* for more details.

ClaimCenter Upgrade Guide – Provides instructions to upgrade ClaimCenter.

ClaimCenter Data Dictionary – Describes the ClaimCenter data model, including your custom data model extensions. To generate the dictionary, go to the `ClaimCenter/bin` directory and run the `gwcc regen-dictionary` command. To view the dictionary, open the `ClaimCenter/build/dictionary/data/index.html` file. For more information about generating and using the *Data Dictionary*, see the *ClaimCenter Configuration Guide*.

ClaimCenter Security Dictionary – Documents security permissions, roles, and the relationships between them. Generate the dictionary by going to the `ClaimCenter/bin` directory and running the `gwcc regen-dictionary` command. To view the dictionary, open the `ClaimCenter/build/dictionary/security/index.html` file. For more information about generating the *Security Dictionary*, see the *ClaimCenter Configuration Guide*.

Conventions In This Document

Text style	Meaning	Examples
italic	Emphasis, special terminology, or a book title.	A *destination* sends messages to an external system.
bold	Strong emphasis within standard text or table text.	You **must** define this property.
narrow bold	The name of a user interface element, such as a button name, a menu item name, or a tab name.	Next, click **Submit**.
`monospaced`	Literal text that you can type into code, computer output, class names, URLs, code examples, parameter names, string literals, and other objects that might appear in programming code.	Get the field from the `Address` object.
`monospaced italic`	Parameter names or other variable placeholder text within URLs or other code snippets.	Use `getName(`*`first`*`, `*`last`*`)`. `http://`*`SERVERNAME`*`/a.html`.

Support

For assistance with this software release, contact Guidewire Customer Support:

- At the Guidewire Resource Center – `http://guidewire.custhelp.com`
- By email – `support@guidewire.com`
- By phone – +1-650-356-4955

part I

Installing Guidewire Standard Reporting

chapter 1

Before Starting the Installation

This section describes what you must do and understand before you attempt to install and configure Guidewire Standard Reporting. ClaimCenter Standard Reporting is an optional ClaimCenter configuration. This section assumes the following:

- You are installing Guidewire ClaimCenter 6.0.8 as the ClaimCenter application server.
- You are installing InetSoft StyleReport Enterprise Edition version 10.1 as the report server.

This topic includes:

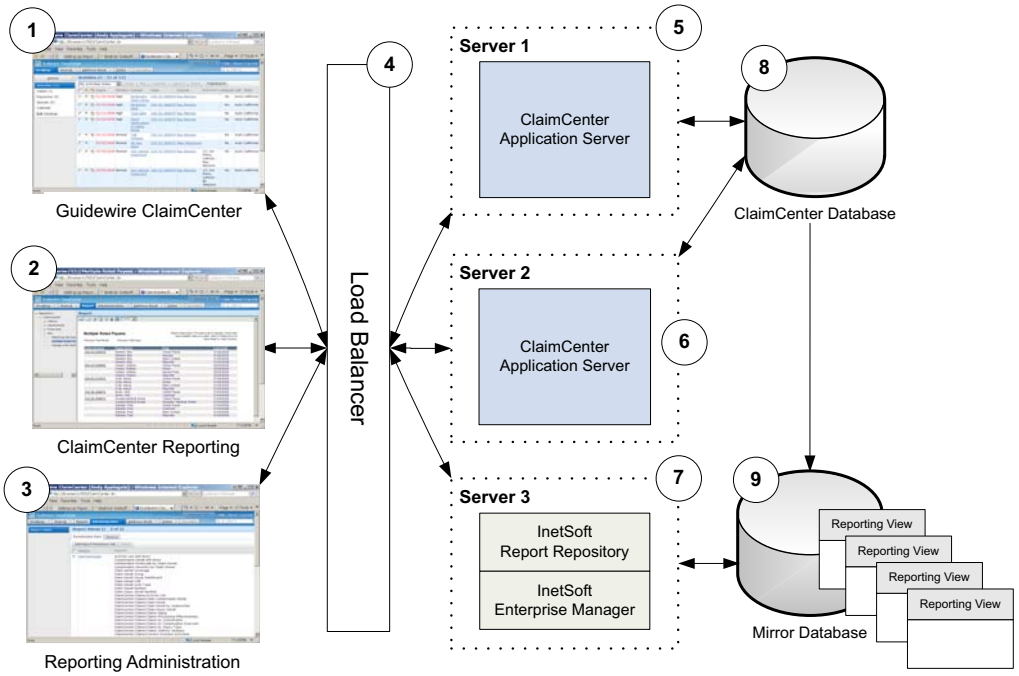
Guidewire ClaimCenter Standard Reporting

IMPORTANT Only install a stand-alone version of InetSoft if you need access to the InetSoft Report Designer. Otherwise, use the InetSoft Enterprise Manager bundled with Guidewire ClaimCenter to access the InetSoft report administration functions.

The following diagram shows the recommended configuration for Guidewire Standard Reporting with ClaimCenter running in a clustered environment with a load balancer. The configuration contains the following:

- One or more ClaimCenter application servers
- One load balancer
- One InetSoft reporting server
- One ClaimCenter database

- One mirror database for reporting data



The data flow through the various application components takes the following form:

1. The ClaimCenter application (1) passes requests and data to the Load Balancer (4). The Load Balancer passes the request to the ClaimCenter application server (6), which manages communication with the ClaimCenter database (8). The Load Balancer also manages any response to the request from the ClaimCenter application server.
2. The Load Balancer manages the data flow between the ClaimCenter **Report** tab and the InetSoft report server (7). The report server communicates with the reporting database (9), which is a mirror of the ClaimCenter database with the addition of specialized reporting views.
3. The InetSoft Enterprise Manager (7) performs reporting administrative tasks. The Load Balancer manages the communication between the ClaimCenter **Report Administration** screen and the report server.

You can cluster multiple ClaimCenter application servers as well as cluster multiple reporting servers (InetSoft Report Repositories).

- For general information on setting up Guidewire ClaimCenter in a clustered environment, see “Managing Clustered Servers” on page 81 in the *System Administration Guide*.
- For specific information on setting up Guidewire ClaimCenter with Standard Reporting in a clustered environment, see “Reporting in a Clustered Environment” on page 75.
- For information on setting up and configuring multiple InetSoft Report Repositories, refer to the InetSoft documentation, specifically the InetSoft *Style Report Administration Reference Guide*.

Mirror Database Recommendations

IMPORTANT You **must** create a dedicated reporting database that is separate from the ClaimCenter database. Failure to do so may create severe performance issues.

Use the following guidelines to create a mirror of the ClaimCenter database for use with the ClaimCenter Standard Reporting.

Oracle 11g

If you intend to use Oracle 11g as your ClaimCenter database, Guidewire recommends the following:

- Replicate the ClaimCenter database on an Oracle Logical Standby (Data Guard) database.
- Create all necessary reporting views on this replicated database.
- Use the requirements of your business practices to determine how often you need to update the mirror database.

Refer to the Oracle documentation for information on setting up an Oracle Logical Standby database. You can access Oracle Data Guard documentation at the following location:

```
http://www.oracle.com/technology/deploy/availability/index.html
```

Microsoft SQL Server 2008

If you intend to use Microsoft SQL Server 2008 as your database, Guidewire recommends the following:

- Replicate the ClaimCenter database on a Microsoft SQL Replication database using the Microsoft SQL Server Management Studio. You will need to create a new Replication *Publication*. Guidewire recommends that you use **only** one-way replication methods, for example, Transactional replication. Do **not** use Updatable subscriptions.
- Create all necessary reporting views on this replicated database.
- Use the requirements of your business practices to determine how often you need to update the mirror database.

Refer to the Microsoft documentation for information on setting up a MS SQL Replication database. You can access this documentation at the following location:

```
http://msdn2.microsoft.com/en-us/library/ms203721.aspx
```

Materialized Views (Oracle)

Guidewire uses a number of materialized views (MVs) for reporting. (This is true for reporting installations running on Oracle only.) The use of materialized views improves performance significantly. Guidewire recommends that you refresh these materialized views at the same time that you refresh the ClaimCenter mirror database.

ClaimCenter Tables Used in the Reporting Views

The ClaimCenter reporting views use the following tables:

cc_* tables

- activity
- activitypattern
- address
- catastrophe
- check
- claim
- claimmetric
- claimcontact
- claimcontractrole
- contact
- exposure
- exposuremetric
- group
- history
- incident
- matter

- policy
- reserveline
- subrogationsummary
- subroadverseparty
- taccount
- taccountlineitem
- taccounttransaction
- transaction
- transactionlineitem
- transactionset
- user

cctl_* tables

- activityclass
- activitystatus
- activitytype
- approvalstatus
- claimmetric
- claimtier
- claimstate
- contactrole
- costcategory
- costtype
- coveragetype
- currency
- detailedinjurytype
- exposuremetric
- exposuretier
- exposurestate
- exposuretype
- faultrating
- flaggedtype
- historytype
- languagetype
- litigationstatus
- lobcode
- losscause
- losspartytype
- losstype
- paymenttype
- policyratingplan
- policytype
- state
- subrogationstatus
- subrogovernmentinvolved
- subroschedrecoverytype
- subroclosedoutcome
- taccounttype
- transaction
- transactionlifecyclestate
- transactionstatus
- underwritingcompanytype

- `underwritinggrouptype`

  Note: Guidewire ClaimCenter does not use either `cc_tmp*` (temporary) tables or the `ccst_*` (staging) tables for reporting purposes.

Installing Guidewire ClaimCenter

Note: See the *ClaimCenter Installation Guide* for information on supported software configurations. Specifically, see "Installation Environments Overview" on page 12 in the *Installation Guide*.

Before you install Guidewire Standard Reporting, you must install Guidewire ClaimCenter 6.0.8:

- To acquire the Guidewire ClaimCenter application software, contact Guidewire Support.
- To install Guidewire ClaimCenter, follow the directions in *ClaimCenter Installation Guide*.

Do not proceed until you have successfully installed and tested ClaimCenter 6.0.8. At this point, verify that you are able to log into ClaimCenter and perform standard operations. However, at this time, you can not access any reports in the **Report** tab or perform reporting administration functions until you install and configure reporting properly.

IMPORTANT If you are upgrading Guidewire Standard Reporting (because you are upgrading Guidewire ClaimCenter), see "Upgrading Guidewire Standard Reporting" on page 18 for details of which sections to consult before proceeding.

Guidewire Standard Reporting Installation Scenarios

Guidewire Standard Reporting provides several different possible installation scenarios, depending on your development and production needs. Before beginning installation, first determine which scenario suits your particular purpose best. Then, go directly to that section.

Production Environment

Guidewire does **not** support the use of a stand-alone version of InetSoft Enterprise Manager, or the use of the InetSoft Report Designer in production environments.

Production scenario	See
ClaimCenter application with embedded InetSoft Enterprise Manager and no Report Designer	"Installing Standard Reporting for Production" on page 19

Development Environment

Guidewire only supports the use of the InetSoft Report Designer in a development environment.

Development scenario	See
The stand-alone InetSoft Report Designer exists on a local machine.	• "Installing InetSoft Style Report Enterprise for Development" on page 123 • "Working with the Report Designer" on page 127

Upgrading Guidewire Standard Reporting

If you are upgrading your version of ClaimCenter, especially if you have created custom reports, then do **not** follow the standard installation procedure for Guidewire Standard Reporting. Instead, consult the following table for the correct upgrade procedure.

Upgrading from....	See
ClaimCenter 4.x to ClaimCenter 6.x	“Upgrading from ClaimCenter 4.x to ClaimCenter 6.x” on page 43
ClaimCenter 5.x to ClaimCenter 6.x	“Upgrading from ClaimCenter 5.x to ClaimCenter 6.x” on page 59

chapter 2

Installing Standard Reporting for Production

This section describes how to acquire, install, and configure Guidewire ClaimCenter Standard Reporting for a *production* environment. This section assumes the following:

- You are installing Guidewire ClaimCenter 6.0.8 as the ClaimCenter application server.
- You are installing InetSoft StyleReport Enterprise Edition version 10.1 as the report server.

This topic includes:

IMPORTANT Do not use the information in this section to upgrade your installation of ClaimCenter Standard Reporting from ClaimCenter 4.x or 5.x to ClaimCenter 6.x. See "Upgrading Guidewire Standard Reporting" on page 18 for details.

Installation Checklist

Installing Guidewire Standard Reporting is a multistep process. The following list describes the various steps involved. Before you start the installation process, review “Guidewire Standard Reporting Installation Scenarios” on page 17.

Task	Description
Step 1: Acquire the InetSoft License Keys	You must have a valid InetSoft license key before you install and configure Guidewire Standard Reporting.
Step 2: Install the Standard Reporting Files	Install the Guidewire Standard Reporting files into a directory that is separate from the Guidewire ClaimCenter application directory.
Step 3: Create the Required Database Reporting Views	Guidewire Standard Reporting uses materialized database views. In general, Guidewire recommends that you do *not* use the same database for both ClaimCenter and Standard reporting, for performance reasons. See “Mirror Database Recommendations” on page 14 for more information.
Step 4: Install the Production Report Server	Install your chosen report server, using its installation instructions.
Step 5: Install Guidewire Standard Reporting	Install Guidewire Standard Reporting. You then need to configure both the reporting files and the ClaimCenter application files.
Step 6: Deploy the Reporting Files	Create and deploy a reporting `.war` or `.ear` file.
Step 7: Test the Reporting Configuration	After you complete your installation and configuration of ClaimCenter standard reporting, verify that you can log into the InetSoft Enterprise Manager administration console.

After you complete the checklist of steps, review the information in “Understanding InetSoft Configuration Options” on page 110. In most cases, you need do nothing with the configuration parameters accessible through the InetSoft Enterprise Manager. However, there are several that are important and which you must set correctly.

Also, review the material in “Administering Reports” on page 105. Especially review the following:

- “Working with ClaimCenter Reporting Permissions” on page 105
- “Synchronizing Reports with the InetSoft Server” on page 107
- “Working with the InetSoft Enterprise Manager” on page 109

Step 1: Acquire the InetSoft License Keys

Guidewire Support provides the following InetSoft license keys for users of ClaimCenter Standard Reporting:

- One license key for evaluation purposes
- Three license keys for development
- One license key for testing and QA
- One CPU license key for the report server

Evaluation License Key

Guidewire provides an evaluation license key for the sole purpose of evaluating the Guidewire reporting solution before you decide to deploy it. Refer to the ClaimCenter release notes that accompany your application installation for the evaluation license key.

Development License Keys

Guidewire provides three license keys for development use. These development license keys link to a specific version of the InetSoft software. You must obtain a new license key if you upgrade your version of the InetSoft software. To obtain new keys, you can register at the InetSoft Support Center to receive InetSoft software support. Then, log into the InetSoft Support Center and select **Upgrade Run Time Keys**.

Production License Keys

Guidewire provides two production (runtime) license keys: one for use with testing and QA, and the other for use with a runtime production server. As with the development license keys, you must upgrade your runtime license keys if the InetSoft software version changes.

IMPORTANT Only upgrade your version of the InetSoft software if instructed to do so by Guidewire service personnel.

Step 2: Install the Standard Reporting Files

Note: This step assumes that you have installed and configured Guidewire ClaimCenter. See "Installing Guidewire ClaimCenter" on page 17 for details.

You must install Guidewire Standard Reporting in a separate folder from that of Guidewire ClaimCenter. For example, if you install Guidewire ClaimCenter into a folder called ClaimCenter, then you can install the report files into a folder called ClaimCenter Reporting. You need separate ZIP files for the Guidewire application and the reporting files. Contact Guidewire Support for instructions on how to download the software.

After you acquire the report ZIP file (`CCReportConfig.zip`), unzip it to the target folder.

IMPORTANT Do not unzip the report files into your root directory. Instead, create a separate folder.

WARNING The build number of the reporting install files **must** match the build number of the Guidewire ClaimCenter installation. If the build numbers do not match, you will have problems with your reporting installation. To determine the build number of your Guidewire Standard Reporting installation, view `CCReportConfig.zip` → `resources` → `README.build###.txt`, for example: `README.build20090224.txt`.

After you install your application and reporting files, and configure them correctly, you then need to build a separate `.war` file for Guidewire ClaimCenter (`cc.war`) and Guidewire Standard Reporting (`sree.war`). You must deploy each `.war` file to a separate Web server (JVM).

Component	ZIP file	Target folder (suggested)	.war file	Deploy to
ClaimCenter	ClaimCenter.zip	ClaimCenter	cc.war	Application server
Standard Reporting	CCReportConfig.zip	ClaimCenterReporting	sree.war	Report server

Step 3: Create the Required Database Reporting Views

ClaimCenter reports pull information from specific database reporting views. In the base configuration, Guidewire provides a set of scripts that you run to create the reporting views. You can find these files in the following ClaimCenter reporting installation directory:

report-home`/resources/cc`

SQL Server database, single-byte format. Run the following scripts if the reporting SQL Server database stores all character data in single-byte format in `VARCHAR` columns:

- `cc_sqlserver_views.sql`
- `cc_sqlserver_storedprocedures.sql`

SQL Server database, multi-byte format. Run the following scripts if the reporting SQL Server database stores all character data in Unicode multi-byte format in `NVARCHAR` columns:

- `cc_sqlserver_views_unicode.sql`

- cc_sqlserver_storedprocedures_uncode.sql

IMPORTANT You only need to run the specialized Unicode multi-byte scripts if you use Microsoft SQL Server as the reporting database.

Note: See "Localization and the Reporting Database" on page 139. See also "Configuring SQL Server for ClaimCenter" on page 23 in the *Installation Guide* for a discussion of the <database> element in config.xml. Especially, see the discussion on how to set the unicodecolumns attribute on the <database> element. You need to set this attribute specifically for SQL Server databases that use a multi-byte format to store character data.

Oracle database. Run the following scripts if using the Oracle for the reporting database:

- cc_oracle_views.sql
- cc_oracle_storedprocedures.sql

Creating the ClaimCenter Reporting Views

Guidewire provides the following instructions for creating the necessary database views through scripts as examples only. Contact your database administrator for help if necessary. These instructions assume the use of a third-party tool such as Oracle SQL Developer for running the script to create the Oracle views.

IMPORTANT If this is a new ClaimCenter installation, it is possible that the tables ClaimCenter uses to create to create the reporting views do not exist yet. If they do not exist, you must start the ClaimCenter application server and allow it to run through its startup process. This operation creates the ClaimCenter tables needed to create the reporting views. Do not continue until you see these tables.

To create the necessary SQL Server 2008 database views

1. Open Microsoft SQL Server Management Studio.
2. Select the mirror reporting database and expand the **Tables** node.
3. Verify that the database contains a number of tables starting with dbo.cc_*. (If you have named your database schema differently—rather than dbo–use that name instead.)
4. In the Microsoft SQL Server Management Studio, run the following SQL scripts, in the listed order:

```
cc_sqlserver_views.sql
cc_sqlserver_storedprocedures.sql
```

5. Select the mirror reporting database from the **Database** tree list, and expand **Views**. Verify that you now have a number of views starting with dbo.ccrv_*. Do not proceed until these views are present.
6. Close all open query windows and exit SQL Server Management Studio.

IMPORTANT You must rerun the appropriate xx_sqlserver_views.sql script if you make a change to the data model configuration that affects a typelist used in ClaimCenter Standard Reporting. Specifically, you *must* run this script each and every time that you add a new language locale to your configuration.

To create the necessary Oracle database views

Note: You must have the Oracle database create type privilege to be able to create the necessary types, functions, and views using the report script.

1. Open Oracle SQL Developer and connect to the ClaimCenter mirror database.
2. Verify that the mirror reporting database contains the ClaimCenter application tables.

3. Run the following SQL scripts, in the listed order:

```
cc_oracle_views.sql
cc_oracle_storedprocedures.sql
```

4. Select the mirror reporting database, and expand **Views**. Verify that you now have a number of views starting with `ccrv_*`. Do not proceed until you see that these views are present.

5. Close all open query windows, and exit Oracle SQL Developer.

Step 4: Install the Production Report Server

IMPORTANT Guidewire requires the Windows operating system for *report development* as InetSoft only supports its Report Designer in a Windows environment. You can, however, use any Guidewire-supported combination of operating system and application server (used for Guidewire ClaimCenter) in a *reporting production environment*.

Guidewire supports the following for use as a report server:

- Apache Tomcat
- Oracle WebLogic
- IBM WebSphere

Guidewire supports the use of UNIX/Linux platforms for *production* report servers only. However, if you use one of these operating system, Guidewire Standard Reporting requires some specific setup to work correctly as a server-side application. Both of these platforms require that you start the Java Virtual machine (JVM) that supports reporting with an additional option:

```
-Djava.awt.headless=true
```

Even though Reporting on Linux does not use a visual environment, Linux still requires that you install visual environment components. Such packages correspond to the UNIX/Linux Xorg application. Thus, Reporting on Linux requires that you install additional Xorg packages. This includes the following:

- `expat`
- `fontconfig`
- `freetype`
- `xorg-x11-deprecated-libs`
- `xorg-x11-libs`
- `xorg-x11-Mesa-libGL`
- `zlib`

Note: Every Linux installation can differ and it is possible that you need to install more packages. Contact Guidewire Support for details.

Development environment. The following table summarizes the requirements for operating systems and report servers in a *development* environment:

Development OS	Tomcat	WebSphere	WebLogic
Windows	No change needed	No change needed	No change needed
Linux	Combination not supported	Combination not supported	Combination not supported
AIX	Combination not supported	Combination not supported	Combination not supported

Production environment. The following table summarizes the requirements for operating systems and report servers in a *production* environment:

Production OS	Tomcat	WebSphere	WebLogic
Windows	No change needed	No change needed	No change needed
Linux	`awt.headless = true` Xorg packages required	`awt.headless = true` Xorg packages required	`awt.headless = true` Xorg packages required
AIX	Combination not supported	`awt.headless = true`	`awt.headless = true`

Naming Guidelines

There are two important limitations on host names and database names:

- A host name cannot contain underscores.
- A database name cannot contain dashes.

Use of these characters can prevent the ClaimCenter application server from connecting to the reporting server. If you want to include your host name in your database name (or the reverse), you must translate the dashes in the host name into underscores in the database name. For example:

Host name	`this-is-your-host-name`
Database name	`this_is_your_host_name_db`

Reporting JVM Requirements

Guidewire ClaimCenter and Guidewire Standard Reporting must each run in its own server. If both servers exist on the same machine, each must run in its own JVM. This requires that you manage server ports correctly so as to avoid port conflicts between the two servers. For example, if you use two Tomcat servers, you can configure the ports as follows:

Server	Ports
Application	• Connector - 8080 • Server - 8005
Report	• Connector - 7070 • Server - 8006

Using Apache Tomcat as the Report Server

Guidewire supports Standard Reporting in *production* on the same platforms as Guidewire ClaimCenter. For the supported version of Apache Tomcat to use as a report server, see “Installation Environments Overview” on page 12 in the *Installation Guide*.

Install Apache Tomcat using its installation instructions.

See Also

- “Deploying Reporting to a Tomcat Server” on page 29

Using Oracle WebLogic as the Report Server

Guidewire supports Standard Reporting in *production* on the same platforms as Guidewire ClaimCenter. For the supported version of Oracle WebLogic to use as a report server, see “Installation Environments Overview” on page 12 in the *Installation Guide*.

Install Oracle WebLogic using its installation instructions.

See Also

- “Deploying Reporting to a WebLogic Server” on page 30

Using IBM WebSphere as the Report Server

Guidewire supports Standard Reporting in *production* on the same platforms as Guidewire ClaimCenter. For the supported version of IBM WebSphere to use as a report server, see “Installation Environments Overview” on page 12 in the *Installation Guide*.

Install IBM WebSphere using its installation instructions.

See Also

- “Deploying Reporting to a WebSphere Server” on page 31

Step 5: Install Guidewire Standard Reporting

Perform the following steps only after you have acquire the ClaimCenter reporting ZIP file and install it into its own directory. Before continuing, you must first perform the following steps:

- Step 1: Acquire the InetSoft License Keys
- Step 2: Install the Standard Reporting Files
- Step 3: Create the Required Database Reporting Views
- Step 4: Install the Production Report Server

To configure and deploy Guidewire Standard Reporting, you must perform a number of multi-part steps. These include:

- Set Up the Reporting Configuration Files
- Set Up the ClaimCenter Application Files

IMPORTANT To view reports with Guidewire ClaimCenter, you must also set the ClaimCenter reporting permissions correctly. After you complete the steps outlined in this section, review “Working with ClaimCenter Reporting Permissions” on page 105.

Set Up the Reporting Configuration Files

Perform the following steps in the Guidewire reporting directory, not the ClaimCenter directory.

IMPORTANT Ensure that you use a forward slash (/) **only** as the path deliminator in property files. Otherwise, the system does not interpret the path correctly.

Modify *build.properties*

Open *report-home*/cc/build.properties for editing and make the following modifications:

1. Set `build.wsdlurl` to the following:

   ```
   build.wsdlurl=http://hostname:port/cc/soap/ISREEAuthenticationAPI?wsdl
   ```

   This specifies a URL that points to the *ClaimCenter application server.* For example, this can be the following:

   ```
   http://MyAppServer:8580/cc/soap/ISREEAuthticationAPI?wsdl
   ```

   To test if your URL is correct, you can point your browser directly to this URL and verify that you can see the WSDL definition. The ClaimCenter application server must be up and running.

2. Set `build.sree_home` to the report server directory to which you intend to deploy the InetSoft `.war` file. For example, if you use Apache Tomcat as the report server, then set this value to *`tomcat-home`*`/webapps/sree`. If you install Tomcat on a Microsoft Windows machine at `C:\apache-tomcat-6.0.18`, then set this value to the following:

   ```
   C:/apache-tomcat-6.0.18/webapps/sree
   ```

   InetSoft uses this property to populate values in file `web.xml`. If you do not set this property correctly, it is possible that ClaimCenter cannot reach the web services on the InetSoft server.

Modify *sree.properties*

Open *`report-home`*`/cc/conf/sree.properties` for editing and modify the following `sree` properties:

- `license.key=###`

  Enter your license key. See "Step 1: Acquire the InetSoft License Keys" on page 20 for information on InetSoft license keys.

- `gw.url=http//:`*`hostname`*`:`*`port`*`/cc/soap/ISREEAuthenticationAPI`

  Enter the host name, port, and the Web name used in the Web application container for *Guidewire ClaimCenter*. The Web name (`app_context`) is the same as used in the URL for Guidewire ClaimCenter. For example, if the application server URL is:

  ```
  http://MyAppServer:8580/cc/ClaimCenter.do
  ```

  then, set `gw.url` to the following:

  ```
  http://MyReportServer:8580/cc/soap/ISREEAuthenticationAPI
  ```

- `APPURL=`*`hostname`*`:`*`port`*`/cc`

  The report templates (`.srt`) files use the value of the `APPURL` property to generate the correct server URL. See "Configuring Reports to Access Claim Information" on page 107 for more information.

- `scheduler.classpath`

  Enter the correct classpath for the Axis JAR files on `scheduler.classpath`. This assures that the correct version of the Axis classes loads before the InetSoft classes (`etools.jar` and `sree_pro.jar`).

  ```
  scheduler.classpath=$(sree.home);
                      $(sree.home)/../lib/axis.jar;
                      $(sree.home)/../lib/wsdl4j.jar;
                      $(sree.home)/../lib;
                      $(sree.home)/../lib/sree_pro.jar;
                      $(sree.home)/../lib/etools.jar;
                      $(sree.home)/../lib/ojdbc14.jar;
                      $(sree.home)/../lib/sqljdbc.jar;
                      $(sree.home)/../lib/gw-sree.jar
  ```

- `dependency.checker`

  Create the following entry:

  ```
  dependency.checker.interval=43200000
  ```

  The base configuration `sree.properties` files does not contain a dependency checker value. Thus, by default, the dependency checker executes twice a minute. This consumes a very large amount of CPU time in production. This value sets the check rate to once a day.

- `datasource.registry.file`

  Set this value to the correct database-specific location for `datasource.xml`. For example:

  ```
  $(sree.home)/sql/datasource.xml
  ```

- `query.registry.file`

  Set this value to the correct database-specific location for `query.xml`. For example:

  ```
  $(sree.home)/sql/query.xml
  ```

IMPORTANT If you want to use SOAP calls to access the InetSoft server from outside of Guidewire ClaimCenter, then you need to provide SOAP authentication credentials in `sree.properties`. See "SOAP Access from Outside Guidewire ClaimCenter" on page 38 for details.

Modify *datasource.xml*

Open *report-home*/cc/conf/datasource.xml for editing and make the following modifications to define the reporting database. Replace the variables as needed.

SQL Server:

```
<datasource name="ClaimCenter" type="jdbc">
  <ds_jdbc url="jdbc:sqlserver://hostname\sqlserver2008:port"
           driver="com.microsoft.sqlserver.jdbc.SQLServerDriver"
           requireLogin="true"
           transactionIsolation="-1"
           ansiJoin="false"
           defaultDB="report-database-name"
           user="database-user"
           password="database-password">
  </ds_jdbc>
</datasource>
```

Oracle:

```
<datasource name="ClaimCenter" type="jdbc">
  <ds_jdbc url="jdbc:oracle:thin:database-user/database-password@hostname:port:service-name"
           driver="oracle.jdbc.driver.OracleDriver"
           requireLogin="true"
           transactionIsolation="-1"
           ansiJoin="false"
           defaultDB="report-database-name"
           user="database-user"
           password="database-password">
  </ds_jdbc>
</datasource>
```

Note: For Oracle, you must set the `url` attribute on `<ds_jdbc>` with the database user name and database password. This must match the values that you set for the `user` and `password` attributes on `<ds_jdbc>` as well.

Set Up the ClaimCenter Application Files

Perform the following steps in the Guidewire application directory, not the reporting directory.

Modify ClaimCenter *config.xml* in Studio.

Within the Studio **Resources** tree, navigate to **Other Resources**:

1. Open `config.xml` for editing. This step automatically adds a `config.xml` file to the `modules/configuration/config` directory if one does not currently exist.
2. Add the following entry near the bottom of the file:

   ```
   <param name="StyleReportURL" value="http://hostname:port/sree"/>
   ```

   Enter the host name and port of the *load balancer server*. (See “Configuring Report Clustering” on page 76 for more information on load balancers.)

   For example, enter the following:

   ```
   <param name="StyleReportURL" value="http://LoadBalancerServer:7070/sree"/>
   ```

   If there is an existing report parameter in the file, comment it out.
3. Comment out the default development database and set the correct application production database. See the *ClaimCenter Installation Guide* for details, if necessary.

Build the ClaimCenter *cc.war* File

Build the ClaimCenter application cc.war file, and copy it to the application server.

1. Open a command prompt window and navigate to the application `bin` directory.
2. Enter the appropriate build command for your application server. For example, for Apache Tomcat, enter the following:

   ```
   gwcc build-war
   ```

Consult the *ClaimCenter Installation Guide*, if necessary.

3. Copy the generated `.war` file to the production application server.

IMPORTANT If you implement reporting with the Microsoft SQL Server database, see "Defining the jdbcURL" on page 46 in the *Installation Guide*. You must construct your `jdbcURL` database configuration parameter differently if you run Microsoft SQL Server with a server that uses a named instance.

Start Guidewire ClaimCenter

Start the ClaimCenter application server, then start Guidewire Studio. (To open Studio, navigate to the ClaimCenter root directory and double-click `studio.bat`.) You do not actually need to start the application server to start Guidewire Studio. However, you do need to have the application server running before you attempt to build the reporting `sree` file.

Modify *sree* Web Services in Studio.

Within the Studio **Resources** tree, navigate to **Web Services** → **sree**.

1. Click **Lookup Services**.
2. Select the **Enable** checkbox next to the **Override URL** field
3. Enter the URL to the endpoint of your web services calls in the **Override URL** field.

   The easiest way to accomplish this is to copy the `SoapRepository` URL from the **Actual URL** field to the **Override URL** field. Replace the default host name and port with the actual host name and port of the *report server* (or *load balancer server*, if using one).

   Use the following syntax:

   ```
   http://server:port/sree/services/SoapRepository.SoapRepositoryHttpSoap11Endpoint/
   ```

   For example:

   ```
   http://LoadBalancer:7070/sree/services/SoapRepository.SoapRepositoryHttpSoap11Endpoint/
   ```

4. Save your work.

You need to perform this step as the existing web service endpoint points to the location to which the WSDL points. The default URL points to the value described in the WSDL.

It is important to enter an override URL under the following circumstances:

- If you have not refreshed your web service using the Web Service editor **Refresh** button.
- If your reporting server URL changes between configuration time and testing or production.

If you use different production and development environments, Guidewire recommends that you use the **Web Services** settings and environments to configure these values. For example:

- For development, point to a reporting server on your local host.
- For testing, point to the testing report server (perhaps with a different configuration).
- For production, point to a load balancer.

Entering an override URL is essential if you plan on running your web service on different development and production environments. Typically, much of the development work is undertaken in a local development environment, without a load balancer. You use the override URL to access the `SoapRepository` of your local report server. In a production environment, it is likely that you have a different configuration. In that case, you want your **Override URL** value to point somewhere else, such as at a load balancer.

Note: For more information on working with Web Services in Guidewire Studio, see "Using the Web Services Editor" on page 145 in the *Configuration Guide*.

Step 6: Deploy the Reporting Files

After you install and configure Guidewire Standard Reporting, you must create a `.war` (`.ear`) file and deploy it to your production report server. See the following topics for details:

- Deploying Reporting to a Tomcat Server
- Deploying Reporting to a WebLogic Server
- Deploying Reporting to a WebSphere Server

Report URL

In the base configuration, Guidewire uses the following for the report URL. Replace the `machine` variable with the actual server name.

Server type	URL
Tomcat	`http://machine:7070/sree`
WebLogic	`http://machine:7001/sree`
WebSphere	`http://machine:9080/sree`

Deploying Reporting to a Tomcat Server

You must install Reporting in its own Tomcat instance. If necessary, change the port numbers to allow another Tomcat instance to run. If you are using Tomcat as the Guidewire application server, then use that same version as the report server. In any case, consult the hardware and software guidelines in the *ClaimCenter Installation Guide* for the most current information on supported software versions.

Example Tomcat Settings

The following assumes that you have Apache Tomcat 6.0.18 running as the Report Server in a separate Tomcat installation, in a Microsoft Windows environment.

In the following directory, edit the listed files:

```
C:\apache-tomcat-6.0.18\conf
```

server.xml

```
<Server port="8006" shutdown="SHUTDOWN">
<Connector port="7070" protocol="HTTP/1.1"
```

logging.properties

```
org.apache.tomcat.util.http.Parameters.level = SEVERE
```

Heap size. A maximum JVM heap size of 512M is generally sufficient for development environments. For production, Guidewire recommends that you do performance testing. *By default, Tomcat allocates 64M to a JVM instance during start-up.* This is insufficient for most needs, especially in a production environment. Setting the JVM heap size this low has the potential to disable the production server as reports driven by production data typically include huge amounts of data processing.

To create a sree.war file

1. Open a command prompt.
2. Navigate to report directory *`report-home`*`/cc`.
3. Enter the following command:

```
ant tomcat
```

This command builds a .war file (sree.war) that contains all the files required to run the ClaimCenter reports. It contains the following:

- JAR files
- Report content files
- Report configuration files

Upon completion of the ant tomcat command, the script places the newly created .war file in a new dist directory, in the following location:

report-home/cc/dist/sree.war

Useful Ant commands. Other useful Ant commands include the following:

Ant command	Description
ant clean	Deletes any files previously created by the build process. By default, the ant tomcat command first invokes ant clean.
ant help	Provides descriptions of the other available ant commands

To deploy reporting to Apache Tomcat

1. Create the sree.war file.
2. Remove any previous versions of the sree web application that exist. Thus, for Tomcat:
   - Delete the sree directory: *tomcat-home*/webapps/sree
   - Delete the sree.war file from the webapps directory: *tomcat-home*/webapps/sree.war
3. Copy *report-home*/dist/sree.war over to the report server file system. For Tomcat, copy it to the following location:

   tomcat-home/webapps
4. Start the Tomcat server.

IMPORTANT You must rebuild the .war file and redeploy if you make a change to config.xml. You must also rebuild and redeploy if you change sree.xml (the **sree** Web service).

Deploying Reporting to a WebLogic Server

Note: Guidewire supports Standard Reporting in *production* on the same platforms as Guidewire ClaimCenter. For the supported version of the WebLogic application to use as a report server, see “Installation Environments Overview” on page 12 in the *Installation Guide*.

After you install Guidewire Standard Reporting, you must create a sree.ear directory and deploy it to your WebLogic report server.

To create a WebLogic sree.ear

1. Start the ClaimCenter application server.
2. Open a command prompt window and navigate to the reporting install cc directory. For example:

   ```
   CCReporting/cc
   ```
3. Run the following command:

   ```
   ant weblogic
   ```

   This command creates the following sree.ear directory:

   ```
   CCReporting/cc/dist/sree.ear
   ```

This action creates a reporting sree.ear directory that you need to deploy to your production reporting server.

To deploy the WebLogic sree.ear

1. Copy the `sree.ear` directory to the WebLogic report server machine, to any temporary destination.
2. Start the WebLogic server.
3. Start the **Administration** console.
4. Click **Deployments** on the **Home Page**.
5. Under **Deployments**, click **Install**.
6. In the **Install Application Assistant**:
   a. Set the path for the `sree.ear` directory (for example `C:\TEMP\sree.ear`) and click **Next**.
   b. Select **Install this deployment as an application** and click **Next**.
   c. Under **Source accessibility**, select **Copy this application onto every target for me** and click **Finish**.
7. Select **sree** in the **Deployment** table.
8. Click **Start**.
9. Check to see if the server is up and running.

To access the InetSoft Enterprise Manager administration console

After the application starts, you can access the InetSoft Enterprise Edition administration console at the following location:

```
http://host:port/sree/EnterpriseManager
```

For example:

```
http://localhost:7001/sree/EnterpriseManager
```

In actual practice, you need to replace `localhost` with the actual production server name.

Deploying Reporting to a WebSphere Server

Note: Guidewire supports Standard Reporting in *production* on the same platforms as Guidewire ClaimCenter. For the supported version of IBM WebSphere to use as a report server, see “Installation Environments Overview” on page 12 in the *Installation Guide*.

After you install Guidewire Standard Reporting, you must create a WebSphere `sree.war` file and deploy it to your WebSphere report server.

To enable the visual environment components in WebSphere

1. Start WebSphere and open the administration console.
2. Navigate to the following location:

   Servers → **Application Servers** → **WebSphere application servers** → **server1** → **Server Infrastructure** → **Java and Process Management** → **Process Definition** → **Java Virtual Machine** → **Custom Properties**
3. Create a new custom property:

   Name: `java.awt.headless`

   Value: `true`

   Description: Setting to allow Guidewire Standard Reporting to work within WebSphere
4. Save your changes to the master configuration.
5. Restart the WebSphere server. The change does not go into effect until you restart the server.

To create a WebSphere sree.ear

1. Start the ClaimCenter application server.
2. Open a command prompt window and navigate to the reporting install `cc` directory. For example:
   ```
   CCReporting/cc
   ```
3. Run the following command:
   ```
   ant websphere
   ```
   This command creates the following `sree.war` file:
   ```
   CCReporting/cc/dist/sree.war
   ```

This action creates a reporting `sree.war` file that you need to deploy to your production reporting server.

To deploy the WebSphere sree.war file

After you install and configure Guidewire Standard Reporting, you must deploy your newly build `sree.war` file to your WebSphere report server. To do this, perform the following steps.

1. Start WebSphere and open the administration console.

   Note: Initially, you can use any name for the login, or you can leave the field blank.
2. Navigate to the following location:

   Applications → **Application Types** → **WebSphere enterprise applications** → **Enterprise Applications**
3. Click **Install**.
4. Browse and select `sree.war`.
5. Enter `sree` as the context root in the appropriate place in the wizard. This is the only option that you need to set in the deployment process. For all other options, accept the default.
6. Complete the installation.
7. Save your changes to the master configuration.

To manage the new sree application

Perform the following steps, or WebSphere generates a 500 error at the following location:

```
http://localhost:9080/sree/services/SoapRepository?wsdl
```

1. Navigate to **Applications** → **Application Types** → **WebSphere enterprise applications**.
2. Select the checkbox next to `sree.war` and click **Stop** to stop the server.
3. Click the `sree.war` link.
4. In the upper right section of the main panel, click **Manage Modules.**
5. Click `sree.war` again.
6. From the **Classes loader order** drop-down, select the **Classes loaded with local classloader first (parent last)** option.
7. Keep clicking **OK** until you see the **Enterprise Applications** screen again.
8. Click **Save**.
9. Select the checkbox next to `sree.war` and click **Start** to restart the server.

To access the InetSoft Enterprise Manager administration console

After the application starts, you can access the InetSoft Enterprise Edition administration console at the following location:

```
http://host:port/sree/EnterpriseManager
```

For example:

```
http://localhost:9080/sree/EnterpriseManager
```

In actual practice, you need to replace `localhost` with the actual production server name.

Step 7: Test the Reporting Configuration

Perform the following steps to test your reporting configuration.

1. **Start the report server.** Start the report server, and point a browser window to your report server to verify that you configured it correctly. Use the following syntax for the report server URL.

   ```
   http://report-server:port/sree
   ```

   If successful, you see the **InetSoft Examples Home** page, in which you can access the InetSoft Enterprise Manager. To log into the Enterprise Manager, you must have the ClaimCenter application server running because it manages the authentication for InetSoft.

   IMPORTANT Verify that both the application and report servers are up and operating correctly before continuing.

   ---

2. **Refresh sree Web Services in Studio.** Within Guidewire Studio, refresh the **sree** Web Service. (Click the green button next to the URL.) Studio now displays a list of APIs contained in the SoapRepository of the report server. You **must** refresh the **sree** Web Service before you continue. Verify that you can see the list of report APIs.

3. **Configure InetSoft Enterprise Manager.** Log into the InetSoft Enterprise Manager, navigate to **Server** → **Presentation** → **DHTML** → **Windows and Frames**, and uncheck the following:

   - **Create New Window from Repository**
   - **Create New Window when Drilldown**

   This causes the ClaimCenter reports to open within the **Report** tab in ClaimCenter itself, rather than a separate window.

4. **Verify report operation in ClaimCenter.** Log into Guidewire ClaimCenter under an administrative account. Verify that the report server is running by selecting the **Report** tab.

   - If ClaimCenter cannot access the report server, it generates a message to that effect.
   - If ClaimCenter can access the report server, you see a blank **Report** screen. The screen is blank because you need to set the report permissions in ClaimCenter before you can view reports.

   See “Working with ClaimCenter Reporting Permissions” on page 105 for information on setting permissions.

part II

Working with Guidewire Standard Reporting

chapter 3

Understanding Report Security

Guidewire ClaimCenter provides a robust security framework to handle secure communications between the ClaimCenter application and the InetSoft reporting server. Using this security framework, any user defined in the ClaimCenter system can administer or run reports—given the proper permissions. Guidewire also supports third-party security plugins for user authentication if you wish to use your own directory service.

This topic includes:

The Reporting Security Framework

In the basic flow of report integration, a ClaimCenter report administrator first synchronizes against the InetSoft report server and then retrieves all report templates (replets). ClaimCenter then saves these replets to the ClaimCenter database. Once saved, the report administrator can create report permission sets. These sets contain reports and can be assigned to roles. Based on their roles, different users can view and run different sets of reports.

SOAP Communication

Guidewire ClaimCenter and the InetSoft Report Server communicate using SOAP. To manage this communication, Guidewire supplies the following:

Item	Description
`sree`	Web service that enables Guidewire ClaimCenter to communicate with the InetSoft report server. Guidewire defines this Web Service as part of the base configuration platform. You can view and modify the Web Service in Guidewire Studio. The Web Service contains a URL pointing to the InetSoft report server's `SoapRepository.wsdl`. By using APIs defined in the `SoapRepository`, you can log into the report server and retrieve replets (report templates).

Item	Description
ISREEAuthenticationAPI	Platform-level API that enables the report server to communicate with a Guidewire application server. It is mainly responsible for user authentication.
GWSRUser	SOAP entity encapsulating report information pertaining to a user. It contains: • User name • User roles • User permissions • User's preferred language (The default is U.S. English.) • Group IDs for the groups of which the user is a supervisor or manager, including the child groups • Group IDs for the groups of which the user is a member Guidewire uses GWSRUser to pass information from ClaimCenter to the report server through SOAP communications.
GWPrincipal	Contains GWSRUser and is used with session.

SOAP Access from Outside Guidewire ClaimCenter

If you want to use SOAP calls to access the InetSoft server from outside of Guidewire ClaimCenter, then you need to provide SOAP authentication credentials in the InetSoft sree.properties file. Otherwise, you see the following message:

```
Unauthorized access please log in
```

You must set the following sree.property values to those for a Guidewire user with the reporting_admin permission:

- gw.soapuser.username
- gw.soapuser.userpassword

See Also

- "InetSoft Security Plugin" on page 39
- "Guidewire Authentication API" on page 39

Security Management

Guidewire provides security management features both on the InetSoft side and the ClaimCenter side. This includes:

- User Roles
- InetSoft Security Plugin
- Guidewire Authentication API

User Roles

InetSoft provides security management through the following user roles.

InetSoft	User
StyleReport	It is necessary for an admin user to exist in the InetSoft application. InetSoft uses this user to communicate with Guidewire ClaimCenter. If the admin user does not exist, InetSoft reports this as an error at server start-up and generates a SOAP exception.
Report Designer	The InetSoft Report Designer does not use a specific ClaimCenter user, but, instead, uses an anonymous user.

InetSoft Security Plugin

Guidewire provide a security plugin on the InetSoft side in the base configuration. This security plugin extends InetSoft's security framework. It uses the following Guidewire classes:

Class	Description
`GWAuthenticationProvider`	Authenticates the user's identity and report permission. • A user with the `reporting_admin` permission can log into both the InetSoft Enterprise Manager and the ClaimCenter **Report** tab. • A user with the `reporting_view` permission can only log into the **Report** tab. (ClaimCenter checks the report set permission at a later stage.)
`GWSecurityProvider`	Responsible for checking report group permissions for a given user. It authorizes access to a report based on the report set permission given to the user.

Guidewire Authentication API

Guidewire provide an authentication API on the Guidewire side in the base configuration:

Authentication API	Description
`ISREEAuthenticationAPI`	The InetSoft security plugin relies on this class for user authentication. It supports authentication with local and third-party authentication sources (such as an LDAP server).

The `ISREEAuthenticationAPI` API uses the following properties that you define in InetSoft file `sree.properties`:

Property	Description
`gw.soapuser.username`	The user name used by InetSoft to retrieve user, group, role information used by the Virtual Private Model and for authentication. **Note** The Virtual Private Model (VPM) is an optional InetSoft component that provides a reporting security layer. The VPM controls security for individual database tables. Thus, the VPM filters all database queries, regardless of whether someone manually creates a database query or whether the application automatically generates a database query from the data model.
`gw.soapuser.userpassword`	The associated password for the `gw.soapuser.username` property.
`gw.usercache.size`	The cache size to use for user cache, group cache, and role cache.
`gw.usercache.timeoutinseconds`	The number of seconds before the data in the cache becomes stale.

See Also

- "SOAP Communication" on page 37
- "SOAP Access from Outside Guidewire ClaimCenter" on page 38

Managing Reporting Log Ins

Within this security management framework, Guidewire handles the following types of log in operations:

Log in through	Description
InetSoft Enterprise Manager	Only a valid user with the `reporting_admin` permission can log into the Enterprise Manager.
Guidewire **Report** tab	Only a valid user with the `reporting_view` permission can log into the ClaimCenter **Report** tab and view reports. In addition, for a user to view a report, the report administrator must assign a report permission set to a role to which this user belongs.
SOAP API	A SOAP log-in call occurs during a **Sync Report** operation or if a user clicks on a report in ClaimCenter. Only a user with the `reporting_admin` permission can synchronize reports.

Report Security

Guidewire implements the following report security using the reporting security framework:

- Report Synchronization
- Report Permission Group Definition
- Report Permission Sets
- Report Viewing
- Report Viewing in InetSoft

Report Synchronization

Guidewire restricts the ability to synchronize reports against the InetSoft report server to users with the `reporting_admin` permission. Upon synchronization, ClaimCenter saves the synchronized reports into the ClaimCenter applications database, which also stores the information on the available reports. This preserves foreign keys between the reports and the roles in the event of incorrect SOAP results. As a consequence, this limits SOAP activity to the simple case in which an administrator wants to synchronize the local repository with StyleReport.

IMPORTANT If the you delete a report from the StyleReport side using the InetSoft Enterprise Manager, then you need to also delete it from ClaimCenter as well.

Report Permission Group Definition

Report permission sets control which reports a user can view. Just as with other ClaimCenter system permissions, you add the reporting permission to a role. Therefore, all users with the same role get the same report permissions. A user with the `reporting_admin` permission can create report permission sets from the **Report Admin** screen under the **Administration** tab. The report administrator can create and delete report permission sets. Each set can contain any number of reports.

Report Permission Sets

For a user to be able to view a report in the **Report** tab, you must:

1. Add that report to a report permission set.
2. Link that permission set to a specific role.
3. Assign that role to the user.

You manage report permission sets in the ClaimCenter **Administration** tab → **Report Admin** → **Report Admin** screen. Within this screen, you can add new report permission sets or modify existing report permission sets. You must explicitly add folders, subfolders, and reports to the report permission set. This means that selecting (checking) a folder does **not** automatically add all the child reports in the folder to the report permission set. You must also check any reports in that folder that you want to include in the report permission set. This is also true of subfolders. Selecting (checking) a parent folder does not automatically include all its subfolders in the report permission set. Again, you must explicitly check each subfolder that you want to include in the report permission set.

See Also

- “Working with ClaimCenter Reporting Permissions” on page 105
- “Synchronizing Reports with the InetSoft Server” on page 107

Report Viewing

A user with the `report_view` permission can view reports only. The report permission sets assigned to the user dictates what reports the user can view. The user accesses the reports from the **Report** tab. By default, ClaimCenter does not display drill-down reports even if they appear in the report permission set.

The base configuration **Dashboard** reports require the Group IDs of the current user. This includes:

- All groups for which the user is the supervisor or the manager *and* all the child groups of these groups.
- All groups of which the user is a member.

Report Viewing in InetSoft

If you have the requisite InetSoft login authority, you can view ClaimCenter reports directly within InetSoft itself. However, it is possible for InetSoft to display more reports than the login role has permission to view. This can happen if a user has permission to view not just a specific report, but the user also has permission to view that report's parent folder as well.

To ensure that InetSoft does not incorrectly display a particular report for a role, the report administrator must remove the specific report **and** its parent folder from that InetSoft role.

chapter 4

Upgrading from ClaimCenter 4.x to ClaimCenter 6.x

Upgrading your installation of ClaimCenter with Standard Reporting from release 4.x to 6.x requires that you also upgrade the InetSoft reporting server. The following table lists the required software versions:

Guidewire ClaimCenter	InetSoft StyleReport Enterprise Edition
release 4.x	version 8
release 6.x	version 10.1

Guidewire ClaimCenter 4.x implements the reporting data model used in version 8.0 of InetSoft StyleReport. ClaimCenter 6.x and InetSoft version 10.1 implement a *logical model* for report design. As a consequence, moving from version 8.0 to version 10.1 of the InetSoft software requires that you manually perform upgrade tasks specifically for reporting. This is true, especially if you have created customized reports.

This topic includes:

Starting the Upgrade Process

Installation of ClaimCenter 6.x Standard Reporting requires the installation of InetSoft StyleReport Enterprise Edition, version 10.1.

- If you have *not* made changes to the reports installed with the base ClaimCenter 4.x release, you need do very little to upgrade to ClaimCenter 6.x Standard Reporting. See "Upgrading from ClaimCenter 4.x to ClaimCenter 6.x with No Report Customization" on page 45 for the upgrade steps.

- If you *have* made changes to the base reporting structure or customized the reports installed with the base ClaimCenter 4.x release, you need to perform more extensive upgrade steps. See "Upgrading from ClaimCenter 4.x to ClaimCenter 6.x with Report Customization" on page 47 for the upgrade steps.

Before you start the reporting upgrade process, upgrade your base ClaimCenter installation to release 6.x and verify that it is working smoothly.

See Also

- For information on upgrading ClaimCenter itself, see the *ClaimCenter Upgrade Guide*.
- For information on the InetSoft report model, consult the InetSoft documentation. Specifically, refer to the following InetSoft documents:
  - InetSoft Technology *Report Designer Guide, version 10.1*
  - InetSoft Technology *Data Modeler Guide, version 10.1*

Install InetSoft Enterprise Edition (EE) version 10.1

IMPORTANT You only need to install the stand-alone version of the InetSoft software if you intend to use the InetSoft Report Designer. For more information, see "Working with the Report Designer" on page 127.

To install the InetSoft Report Designer, follow the installation instructions in "Working with the Report Designer" on page 127. You must obtain a new license key to install InetSoft StyleReport EE version 10.1. If you do not have the required license key, contact Guidewire Support.

If you do not intend to use the InetSoft Report Designer, then do not install the stand-alone version of this software. Instead, use the bundled version that Guidewire provides in the `CCReportConfig.zip` file.

Running Both InetSoft Report Designer v8 and v10

For some tasks related to upgrading reports, it is possible that you need to run both InetSoft Report Designer versions 8 and 10.1. You do not need to un-install InetSoft StyleReport EE version 8 to install InetSoft StyleReport EE version 10.1.

You must, however, configure separate repositories for their `.stylereport` files to run them simultaneously on one machine. By default, the `.stylereport` file resides in your user home directory. On a Windows machine, your user home directory is some variant of the following:

```
C:\Documents and Settings\UserName
```

The `.stylereport` file contains, among other items, the InetSoft license key for that particular version. Overwriting the file with an inappropriate license key causes the application to not work.

IMPORTANT Do not delete the InetSoft 8.0 application directories until you complete the upgrade process. You need certain files from the 8.0 directories for the upgrade to 10.1 if you customized any of your reports.

To prevent one version of the application from overwriting the other's `.stylereport` file, perform the following steps.

Launching Report Designer from the Start Menu

If you plan to launch Report Designer from the Windows **Start** menu, do the following:

1. Create two subdirectories in your user home directory, one for InetSoft version 8 and one for InetSoft version 10.1.
2. Install one version of the InetSoft software and move its `.stylereport` file to the appropriate folder.

3. Install the other version of the InetSoft software and move its `.stylereport` file to the other folder.
4. For each version of InetSoft, open file `InetSoft\bin\designer.lax` for editing and add the following line to each:

   ```
   lax.nl.java.option.additional=-Duser.home="file_path"
   ```

   In each, replace *file_path* with the correct directory path for that version's `.stylereport` file.

   Note: The last line of the `.lax` file must remain empty.

Launching from the command line. If you plan on starting the Report Designer from the command line, add the following Java option to the command:

```
-Duser.home="file_path"
```

Again, replace *file_path* with the desired directory. Include the double quotations marks as part of the command.

Upgrade InetSoft Charts

InetSoft has changed the engine that generates charts in the InetSoft version 10.1 release. InetSoft still generates charts that were created prior to this change. However, you can now consider these charts as deprecated. At some future time, InetSoft will no longer support these types of charts. Guidewire strongly encourages you to recreate these charts as part of your InetSoft upgrade process.

See Also

- For information on InetSoft charts and how to work with them, see the InetSoft Technology *Report Designer Guide, version 10.1*.

Upgrading from ClaimCenter 4.x to ClaimCenter 6.x with No Report Customization

IMPORTANT Perform the manual upgrade steps in this section if your installation of ClaimCenter 4.x Standard Reporting uses the reports provided in the ClaimCenter base configuration *only*. You must not have customized any of the base configuration reports.

You do not need to change any of the following if you did not customize any of the ClaimCenter base configuration reports or if you did not create new ones:

- Existing reporting views
- Existing ClaimCenter reports
- Existing ClaimCenter report permissions

The reports that Guidewire provides as part of the base ClaimCenter 6.x Standard Reporting installation simply replace the base configuration reports for ClaimCenter 4.x Standard Reporting.

After you have installed ClaimCenter 6.x and performed all upgrade steps listed in the *ClaimCenter Upgrade Guide*, you need to do the following to upgrade Standard Reporting:

- Remove Unneeded SQL Views from ClaimCenter 6.x Mirror Database
- Verify Existence of Necessary Mirror Database Tables
- Add Reports to Report Permission Sets

Remove Unneeded SQL Views from ClaimCenter 6.x Mirror Database

If the following SQL view exists in the ClaimCenter 6.x reporting database, remove it:

```
ccrpt_loss_run
```

Manually drop Oracle reporting views

IMPORTANT This is an issue that affects Oracle databases only. SQL Server does not use materialized views.

ClaimCenter 6.x on Oracle uses materialized views for reporting. This means that you must delete the previous reporting views from 4.x before creating the new reporting views in 6.x, to avoid a name duplication conflict in the database. To correct this issue, do *one* of the following:

- Remove the reporting views in the 4.x database before the upgrade.
- Delete the reporting views from the 6.x database after the upgrade, but before you run the 6.x view creation script.

Verify Existence of Necessary Mirror Database Tables

As with ClaimCenter 4.x Standard Reporting, Guidewire requires that you pull report data from a mirror of the ClaimCenter production database. For a list of the necessary reporting database reporting tables, see “Mirror Database Recommendations” on page 14. Do not proceed until you can see the required reporting database tables.

Add Reports to Report Permission Sets

Guidewire ClaimCenter did not use report permission sets in ClaimCenter 4.x for reporting. Instead, to give an individual user access to reporting in that release, you assigned that user a user role with specific reporting permissions. In ClaimCenter 6.x, these user roles are now permissions for roles. A report permission set in ClaimCenter 6.x is a group of reports that are assigned to a specific ClaimCenter role.

Guidewire provides the following report permission sets in the base ClaimCenter 6.x configuration:

Permission set	Assigned to
reportmanager	• Manager Reporting Admin Superuser
reportuser	• Claims Supervisor
viewaggclaimmetrics	• Claims Supervisor Manager Superuser
viewownmetricalerts	• Adjuster Claims Supervisor Manager Superuser
viewsupmetricalerts	• Claims Supervisor Manager Superuser

These report permission sets are empty by default. You need to add reports to these permission sets *after* you synchronize ClaimCenter with the report server. To enable a user with a specific role to view a report, you must:

- Add a report to a permission set.
- Assign that permission set to the specific user role.

You add reports to a permission set in the **Administration** → **Report Admin** screen. See “Working with ClaimCenter Reporting Permissions” on page 105 for information on how to add a report to a report permission set.

IMPORTANT You need to add reports to these permission sets *after* you synchronize ClaimCenter with the report server. Otherwise, you cannot see the reports to add.

Upgrading from ClaimCenter 4.x to ClaimCenter 6.x with Report Customization

IMPORTANT Perform the manual upgrade steps in this section *only* if your installation of ClaimCenter 4.x Standard Reporting contains base configuration reports that have been customized.

First, verify that you completed the upgrade steps listed in “Upgrading from ClaimCenter 4.x to ClaimCenter 6.x with No Report Customization” on page 45. Next, review the material in this section, as it is possible that you need to perform some or all of the listed upgrade steps. For example, if you have customized reports provided in the ClaimCenter base configuration (or added new reports), then you may need to perform additional upgrade steps. If you have changed any item in the following list, then perform the upgrade step associated with that report component:

If you changed or added	Upgrade	See section
Data model (in 4.x reports)	`datasource.xml`	“If You Created a Data Model in 4.x” on page 47
Queries	`query.xml`	“If You Customized or Added a Report” on page 50
Repository	`repository.xml`	“If You Customized or Added a Report” on page 50
Report templates	`*.srt`	“If You Customized or Added a Report” on page 50
Report data	SQL views	“If You Modified the Reporting SQL Views” on page 54
Reporting view environment	`sree.properties`	“If You Modified the Report Environment” on page 54
Localization	`sreeBundle_*.properties`	“If You Modified Report Localization Files” on page 55
Parameter sheets, entities	`stylereport.srl`	“If You Modified Parameter Sheets or Entities” on page 56
Worksheets	`asset.dat`	“If You Used Worksheets” on page 57

You only need to perform a particular upgrade step if you modified that particular reporting component. For example, if you modified a report query, then follow the upgrade steps listed for “If You Customized or Added a Report” on page 50.

IMPORTANT After you modify ClaimCenter Standard Reporting as described in the following sections, you must rebuild `sree.war` and redeploy it to the reporting server. See “Step 5: Install Guidewire Standard Reporting” on page 25 (especially step 1) for information on how to build a `sree.war` file.

If You Created a Data Model in 4.x

If you created a report that used the InetSoft data model for use with ClaimCenter Standard Reporting 4.x, then do the following:

1. Verify that the data model object names do not conflict with the object names used in the base configuration of ClaimCenter 6.x Standard Reporting.

2. Merge your modified ClaimCenter 4.x `datasource.xml` file with the ClaimCenter 6.x base configuration `datasource.xml` file. You can locate the ClaimCenter 6.x `datasource.xml` file in the following directory:

   report-home`/cc/conf/oracle`

   or

   report-home`/cc/conf/sql`

   The following example illustrates the merge process.

Example *datasource.xml* Merge Process

This example illustrates how to reconcile and merge multiple reporting `datasource.xml` files.

- The first `datasource.xml` file exists in your modified ClaimCenter 4.x Standard Reporting installation.
- The second `datasource.xml` file exists in your new installation of ClaimCenter 6.x Standard Reporting.

You must merge these two files and replace the existing file in your new installation of ClaimCenter 6.x Standard Reporting with the merged version.

The example uses the following XML files:

- Modified ClaimCenter 4.x Standard Reporting datasource.xml File
- Guidewire ClaimCenter Standard Reporting 6.x datasource.xml File
- Merged Version of the datasource.xml File

Modified ClaimCenter 4.x Standard Reporting *datasource.xml* File

```
<?xml version="1.0" encoding="UTF-8" ?>

<registry>
  <Version>6.0</Version>

  <datasource name="ClaimCenter" type="jdbc">
      <ds_jdbc url="jdbc:sqlserver://devdb2\SQLSERVER2008" ... </ds_jdbc>
  </datasource>

  <DataModel datasource="ClaimCenter">

    <LogicalModel name="Ext_cclmClaim" partition="Ext_ccpvClaim" datasource="ClaimCenter">
      <Description> <![CDATA[Logical model for claim metrics]]> </Description>

      <entity name="Loss Cause" x="-1" y="-1">
        <attribute name="Loss Cause" type="string" browse="true">
          <table> <![CDATA[dbo.ccrv_claim_metrics]]> </table>
            <column> <![CDATA[LossCause]]> </column>
          </attribute>
          ...

      </entity>
      ...

    </LogicalModel>
    ...

    <partition name="Ext_ccpvClaim">
      <table name="dbo.ccrv_claim_metrics" x="60" y="35"> </table>
    </partition>
    ...

    <vpms>
      <vpmObject class="inetsoft.uql.erm.VirtualPrivateModel">
        <name> <![CDATA[Ext_ccvpmClaimCenter]]> </name>
        <conditions>
          ...
        </conditions>
      </vpmObject>
    </vpms>
    ...
  </DataModel>

</registry>
```

Guidewire ClaimCenter Standard Reporting 6.x datasource.xml File

```
<?xml version="1.0" encoding="UTF-8" ?>

<registry>
  <Version>6.0</Version>

  <datasource name="ClaimCenter" type="jdbc">
    <ds_jdbc url="jdbc:sqlserver://devdb2\SQLSERVER2008" ... </ds_jdbc>
  </datasource>

  <DataModel datasource="ClaimCenter">

    <LogicalModel name="cclmExposure" partition="ccpvExposure" datasource="ClaimCenter">
      <Description> <![CDATA[Logical model for exposure metrics]]> </Description>

      <entity name="Organization" x="-1" y="-1">
        <attribute name="Exposure Group" type="string" browse="true">
          <table> <![CDATA[dbo.ccrv_exposure_metrics]]> </table>
          <column> <![CDATA[GroupName]]> </column>
        </attribute>
        ...

      </entity>
      ...

    </LogicalModel>
    ...

    <partition name="ccpvExposure">
      <table name="dbo.ccrv_exposure_metrics" x="50" y="36"> </table>
    </partition>
    ...

    <vpms>
      <vpmObject class="inetsoft.uql.erm.VirtualPrivateModel">
        <name> <![CDATA[ccvpmClaimCenter]]> </name>
        <conditions>
          ...
        </conditions>
      </vpmObject>
    </vpms>
   ...

  </DataModel>

</registry>
```

Merged Version of the datasource.xml File

Note: The following XML code shows merged entries from the modified `datasource.xml` file in bold font.

```
<?xml version="1.0" encoding="UTF-8" ?>

<registry>
  <Version>6.0</Version>

  <datasource name="ClaimCenter" type="jdbc">
    <ds_jdbc url="jdbc:sqlserver://devdb2\SQLSERVER2008" ... </ds_jdbc>
  </datasource>

  <DataModel datasource="ClaimCenter">

    <LogicalModel name="cclmExposure" partition="ccpvExposure" datasource="ClaimCenter">
      <Description> <![CDATA[Logical model for exposure metrics]]> </Description>
      <entity name="Organization" x="-1" y="-1">
        <attribute name="Exposure Group" type="string" browse="true">
          <table> <![CDATA[dbo.ccrv_exposure_metrics]]> </table>
          <column> <![CDATA[GroupName]]> </column>
        </attribute>
        ...

      </entity>
      ...

    </LogicalModel>

    <LogicalModel name="Ext_cclmClaim" partition="Ext_ccpvClaim" datasource="ClaimCenter">
      <Description> <![CDATA[Logical model for claim metrics]]> </Description>
      <entity name="Loss Cause" x="-1" y="-1">
```

```
      <attribute name="Loss Cause" type="string" browse="true">
        <table> <![CDATA[dbo.ccrv_claim_metrics]]> </table>
        <column> <![CDATA[LossCause]]> </column>
      </attribute>
      ...

    </entity>
    ...

  </LogicalModel>
  ...

  <partition name="ccpvExposure">
    <table name="dbo.ccrv_exposure_metrics" x="50" y="36"> </table>
  </partition>
  <partition name="Ext_ccpvClaim">
    <table name="dbo.ccrv_claim_metrics" x="60" y="35"> </table>
  </partition>
  ...

  <vpms>
    <vpmObject class="inetsoft.uql.erm.VirtualPrivateModel">
      <name> <![CDATA[ccvpmClaimCenter]]> </name>
        <conditions> </conditions>
    </vpmObject>
    <vpmObject class="inetsoft.uql.erm.VirtualPrivateModel">
      <name> <![CDATA[Ext_ccvpmClaimCenter]]> </name>
       <conditions> </conditions>
    </vpmObject>
  </vpms>
 </DataModel>
</registry>
```

If You Customized or Added a Report

To update custom reports that you created for ClaimCenter 4.x Standard Reporting, you must update these reports to use the InetSoft data model used for ClaimCenter 6.x Standard Reporting. This requires that you work with the following files:

File	Affects
`query.xml`	Queries
`repository.xml`	Repository
`*.srt`	Report templates

See the relevant section on how to handle each of the following cases:

- Set APPURL to Generate Correct Server URL for Drill-Down Links
- Rebuild Custom Report Bound to an Existing Query
- Rebuild Custom Report Bound to a New or Modified Query

Set APPURL to Generate Correct Server URL for Drill-Down Links

Many ClaimCenter reports provide hyper-text links (on the claim number) that you can click to access claim information directly within ClaimCenter itself. If this feature is active, clicking the link takes you directly to the relevant claim summary, financial or activities page. (This feature is only available in reports that contain claim numbers.)

The report templates (`.srt`) files now use the value of the (`sree.properties`) `APPURL` property to generate the correct server URL. For example, in the `Claim List.srt` file, you see the following:

```
<onLoad><![CDATA[parameter.serverURL = inetsoft.sree.SreeEnv.getProperty("APPURL");]]></onLoad>
```

If you have created custom reports that use the deprecated `beanLink` property to provide report drill-down functionality, then you must upgrade your report templates files to use the `APPURL` property. For example, replace `beanLink` in the `Claim Detail.srt` file:

```
<onLoad><![CDATA[parameter.serverURL = (beanLink['Message.Text']);]]></onLoad>
```

with the following:

```
<onLoad><![CDATA[parameter.serverURL = inetsoft.sree.SreeEnv.getProperty("APPURL");]]></onLoad>
```

Note: You only need to perform this step on reports that you created yourself and which use the deprecated `beanLink` property. Guidewire has corrected this issue for all base configuration reports for the ClaimCenter 6.0 release.

See Also

- "Configuring Reports to Access Claim Information" on page 107

Rebuild Custom Report Bound to an Existing Query

If you have created a new report *bound to an existing query*, then Guidewire recommends that you rebuild the report and bind it to the logical model. To proceed, you must have a copy of file `query.xml` from your ClaimCenter 4.x Standard Reporting installation. In a *development* environment, copy the `query.xml` file to the query registry location being used for ClaimCenter 6.x report development. You also need available both of the following:

- The ClaimCenter 4.x Standard Reporting queries
- The ClaimCenter Standard Reporting 6.x logical models

The following example illustrates the process of updating a custom report that is bound to an existing query.

1. Open the report in Report Designer version 10.1.
2. Right-click on the header portion of the table and select **Bind Data**.
3. On the **Data** tab, select the query to modify, then click **Edit**.
4. In the Data Modeler, click on the **Main Query** tab if parsed, else review the SQL String.
5. Determine which SQL views the query uses.
6. Determine the corresponding logical model.

   For example for the Subrogated query that uses `ccrv_claim_metrics`, the corresponding logical model would be `cclmClaim`.
7. Copy the table.

   For example, for the report **Subrogated Claims**, right-click on the table, select **Copy**, select the **Message** text, then select **Paste** from the right-click menu.
8. In the newly copied table, right-click and select **Bind Data**, and (within the **Data** tab) select the logical model to use (for example, `cclmClaim`).
9. In **Data Binding**, move the window down to expose the table layout in the Report Designer. Re-add the columns to recreate the table.

   For example, add `Organization.Claim Group`, `Organization.Claim Owner`, and similar items.
10. Click **Finish**.
11. Right-click on the original table and note the values in the **Columns** tab (formulas, sort order, and similar items). Repeat this task for the **Condition** tab, the **Grouping & Summary** tab, and the **Options** tab.
12. In the new table, add all of the values noted in the previous step.

    For example, on the **Condition** tab, enter `[Subrogation.Subrogation Status] [is] [one of] [$(SubroStatus)]`, and similar items.
13. In the original table, for each column, note the values for **Format**, **Alignment**, **Hyperlink**, and the other similar items.
14. In the new table, add the column properties.

15. Delete the original table.
16. Verify that the table ID matches the table reference in the Script for the Message text. If different, assign the table the ID of the original table. Right-click on the table, select **Properties**, then enter the correct ID.
17. After you complete this process, you must merge the report into the repository (`repository.xml`). See the following example for details.

Example repository.xml Merge Process

This example shows how to merge multiple reporting `repository.xml` files.

- The first `repository.xml` file exists in your modified ClaimCenter 4.x Standard Reporting installation.
- The second `repository.xml` file exists in your new installation of ClaimCenter 6.x Standard Reporting.

You must merge these two files and replace the existing file in your new installation of ClaimCenter 6.x Standard Reporting with the merged version.

The example uses the following XML files:

- Modified ClaimCenter 4.x Standard Reporting repository.xml File
- Guidewire ClaimCenter 6.x Standard Reporting repository.xml File
- Merged Version of the repository.xml File

Modified ClaimCenter 4.x Standard Reporting repository.xml File

```
<?xml version="1.0"  encoding="UTF-8"?>
<Registry>
  <Version>6.5</Version>
  <Replet name=".../LOB by Year" folder="false"
          template=".../Ext_LOB by Year.srt"
  </Replet>
  ...
</Registry>
```

Guidewire ClaimCenter 6.x Standard Reporting repository.xml File

```
<?xml version="1.0"  encoding="UTF-8"?>
<Registry>
  <Version>6.5</Version>
  <Replet name=".../First Payment Productivity" folder="false"
          template=".../First Payment Productivity.srt"
  </Replet>
  ...
</Registry>
```

Merged Version of the repository.xml File

Note: The following XML code shows merged entries from the modified `repository.xml` file in bold font.

```
<?xml version="1.0"  encoding="UTF-8"?>
<Registry>
  <Version>6.5</Version>
  <Replet name=".../First Payment Productivity" folder="false"
          template=".../First Payment Productivity.srt"
  </Replet>
  <Replet name=".../LOB by Year" folder="false"
          template=".../Ext_LOB by Year.srt"
  </Replet>
</Registry>
```

Rebuild Custom Report Bound to a New or Modified Query

If you have created a new report *bound to a new or modified query*, then Guidewire recommends that you rebuild the query. Or, if necessary, you might want to rebuild the report and bind it to the logical model as in the previous example. To rebuild the report, see “Rebuild Custom Report Bound to an Existing Query” on page 51.

The following example illustrates rebuilding the query.

1. In the Data Modeler, select the query (for example, Claim Detail).

2. Click on the **Main Query** tab if parsed. Otherwise, click on the **SQL String** tab.
3. If parsed, click on the **Fields** tab.
4. In the area **Query** fields, click on `dbo.ccrv_claim_metrics.JurisdictionState`.
5. Click on the left arrow icon (**Remove**).
6. In the area **Database** fields, click on `dbo.ccrv_claim_metrics.JurisdictionStateTypeCode`.
7. Click on the right arrow icon (**Add**).
8. With the newly added field highlighted in the area **Query** fields, click in the text box **Field Alias**.
9. Enter the value `JurisdictionState`.
10. If rebuilding the query, then merge the query into `query.xml`. See the following example for details.

Example query.xml Merge Process

This example illustrates how to merge multiple reporting `query.xml` files.

- The first `query.xml` file exists in your modified ClaimCenter 4.x Standard Reporting installation.
- The second `query.xml` file exists in your new installation of ClaimCenter 6.x Standard Reporting.

You must merge these two files and replace the existing file in your new installation of ClaimCenter 6.x Standard Reporting with the merged version.

The example uses the following XML files:

- Modified ClaimCenter 4.x Standard Reporting query.xml File
- Guidewire ClaimCenter 6.x Standard Reporting query.xml File
- Merged Version of the query.xml File

Modified ClaimCenter 4.x Standard Reporting query.xml File

```
<?xml version="1.0" encoding="UTF-8" ?>
<registry>
  <Version>4.4</Version>
  <query name="Claim Detail" type="jdbc" datasource="ClaimCenter">
    <query_jdbc>
      <uniform_sql parse="true">
        <table>
          <alias> <![CDATA[dbo.ccrv_claim_metrics]]> </alias>
          <name> <![CDATA[dbo.ccrv_claim_metrics]]> </name>
          ...
      <visible>true</visible>
    </query_jdbc>
  </query>
...
</registry>
```

Guidewire ClaimCenter 6.x Standard Reporting query.xml File

```
<?xml version="1.0" encoding="UTF-8" ?>
<registry>
  <Version>4.4</Version>
</registry>
```

Merged Version of the query.xml File

Note: The following XML code shows merged entries from the modified `query.xml` file in bold font.

```
<?xml version="1.0" encoding="UTF-8" ?>
<registry>
  <Version>4.4</Version>
  <query name="Claim Detail" type="jdbc" datasource="ClaimCenter">
    <query_jdbc>
      <uniform_sql parse="true">
        <table>
        <alias> <![CDATA[dbo.ccrv_claim_metrics]]> </alias>
        <name> <![CDATA[dbo.ccrv_claim_metrics]]> </name>
```

```
        ...
        <visible>true</visible>
      </query_jdbc>
    </query>
    ...
</registry>
```

If You Modified the Reporting SQL Views

The following table lists changes made to the columns for the mirror database views from ClaimCenter 4.x to ClaimCenter 6.x.

View	ClaimCenter 4.x columns	ClaimCenter 6.x columns
ccrpt_loss_run	ccrpt_loss_run PolEffDate IndemnityPaid ExpensePaid ClmStatus IndemnityIncd LineofBus PrmBodyPart DtlBodyPart	ccrv_loss_run PolicyEffectiveDate ClaimCostPmtAmt ExpensePmtAmt ClaimStatus ClaimCostIncd LOB
ccrv_claim	JurisdictionState	JurisdictionStateTypeCode JurisdictionState
ccrv_claim_metrics	JurisdictionState	JurisdictionStateTypeCode JurisdictionState
ccrv_exposure	DetailedBodyPartID DetailedBodyPart	
ccrv_exposure_metrics	DetailedBodyPartID DetailedBodyPart	

Verify that the SQL views used with ClaimCenter 4.x are valid still against the ClaimCenter 6.x database. In practice, Guidewire recommends strongly that you do **not** edit the base configuration reporting views. However, if you have modified them, then perform the following steps:

1. Retrieve the SQL script for the modified SQL view (for example, `ccrv_claim`).
2. Give the SQL script a new name (for example, `Ext_ccrv_claim`).
3. Change the dependencies (such as queries and reports) which relied on the modified SQL view to reference the newly named SQL view.

If You Modified the Report Environment

If you modified the report environment, you are most likely to have made changes to the following `sree.properties` file. The `sree.properties` file stores report viewer environment parameters. Typically, you make changes to this file through the InetSoft StyleReport **Enterprise Manager**. If you have modified any of the environment parameters in your 4.x installation, then you need to merge your two `sree.properties` files, which are:

- The modified file from ClaimCenter 4.x Standard Reporting
- The equivalent new file in ClaimCenter 6.x Standard Reporting

Compare the two `sree.properties` files. If necessary, make the required changes to the ClaimCenter 6.x Standard file manually.

Changed parameters. The following parameters in `sree.properties` are the most likely to have changed between releases.

- `format.date`
- `format.date.time`
- `html.export.button`
- `html.mail.button`

- `html.serverPrint.button`
- `log.output.stderr`
- `mail.smtp.host`

IMPORTANT Notice especially that property `format.datetime` is now `format.date.time`.

Reporting security model. Do **not** remove or modify the following property used by the reporting security model:

```
security.provider=com.guidewire.inetsoft.cc.GWSecurityProvider
```

Guidewire URL. Ensure that you set the value of property `gw.url` properly. Set this value to the location in which you intend to deploy your ClaimCenter server. For example, use some variant of the following:

```
gw.url=http//:<hostname>:<port>/cc/soap/ISREEAuthenticationAPI
```

If You Modified Report Localization Files

Note: For information on localization, see “Report Localization” on page 133. Also, see “Localizing Guidewire ClaimCenter” on page 463 in the *Configuration Guide*.

There are two general cases in which you need to modify localization files:

- **Case 1.** The installation of ClaimCenter 4.x Standard Reporting contains modifications to the base configuration `sreeBundle_en_US.properties` file.
- **Case 2.** The installation of ClaimCenter 4.x Standard Reporting contains a `sreeBundle_*.properties` file that you used to implement a different locale.

For the majority of the reports, localization now uses the attribute name in the logical model. For example, in the logical model `cclmActivity`, in the entity `Activity Attributes`, there is an attribute with the name `Activity Close Date`. To change the display value for this attribute, you would need to make the following entry in the ClaimCenter 6.x Standard Reporting `sreeBundle_en_US.properties` file:

```
Activity\ Close\ Date=Close Date
```

(Notice that in the file, a ‘\ ’ represents a space.)

The following table lists prefixes for the report elements:

Localization names for	Prefix	Example
Parameter sheet labels	`ps`	`psActivityGroup=Activity Group`
Report descriptions	`rd`	`rdActivityList=Report Description: Provides activity detail information`
Report labels	`lbl`	`lblActivityGroup=Activity Group`
Report titles	`rt`	`rtActivityList=Activity List`

Example Modification to sreeBundle_en_US.properties

Suppose that your installation of ClaimCenter 4.x Standard Reporting contains a modified `sreeBundle_en_US.properties` file (in the *`cc reporting app`*`/cc/conf` directory). During the upgrade, you must merge those changes with the `sreeBundle_en_US.properties` file that comes with the base configuration files for ClaimCenter 6.x Standard Reporting.

The example presents the following XML files:

- Existing ClaimCenter 4.x Standard Reporting sreeBundle_en_US.properties File
- Guidewire ClaimCenter 6.x Standard Reporting sreeBundle_en_US.properties File
- Merged sreeBundle_en_US.properties File

Existing ClaimCenter 4.x Standard Reporting sreeBundle_en_US.properties File

```
...
ClaimCostPmtAmt=Payment for Claim Cost
...
```

Guidewire ClaimCenter 6.x Standard Reporting sreeBundle_en_US.properties File

```
...
Claim\ Cost\ Payment\ Amount=Claim Cost Paid
...
ClaimCostPmtAmt=Claim Cost Paid
...
```

Merged sreeBundle_en_US.properties File

Note: The following XML code shows merged entries from the modified `query.xml` file in bold font.

```
...
Claim\ Cost\ Payment\ Amount=Payment for Claim Cost
...
ClaimCostPmtAmt=Payment for Claim Cost
...
```

Example of New sreeBundle_*.properties File

Suppose that your installation contains a `sreeBundle` file for another language. For example, Canadian French uses `sreeBundle_fr_CA.properties`. To upgrade your localized reports, do the following:

- Merge your `sreeBundle_fr_CA.properties` file with `sreeBundle_en_US.properties`, for the left side of the equation (localization Id).
- Rename the resulting file `sreeBundle_fr_CA.properties`.

The example presents the following XML files:

- Guidewire ClaimCenter 6.x Standard Reporting sreeBundle_en_US.properties File
- Existing ClaimCenter 4.x Standard Reporting sreeBundle_fr_CA.properties File
- Merged sreeBundle_fr_CA.properties File

Guidewire ClaimCenter 6.x Standard Reporting sreeBundle_en_US.properties File

```
...
Claim\ Cost\ Payment\ Amount=Claim Cost Paid
...
ClaimCostPmtAmt=Claim Cost Paid
...
```

Existing ClaimCenter 4.x Standard Reporting sreeBundle_fr_CA.properties File

```
...
ClaimCostPmtAmt=Versement pour Revendication Coutez
...
```

Merged sreeBundle_fr_CA.properties File

Note: The following XML code shows merged entries from the modified `query.xml` file in bold font.

```
...
Claim\ Cost\ Payment\ Amount=Versement pour Revendication Coutez
...
ClaimCostPmtAmt=Versement pour Revendication Coutez
...
```

If You Modified Parameter Sheets or Entities

File `stylereport.srl` contains information on parameter sheets and entities. It is an archive file. If you have created or modified any parameter sheets or entities from the base configuration for ClaimCenter 4.x Standard Reporting, then do the following:

1. Extract all of the items from the modified `stylereport.srl` in ClaimCenter 4.x Standard Reporting into a temporary directory.
2. In a different location, extract all of the items from the base configuration `stylereport.srl` in ClaimCenter 6.x Standard Reporting.

3. Add the modified items from ClaimCenter 4.x Standard Reporting to the items from ClaimCenter 6.x Standard Reporting.
4. Zip up the directory that contains all of the ClaimCenter 6.x Standard Reporting items and the modified items from ClaimCenter 4.x Standard Reporting.
5. Include this version of `sytlereport.srl` in the `sree.war` build process.

If You Used Worksheets

If you created worksheets in ClaimCenter Standard Reporting 4.x for new or existing reports, verify that these work sheets work correctly in ClaimCenter Standard Reporting 6.x. InetSoft stores worksheets in the `asset.dat` file. After verifying your worksheets, include `asset.dat` in the `sree.war` build process.

chapter 5

Upgrading from ClaimCenter 5.x to ClaimCenter 6.x

Upgrading your installation of ClaimCenter with Standard Reporting from release 5.x to 6.x requires that you also upgrade the InetSoft reporting server. The following table lists the required software versions:

Guidewire ClaimCenter	InetSoft StyleReport Enterprise Edition
release 5.x	version 9
release 6.x	version 10.1

This topic includes:

Starting the Upgrade Process

Installation of ClaimCenter 6.0.8 Standard Reporting requires the installation of InetSoft StyleReport Enterprise Edition, version 10.1.

- If you have *not* made changes to the reports installed with the base ClaimCenter 5.x release, you need do very little to upgrade to ClaimCenter 6.x Standard Reporting. See "Upgrading from ClaimCenter 5.x to ClaimCenter 6.x with No Report Customization" on page 61 for the upgrade steps.
- If you made changes to the base reporting structure or customized the reports installed with the base ClaimCenter 5.x release, you need to perform more extensive upgrade steps. See "Upgrading from ClaimCenter 5.x to ClaimCenter 6.x with Report Customization" on page 62 for the upgrade steps.

Before you start the reporting upgrade process, upgrade your base ClaimCenter installation to release 6.x and verify that it is working smoothly. See the *ClaimCenter Upgrade Guide* for information on upgrading ClaimCenter itself.

For information on the InetSoft report model, consult the InetSoft documentation. Specifically, refer to the following InetSoft documents:

- InetSoft Technology *Report Designer Guide, version 10.1*
- InetSoft Technology *Data Modeler Guide, version 10.1*

Install InetSoft Enterprise Edition (EE) version 10.1

To install the InetSoft reporting server for use with Guidewire Standard Reporting, follow the installation instructions in “Installing Standard Reporting for Production” on page 19. You must obtain a new license key to install InetSoft StyleReport EE version 10.1. If you do not have the required license key, contact Guidewire Support.

Running Both InetSoft Report Designer v9 and v10.1

IMPORTANT You only need to install the stand-alone version of the InetSoft software if you intend to use the InetSoft Report Designer. For more information, see “Working with the Report Designer” on page 127.

For some tasks related to upgrading reports, it is possible that you need to run both InetSoft Report Designer versions 9 and 10.1. You do not need to un-install InetSoft StyleReport EE version 9 to install InetSoft StyleReport EE version 10.1.

However, you must configure separate repositories for their `.stylereport` files to run them simultaneously on one machine. By default, the `.stylereport` file resides in your user home directory. On a Windows machine, your user home directory is some variant of the following:

```
C:\Documents and Settings\UserName
```

The `.stylereport` file contains, among other items, the InetSoft license key for that particular version. Overwriting the file with an inappropriate license key causes the application to not work.

IMPORTANT Do not delete the ClaimCenter InetSoft 9.0 application directories until you complete the upgrade process. You need certain files from the 9.0 directories for the upgrade to 10.1 if you customized any of your reports.

To prevent one version of the application from overwriting the other’s `.stylereport` file, perform the following steps.

Launching Report Designer from the Start Menu

If you plan to launch Report Designer from the Windows **Start** menu, do the following:

1. Create two subdirectories in your user home directory, one for InetSoft version 9 and one for InetSoft version 10.1.
2. Install one version of the InetSoft software and move its `.stylereport` file to the appropriate folder.
3. Install the other version of the InetSoft software and move its `.stylereport` file to the other folder.
4. For each version of InetSoft, open file `InetSoft\bin\designer.lax` for editing and add the following line to each:

```
lax.nl.java.option.additional=-Duser.home="file_path"
```

In each, replace *file_path* with the correct directory path for that version's `.stylereport` file.

Note: The last line of the `.lax` file must remain empty.

Launching from the command line. If you plan on starting the Report Designer from the command line, add the following Java option to the command:

```
-Duser.home="file_path"
```

Again, replace *file_path* with the desired directory. Include the double quotations marks as part of the command.

Upgrade InetSoft Charts

InetSoft has changed the engine that generates charts in the InetSoft version 10.1 release. InetSoft still generates charts that were created prior to this change. However, you can now consider these charts as deprecated. At some future time, InetSoft will no longer support these types of charts. Guidewire strongly encourages you to recreate these charts as part of your InetSoft upgrade process.

See Also

- For information on InetSoft charts and how to work with them, see the InetSoft Technology *Report Designer Guide, version 10.1*.

Upgrading from ClaimCenter 5.x to ClaimCenter 6.x with No Report Customization

IMPORTANT Perform the manual upgrade steps in this section if your installation of ClaimCenter 5.x Standard Reporting uses the reports provided in the ClaimCenter base configuration *only*. You must not have customized any of the base configuration reports.

You do not need to change any of the following if you did not customize any of the ClaimCenter base configuration reports or if you did not create new ones:

- Existing reporting views
- Existing ClaimCenter reports
- Existing ClaimCenter report permissions

The reports that Guidewire provides as part of the base ClaimCenter 6.x Standard Reporting installation simply replace the base configuration reports for ClaimCenter 5.x Standard Reporting.

After you have installed ClaimCenter 6.x and performed all upgrade steps listed in the *ClaimCenter Upgrade Guide*, you need to do the following to upgrade Standard Reporting:

- Remove Unneeded Tables from ClaimCenter 6.x Mirror Database
- Verify Existence of Necessary Mirror Database Tables
- Add Reports to Report Permission Sets

Remove Unneeded Tables from ClaimCenter 6.x Mirror Database

If any of the following tables exists in the ClaimCenter 6.x reporting database, remove them as Guidewire does not require them in the mirror reporting database:

- `cc_state`
- `cctl_ detailedbodyparttype`

Verify Existence of Necessary Mirror Database Tables

As with ClaimCenter 5.x Standard Reporting, Guidewire requires that you pull report data from a mirror of the ClaimCenter production database. For a list of the necessary reporting database tables, see “Mirror Database Recommendations” on page 14. Do not proceed until you can see the required reporting tables.

Add Reports to Report Permission Sets

Guidewire provides the following report permission sets in the base ClaimCenter 6.x configuration:

Permission set	Assigned to
reportmanager	• Manager Reporting Admin Superuser
reportuser	• Claims Supervisor
viewaggclaimmetrics	• Claims Supervisor Manager Superuser
viewownmetricalerts	• Adjuster Claims Supervisor Manager Superuser
viewsupmetricalerts	• Claims Supervisor Manager Superuser

These report permission sets are empty by default. You need to add reports to these permission sets *after* you synchronize ClaimCenter with the report server. To enable a user with a specific role to view a report, you must:

- Add a report to a permission set.
- Assign that permission set to a specific user role.

You add reports to a permission set in the **Administration** → **Report Admin** screen. See “Working with ClaimCenter Reporting Permissions” on page 105 for information on how to add a report to a report permission set.

IMPORTANT You need to add reports to these permission sets *after* you synchronize ClaimCenter with the report server. Otherwise, you cannot see the reports to add.

Upgrading from ClaimCenter 5.x to ClaimCenter 6.x with Report Customization

IMPORTANT Perform the manual upgrade steps in this section *only* if your installation of ClaimCenter 5.x Standard Reporting contains base configuration reports that have been customized.

First, verify that you completed the upgrade steps listed in “Upgrading from ClaimCenter 5.x to ClaimCenter 6.x with No Report Customization” on page 61. Next, review the material in this section, as it is possible that you need to perform some or all of the listed upgrade steps. For example, if you have customized reports provided in the ClaimCenter base configuration (or added new reports), then you may need to perform additional upgrade steps. If you have made changes, then perform the upgrade step associated with that report component:

If you changed or added	Upgrade	See
Report permission sets	cc/security-mapping.xml	“If You Added a Report Permission Set in 5.x” on page 63
Data model (in 5.x reports)	datasource.xml	“If You Created a Data Model in 5.x” on page 64

If you changed or added	Upgrade	See
Queries	`query.xml`	“If You Modified File query.xml” on page 66
Repository	`repository.xml`	“If You Modified File repository.xml” on page 67
Report templates	`*.srt`	“If You Customized or Added a Report” on page 68
Reporting view environment	`sree.properties`	“If You Modified the Report Environment” on page 69
Localization	`sreeBundle_*.properties`	“If You Modified Report Localization Files” on page 69
Parameter sheets, entities	`stylereport.srl`	“If You Modified Parameter Sheets or Entities” on page 71
Worksheets	`asset.dat`	“If You Used Worksheets” on page 71

You only need to perform a particular upgrade step if you modified that particular reporting component. For example, if you modified a report query, then follow the upgrade steps listed for “If You Modified File query.xml” on page 66.

IMPORTANT After you modify ClaimCenter Standard Reporting as described in the following sections, you must rebuild `sree.war` and redeploy it to the reporting server. See “Step 5: Install Guidewire Standard Reporting” on page 25 for information on how to build a `sree.war` file.

If You Added a Report Permission Set in 5.x

Guidewire provides the following report permission sets in the base configuration:

Permission set	Assigned to

`reportmanager`	• `Manager`
	`Reporting Admin`
	`Superuser`

`reportuser`	• `Claims Supervisor`

`viewaggclaimmetrics`	• `Claims Supervisor`
	`Manager`
	`Superuser`

`viewownmetricalerts`	• `Adjuster`
	`Claims Supervisor`
	`Manager`
	`Superuser`

`viewsupmetricalerts`	• `Claims Supervisor`
	`Manager`
	`Superuser`

To merge new permission sets into ClaimCenter 6.x

If you added one or more new permission sets to the ClaimCenter 5.x base configuration, then you must merge the new permission sets into ClaimCenter 6.x.

1. As part of your general upgrade from ClaimCenter 5.x to ClaimCenter 6.x, merge your report configuration changes in the 5.x `security-mapping.xml` file into the 6.x `security-mapping.xml` file. You can find this file in the following location:

   `<cc-home>/modules/cc/config/reporting`

   Copy this file to the `configuration` directory before making changes to it. *If you modify any other ClaimCenter application directory, you can invalidate your installation.*

   `<cc-home>/modules/configuration/config/reporting`

2. As part of your general upgrade from ClaimCenter 5.x to ClaimCenter 6.x, perform the standard database upgrade.

3. Verify the following:

- For every new permission set in the 5.x `security-mapping.xml` file, there exists the equivalent permission set in the 6.x `security-mapping.xml` file.
- For every permission set entry in the 6.x `security-mapping.xml` file, there exists an equivalent row entry for it in table `cc_reportgroup`.

Note: For information on working with report permissions, see "Working with ClaimCenter Reporting Permissions" on page 105.

If You Created a Data Model in 5.x

If you created a report that used the InetSoft data model for use with ClaimCenter Standard Reporting 5.x, then do the following:

1. Verify that the data model object names do not conflict with the object names used in the base configuration of ClaimCenter 6.x Standard Reporting.
2. Merge your modified ClaimCenter 5.x `datasource.xml` file with the ClaimCenter 6.x base configuration `datasource.xml` file. You can locate the ClaimCenter 6.x `datasource.xml` file in the following directory:

   report-home`/cc/conf/oracle`

   or

   report-home`/cc/conf/sql`

   The following example illustrates the merge process.

Example datasource.xml Merge Process

This example illustrates how to reconcile and merge multiple reporting `datasource.xml` files. The first `datasource.xml` file exists in your modified ClaimCenter 5.x Standard Reporting installation. The second `datasource.xml` file exists in your new installation of ClaimCenter 6.x Standard Reporting. You must merge these two files and replace the existing file in your new installation of ClaimCenter 6.x Standard Reporting with the merged version.

The example uses the following XML files:

- Modified ClaimCenter 5.x Standard Reporting datasource.xml File
- Guidewire ClaimCenter Standard Reporting 6.x datasource.xml File
- Merged Version of the datasource.xml File

Modified ClaimCenter 5.x Standard Reporting datasource.xml File

```
<?xml version="1.0" encoding="UTF-8" ?>

<registry>
  <Version>6.0</Version>

  <datasource name="ClaimCenter" type="jdbc">
      <ds_jdbc url="jdbc:sqlserver://devdb2\SQLSERVER2008" ... </ds_jdbc>
  </datasource>

  <DataModel datasource="ClaimCenter">

    <LogicalModel name="Ext_cclmClaim" partition="Ext_ccpvClaim" datasource="ClaimCenter">
      <Description> <![CDATA[Logical model for claim metrics]]> </Description>

      <entity name="Loss Cause" x="-1" y="-1">
        <attribute name="Loss Cause" type="string" browse="true">
          <table> <![CDATA[dbo.ccrv_claim_metrics]]> </table>
            <column> <![CDATA[LossCause]]> </column>
          </attribute>
          ...

      </entity>
      ...

    </LogicalModel>
    ...
```

```
    <partition name="Ext_ccpvClaim">
      <table name="dbo.ccrv_claim_metrics" x="60" y="35"> </table>
    </partition>
    ...

    <vpms>
      <vpmObject class="inetsoft.uql.erm.VirtualPrivateModel">
        <name> <![CDATA[Ext_ccvpmClaimCenter]]> </name>
        <conditions>
          ...
        </conditions>
      </vpmObject>
    </vpms>
    ...
  </DataModel>

</registry>
```

Guidewire ClaimCenter Standard Reporting 6.x datasource.xml File

```
<?xml version="1.0" encoding="UTF-8" ?>

<registry>
  <Version>6.0</Version>

  <datasource name="ClaimCenter" type="jdbc">
    <ds_jdbc url="jdbc:sqlserver://devdb2\SQLSERVER2008" ... </ds_jdbc>
  </datasource>

  <DataModel datasource="ClaimCenter">

    <LogicalModel name="cclmExposure" partition="ccpvExposure" datasource="ClaimCenter">
      <Description> <![CDATA[Logical model for exposure metrics]]> </Description>

      <entity name="Organization" x="-1" y="-1">
        <attribute name="Exposure Group" type="string" browse="true">
          <table> <![CDATA[dbo.ccrv_exposure_metrics]]> </table>
          <column> <![CDATA[GroupName]]> </column>
        </attribute>
        ...

      </entity>
      ...

    </LogicalModel>
    ...

    <partition name="ccpvExposure">
      <table name="dbo.ccrv_exposure_metrics" x="50" y="36"> </table>
    </partition>
    ...

    <vpms>
      <vpmObject class="inetsoft.uql.erm.VirtualPrivateModel">
        <name> <![CDATA[ccvpmClaimCenter]]> </name>
        <conditions>
          ...
        </conditions>
      </vpmObject>
    </vpms>
   ...

  </DataModel>

</registry>
```

Merged Version of the datasource.xml File

Note: The following XML code shows merged entries from the modified `datasource.xml` file in bold font.

```
<?xml version="1.0" encoding="UTF-8" ?>

<registry>
  <Version>6.0</Version>

  <datasource name="ClaimCenter" type="jdbc">
    <ds_jdbc url="jdbc:sqlserver://devdb2\SQLSERVER2008" ... </ds_jdbc>
  </datasource>
```

```
<DataModel datasource="ClaimCenter">

  <LogicalModel name="cclmExposure" partition="ccpvExposure" datasource="ClaimCenter">
    <Description> <![CDATA[Logical model for exposure metrics]]> </Description>
    <entity name="Organization" x="-1" y="-1">
      <attribute name="Exposure Group" type="string" browse="true">
        <table> <![CDATA[dbo.ccrv_exposure_metrics]]> </table>
        <column> <![CDATA[GroupName]]> </column>
      </attribute>
      ...

    </entity>
    ...

  </LogicalModel>

  <LogicalModel name="Ext_cclmClaim" partition="Ext_ccpvClaim" datasource="ClaimCenter">
    <Description> <![CDATA[Logical model for claim metrics]]> </Description>
    <entity name="Loss Cause" x="-1" y="-1">
      <attribute name="Loss Cause" type="string" browse="true">
        <table> <![CDATA[dbo.ccrv_claim_metrics]]> </table>
        <column> <![CDATA[LossCause]]> </column>
      </attribute>
      ...

    </entity>
    ...

  </LogicalModel>
  ...

  <partition name="ccpvExposure">
    <table name="dbo.ccrv_exposure_metrics" x="50" y="36"> </table>
  </partition>
  <partition name="Ext_ccpvClaim">
    <table name="dbo.ccrv_claim_metrics" x="60" y="35"> </table>
  </partition>
  ...

  <vpms>
    <vpmObject class="inetsoft.uql.erm.VirtualPrivateModel">
      <name> <![CDATA[ccvpmClaimCenter]]> </name>
        <conditions> </conditions>
    </vpmObject>
    <vpmObject class="inetsoft.uql.erm.VirtualPrivateModel">
      <name> <![CDATA[Ext_ccvpmClaimCenter]]> </name>
       <conditions> </conditions>
    </vpmObject>
  </vpms>
 </DataModel>
</registry>
```

If You Modified File query.xml

If you modified file `query.xml`, then use the following example to map out the necessary merge changes.

Example query.xml Merge Process

This example illustrates how to merge multiple reporting `query.xml` files.

- The first `query.xml` file exists in your modified ClaimCenter 5.x Standard Reporting installation.
- The second `query.xml` file exists in your new installation of ClaimCenter 6.x Standard Reporting.

You must merge these two files and replace the existing file in your new installation of ClaimCenter 6.x Standard Reporting with the merged version.

The example uses the following XML files:

- Modified ClaimCenter 5.x Standard Reporting query.xml File
- Guidewire ClaimCenter 6.x Standard Reporting query.xml File
- Merged Version of the query.xml File

Modified ClaimCenter 5.x Standard Reporting query.xml File

```
<?xml version="1.0" encoding="UTF-8" ?>
<registry>
  <Version>4.4</Version>
  <query name="Claim Detail" type="jdbc" datasource="ClaimCenter">
    <query_jdbc>
      <uniform_sql parse="true">
        <table>
          <alias> <![CDATA[dbo.ccrv_claim_metrics]]> </alias>
          <name> <![CDATA[dbo.ccrv_claim_metrics]]> </name>
          ...
      <visible>true</visible>
    </query_jdbc>
  </query>
...
</registry>
```

Guidewire ClaimCenter 6.x Standard Reporting query.xml File

```
<?xml version="1.0" encoding="UTF-8" ?>
<registry>
  <Version>4.4</Version>
</registry>
```

Merged Version of the query.xml File

Note: The following XML code shows merged entries from the modified `query.xml` file in bold font.

```
<?xml version="1.0" encoding="UTF-8" ?>
<registry>
  <Version>4.4</Version>
  <query name="Claim Detail" type="jdbc" datasource="ClaimCenter">
    <query_jdbc>
      <uniform_sql parse="true">
        <table>
        <alias> <![CDATA[dbo.ccrv_claim_metrics]]> </alias>
        <name> <![CDATA[dbo.ccrv_claim_metrics]]> </name>
        ...
      <visible>true</visible>
    </query_jdbc>
  </query>
  ...
</registry>
```

If You Modified File repository.xml

If you modified file `repository.xml`, then use the following example to map out the necessary merge changes.

Example repository.xml Merge Process

This example shows how to merge multiple reporting `repository.xml` files.

- The first `repository.xml` file exists in your modified ClaimCenter 5.x Standard Reporting installation.
- The second `repository.xml` file exists in your new installation of ClaimCenter 6.x Standard Reporting.

You must merge these two files and replace the existing file in your new installation of ClaimCenter 6.x Standard Reporting with the merged version.

The example uses the following XML files:

- Modified ClaimCenter 5.x Standard Reporting repository.xml File
- Guidewire ClaimCenter 6.x Standard Reporting repository.xml File
- Merged Version of the repository.xml File

Modified ClaimCenter 5.x Standard Reporting repository.xml File

```
<?xml version="1.0"  encoding="UTF-8"?>
<Registry>
  <Version>6.5</Version>
  <Replet ... name="/LOB by Year" folder="false"
          ... template="/Ext_LOB by Year.srt"
  </Replet>
```

```
  ...
</Registry>
```

Guidewire ClaimCenter 6.x Standard Reporting repository.xml File

```
<?xml version="1.0"  encoding="UTF-8"?>
<Registry>
  <Version>6.5</Version>
  <Replet ... name="/First Payment Productivity" folder="false"
          ... template="/First Payment Productivity.srt"
  </Replet>
  ...
</Registry>
```

Merged Version of the repository.xml File

Note: The following XML code shows merged entries from the modified `repository.xml` file in bold font.

```
<?xml version="1.0"  encoding="UTF-8"?>
<Registry>
  <Version>6.5</Version>
  <Replet ... name="/First Payment Productivity" folder="false"
          ... template="/First Payment Productivity.srt"
  </Replet>
  <Replet ... name="/LOB by Year" folder="false"
          ... template="/Ext_LOB by Year.srt"
  </Replet>
</Registry>
```

If You Customized or Added a Report

To update custom reports that you created for ClaimCenter 5.x Standard Reporting, you must update these reports to use the InetSoft data model used for ClaimCenter 6.x Standard Reporting. See the following:

- Set APPURL to Generate Correct Server URL for Drill-Down Links

Set APPURL to Generate Correct Server URL for Drill-Down Links

Many ClaimCenter reports provide hyper-text links (on the claim number) that you can click to access claim information directly within ClaimCenter itself. If this feature is active, clicking the link takes you directly to the relevant claim summary, financial or activities page. (This feature is only available in reports that contain claim numbers.)

The report templates (`.srt`) files now use the value of the (`sree.properties`) `APPURL` property to generate the correct server URL. For example, in the `Claim List.srt` file, you see the following:

```
<onLoad><![CDATA[parameter.serverURL = inetsoft.sree.SreeEnv.getProperty("APPURL");]]></onLoad>
```

If you have created custom reports that use the deprecated `beanLink` property to provide report drill-down functionality, then you must upgrade your report templates files to use the `APPURL` property. For example, replace `beanLink` in the `Claim Detail.srt` file:

```
<onLoad><![CDATA[parameter.serverURL = (beanLink['Message.Text']);]]></onLoad>
```

with the following:

```
<onLoad><![CDATA[parameter.serverURL = inetsoft.sree.SreeEnv.getProperty("APPURL");]]></onLoad>
```

Note: You only need to perform this step on reports that you created yourself and which use the deprecated `beanLink` property. Guidewire has corrected this issue for all base configuration reports for the ClaimCenter 6.0 release.

See Also

- “Configuring Reports to Access Claim Information” on page 107

If You Modified the Report Environment

If you modified the report environment, you are most likely to have made changes to the following:

- The Sree Properties File
- The Build Properties File

The Sree Properties File

The `sree.properties` file stores report viewer environment parameters. Typically, you make changes to this file through the InetSoft StyleReport **Enterprise Manager**. If you have modified any of the environment parameters in your 5.x installation, then you need to merge your two `sree.properties` files, which are:

- The modified file from ClaimCenter 5.x Standard Reporting
- The equivalent new file in ClaimCenter 6.x Standard Reporting

Compare the two `sree.properties` files. If necessary, make the required changes to the ClaimCenter 6.x Standard file manually.

Changed parameters. The following parameters in `sree.properties` are the most likely to have changed between releases.

- `format.date`
- `format.date.time`
- `html.export.button`
- `html.mail.button`
- `html.serverPrint.button`
- `log.output.stderr`
- `mail.smtp.host`

IMPORTANT Notice especially that property `format.datetime` is now `format.date.time`.

Reporting security model. Do **not** remove or modify the following property used by the reporting security model:

```
security.provider=com.guidewire.inetsoft.cc.GWSecurityProvider
```

Guidewire URL. Ensure that you set the value of property `gw.url` properly. Set this value to the location in which you intend to deploy your ClaimCenter server. For example, use some variant of the following:

```
gw.url=http//:<hostname>:<port>/cc/soap/ISREEAuthenticationAPI
```

The Build Properties File

The *report-home*`/cc/build.properties` file contains an additional property that you need to set correctly.

Sree home. Guidewire has added a new property that InetSoft uses to populate values in `web.xml`. It is called `build.sree_home`. You set this value to the directory path in which you intend to install the `sree.war` file. For example, if you are using Apache Tomcat as the reporting server, then use some variant of the following:

```
build.sree_home=C:/apache-tomcat-6.0.14/webapps/sree
```

See “Set Up the Reporting Configuration Files” on page 25 for more information on the reporting parameter.

If You Modified Report Localization Files

Note: For information on localization, see “Report Localization” on page 133. Also, see “Localizing Guidewire ClaimCenter” on page 463 in the *Configuration Guide*.

There are two general cases in which you need to modify localization files:

- **Case 1.** The installation of ClaimCenter 5.x Standard Reporting contains modifications to the base configuration `sreeBundle_en_US.properties` file.

- **Case 2.** The installation of ClaimCenter 5.x Standard Reporting contains a `sreeBundle_*.properties` file that you used to implement a different locale.

For the majority of the reports, localization uses the attribute name in the logical model. For example, in the logical model `cclmActivity`, in the entity `Activity Attributes`, there is an attribute with the name `Activity Close Date`. To change the display value for this attribute, you would need to make the following entry in the ClaimCenter 5.x Standard Reporting `sreeBundle_en_US.properties` file:

```
Activity\ Close\ Date=Close Date
```

(Notice that in the file, a '\ ' represents a space.)

The following table lists prefixes for the report elements:

Localization names for	Prefix	Example
Parameter sheet labels	`ps`	`psActivityGroup=Activity Group`
Report descriptions	`rd`	`rdActivityList=Report Description: Provides activity detail information`
Report labels	`lbl`	`lblActivityGroup=Activity Group`
Report titles	`rt`	`rtActivityList=Activity List`

Example Modification to sreeBundle_en_US.properties

Suppose that your installation of ClaimCenter 5.x Standard Reporting contains a modified `sreeBundle_en_US.properties` file (in the *report-home*`/cc/conf` directory). During the upgrade, you must merge those changes with the `sreeBundle_en_US.properties` file that comes with the base configuration files for ClaimCenter 6.x Standard Reporting.

The example presents the following XML files:

- Existing ClaimCenter 5.x Standard Reporting sreeBundle_en_US.properties File
- Guidewire ClaimCenter 6.x Standard Reporting sreeBundle_en_US.properties File
- Merged sreeBundle_en_US.properties File

Existing ClaimCenter 5.x Standard Reporting sreeBundle_en_US.properties File

```
...
ClaimCostPmtAmt=Payment for Claim Cost
...
```

Guidewire ClaimCenter 6.x Standard Reporting sreeBundle_en_US.properties File

```
...
Claim\ Cost\ Payment\ Amount=Claim Cost Paid
...
ClaimCostPmtAmt=Claim Cost Paid
...
```

Merged sreeBundle_en_US.properties File

Note: The following XML code shows merged entries from the modified `query.xml` file in bold font.

```
...
Claim\ Cost\ Payment\ Amount=Payment for Claim Cost
...
ClaimCostPmtAmt=Payment for Claim Cost
...
```

Example of New sreeBundle_*.properties File

Suppose that your installation contains a `sreeBundle` file for another language. For example, French (France) uses `sreeBundle_fr_FR.properties`. To upgrade your localized reports, do the following:

- Merge your `sreeBundle_fr_FR.properties` file with `sreeBundle_en_US.properties`, for the left side of the equation (localization Id).

- Rename the resulting file `sreeBundle_fr_FR.properties`.

The example presents the following XML files:

- Existing ClaimCenter 5.x Standard Reporting sreeBundle_fr_FR.properties File
- Guidewire ClaimCenter 6.x Standard Reporting sreeBundle_en_US.properties File
- Merged sreeBundle_fr_FR.properties File

Existing ClaimCenter 5.x Standard Reporting sreeBundle_fr_FR.properties File

```
...
ClaimCostPmtAmt=Versement pour Revendication Coutez
...
```

Guidewire ClaimCenter 6.x Standard Reporting sreeBundle_en_US.properties File

```
...
Claim\ Cost\ Payment\ Amount=Claim Cost Paid
...
ClaimCostPmtAmt=Claim Cost Paid
...
```

Merged sreeBundle_fr_FR.properties File

Note: The following XML code shows merged entries from the modified `query.xml` file in bold font.

```
...
Claim\ Cost\ Payment\ Amount=Versement pour Revendication Coutez
...
ClaimCostPmtAmt=Versement pour Revendication Coutez
...
```

If You Modified Parameter Sheets or Entities

File `stylereport.srl` contains information on parameter sheets and entities. It is an archive file. If you have created or modified any parameter sheets or entities from the base configuration for ClaimCenter 5.x Standard Reporting, then do the following:

1. Extract all of the items from the modified `stylereport.srl` in ClaimCenter 5.x Standard Reporting into a temporary directory.
2. In a different location, extract all of the items from the base configuration `stylereport.srl` in ClaimCenter 6.x Standard Reporting.
3. Add the modified items from ClaimCenter 5.x Standard Reporting to the items from ClaimCenter 6.x Standard Reporting.
4. Zip up the directory that contains all of the ClaimCenter 6.x Standard Reporting items and the modified items from ClaimCenter 5.x Standard Reporting.
5. Include this version of `sytlereport.srl` in the `sree.war` build process.

If You Used Worksheets

If you created worksheets in ClaimCenter Standard Reporting 5.x for new or existing reports, verify that these work sheets work correctly in ClaimCenter Standard Reporting 6.x. InetSoft stores worksheets in the `asset.dat` file. After verifying your worksheets, include `asset.dat` in the `sree.war` build process.

part III

Working with Reports

chapter 6

Reporting in a Clustered Environment

Guidewire requires that you configure Guidewire ClaimCenter and Guidewire Standard Reporting in a clustered configuration with a load balancer to improve speed and performance. Guidewire provides the following example for setting up a ClaimCenter cluster only as an illustration of the configuration involved. You must modify this example to meet your business needs.

This topic includes:

For more information:

- For general information on setting up Guidewire ClaimCenter in a clustered environment, see "Managing Clustered Servers" on page 81 in the *System Administration Guide*.
- For information on setting up and configuring multiple InetSoft Report Repositories, refer to the InetSoft documentation, specifically the InetSoft *Style Report Administration Reference Guide*.

Guidewire Standard Reporting and Load-Balancing

Guidewire requires the use of a load-balancer if you use Guidewire Standard Reporting with Guidewire ClaimCenter. This applies to the minimal configuration of a single ClaimCenter node and a single reporting node, as with any setup that uses a larger number of nodes.

Using a load-balancer in conjunction with proper configuration ensures that a web browser sees Guidewire Standard Reporting and ClaimCenter as running from the same domain. If not, the web browser likely identifies the switch from one domain to another as a cross-scripting security risk and therefore prevents the access to Guidewire Standard Reporting.

By default, as a user attempts to access Standard Reporting from within ClaimCenter, the ClaimCenter application server tests (pings) the report server using the load-balanced URL. A successful test verifies that the report server is running and accessible. If successful, ClaimCenter displays a reporting frame within the ClaimCenter **Report** tab.

IMPORTANT It is possible that the network security in place at an installation site can prevent one application from contacting (pinging) another in this fashion. If so, you need to resolve this issue. Guidewire requires that the ClaimCenter application server ping the report server before loading report content into the ClaimCenter **Report** tab.

Configuring Report Clustering

The following example cluster configuration uses four machines (hosts).

- Host A (`hostA_lb`) is the load balancer. In this example, Guidewire uses the Apache HTTP Server 2.2 as the load balancer.
- Host B (`hostB_app1`) is a Guidewire ClaimCenter application server, using Tomcat as the server.
- Host C (`hostC_app2`) is a Guidewire ClaimCenter application server, using Tomcat as the server.
- Host D (`hostD_rpt`) is the InetSoft report server, using Tomcat as the server.

The following table lists the host information and the software loaded on each.

Host name	Software	Purpose	Server designation	Proxy port	Connector
hostA_lb	Apache HTTP Server 2.2	Load balancer	Not applicable	80	—
hostB_app1	Guidewire ClaimCenter	Application server	`tomcatApp1`	80	13510
hostC_app2	Guidewire ClaimCenter	Application server	`tomcatApp2`	80	13510
hostD_rpt	InetSoft StyleReport	Report server	`tomcatRptSvr`	80	7270

To configure a Guidewire ClaimCenter cluster with Standard reporting, perform the following steps:

- Step 1: Configure the Load Balancer
- Step 2: Configure Application Server One
- Step 3: Configure Application Server Two
- Step 4: Configure the Report Server
- Step 5: Start the Cluster Nodes
- Step 6: Test Your Clustering Configuration

Step 1: Configure the Load Balancer

Do the following on the load balancer host (`hostA_lb`):

1. Download and install Apache HTTP server 2.2 to use as the load balancer for the cluster.
2. Modify `httpd.conf` on `hostA_lb` (the load balancer) to configure the HTTP server.

```
#
# This is the main Apache HTTP server configuration file.  It contains the
# configuration directives that give the server its instructions.
# ...
...
LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
LoadModule proxy_connect_module modules/mod_proxy_connect.so
LoadModule proxy_http_module modules/mod_proxy_http.so
...
<VirtualHost _default_:80>
```

```
# Allow access to all
<Proxy *>
  Order Deny,Allow
  Allow from all
</Proxy>

# Do not allow access to the root directory of the application server
<Directory />
  Order Deny,Allow
  Deny from all
</Directory>

# Allow access to the ClaimCenter application directory and its subdirectories to all
<Directory /cc>
  Order Deny,Allow
  Allow from all
</Directory>

# Allow access to sree (the reporting application directory) and its subdirectories to all
<Directory /sree >
  Order Deny,Allow
  Allow from all
</Directory>

#The Virtual Host load-balances requests to ClaimCenter to hostB_app1 & hostC_app2 on port 13510
<Proxy balancer://ClaimCenter1>
  BalancerMember http://hostB_app1:13510 route=tomcatApp1
  BalancerMember http://hostC_app2:13510 route=tomcatApp2
</Proxy>
ProxyPass        /cc  balancer://ClaimCenter1/cc stickysession=JSESSIONID nofailover=Off
ProxyPassReverse /cc      http://hostB_app1:13510/cc
ProxyPassReverse /cc      http://hostC_app2:13510/cc

#The Virtual Host load-balances requests to Reporting to hostD_rpt on port 7270
<Proxy balancer://sree1>
  BalancerMember http://hostD_rpt:7270 route=tomcatRptSvr
</Proxy>
ProxyPass         /sree  balancer://sree1/sree stickysession=JSESSIONID nofailover=Off
ProxyPassReverse  /sree      http://hostD_rpt:7270/sree

#Logs redirected to appropriate location
LogLevel debug
ErrorLog logs/ReportingInvestigation.log

</VirtualHost>
```

Step 2: Configure Application Server One

Do the following to configure Guidewire ClaimCenter for application server one (`hostB_app1`).

1. Install Guidewire ClaimCenter in a development environment.

IMPORTANT In general, Guidewire does **not** recommend that you install ClaimCenter application files on a production server. Instead, install the application on a development server, build the WAR file there, and then deploy that WAR file to the production application server.

---

2. Open the `ClaimCenter config.xml` file.

   a. Add the following to indicate that application server is a part of a cluster. Enter the correct IP value for the multicast address.

   ```
   <param name="ClusteringEnabled" value="true"/>
   <param name="ClusterMulticastAddress" value="228.9.9.9"/>
   <registry>
     <server isbatchserver="true" serverid="hostB_app1"/>
     <server serverid="hostB_app1"/>
   </registry>
   <param name="ConfigVerificationEnabled" value="false"/>
   ```

   Multicast address. The valid range for multicast addresses is 224.0.0.0 to 239.255.255.255. See the following for more information:

   ```
   http://en.wikipedia.org/wiki/Multicast_address
   ```

   You must set the multicast address to the same value for all ClaimCenter servers.

Registry tag. For information on the use of the `<registry>` tag, see "Configuring the Registry Element for Clustering" on page 84 in the *System Administration Guide*.

b. Add the following to activate ClaimCenter Standard Reporting. Notice that the `StyleReportURL` value points to the load balancer machine name.

```
<param name="StyleReportURL" value="http://hostA_lb/sree"/>
```

3. Open the Studio Web Services editor for the web service `sree` and select the `Enable` checkbox next to `Override URL` for the `SoapRepository` web service. (Click **Lookup Services** if you do not see the checkbox.) Ensure that the URL points to the load balancer:

```
http://hostA_lb/sree/services/SoapRepository.SoapRepositoryHttpSoap11Endpoint
```

This ensures that all web service calls outbound from a ClaimCenter application server to the InetSoft report server pass through the load balancer (which clustering requires).

4. Build a `cc.war` file for the ClaimCenter application.

5. Deploy the `war` file to the `webapps` directory on the `hostB_app1` Tomcat server. For information on building application war files, consult *ClaimCenter Installation Guide.*

6. Open the Tomcat `server.xml` file and add the following:

a. Add the following to add a connector to this application server:

```
<Connector conectionTimeout="20000"
           maxThreads="150"
           port="13510"
           protocol="HTTP/1.1" scheme="http"
           proxyName="hostA_lb"
           proxyPort="80" />
```

Set the `port` value to the value that you set for the load balancer connector in `server.xml`.

b. Add the following to configure the `jvmRoute` value:

```
<Engine name="Catalina" defaultHost="localhost" jvmRoute="tomcatApp1">
```

Step 3: Configure Application Server Two

To configure a second application server for the cluster, you need merely repeat the procedure outlined in Step 2: Configure Application Server One, making the following changes:

File	tomcatApp1	tomcatApp2
`config.xml`	`<server isbatchserver="true" serverid="hostB_app1"/>`	`<server isbatchserver="false" serverid="hostC_app2"/>`
`server.xml`	`<Engine ... jvmRoute="tomcatApp1">`	`<Engine ... jvmRoute="tomcatApp2">`

To add additional application servers to the cluster, repeat these steps and modify the listed files accordingly.

Step 4: Configure the Report Server

Do the following to configure the report server.

1. Open `build.properties` (in the InetSoft application) for editing and add the following to it:

```
build.wsdlurl=http://hostA_lb/cc/soap/ISREEAuthenticationAPI?wsdl
build.sree_home=<Tomcat_home>/webapps/sree
```

2. Open `sree.properties` (in the InetSoft application) and set property `gw.url` to the following:

```
gw.url=http://hostA_lb/cc/soap/ISREEAuthenticationAPI
```

This ensures that the InetSoft report server connects back to the Guidewire ClaimCenter server through the load balancer.

3. Build a report `sree.war` file and deploy it to the reporting server (`hostD_rpt`). For example, for Tomcat, run the following command:

   ```
   ant Tomcat
   ```

4. On the report server (`hostD_rpt`), open `server.xml`:

   a. Add the following to add a connector to the report server (`hostD_rpt`).

   ```
   <Connector conectionTimeout="20000"
              maxThreads="150"
              port="7270"
              protocol="HTTP/1.1" scheme="http"
              proxyName="hostA_lb"
              proxyPort="80" />
   ```

   Set the `port` value to the value that you set for the *load balancer connector* in `server.xml`.

   b. Add the following to configure the `jvmRoute` value:

   ```
   <Engine name="Catalina" defaultHost="localhost" jvmRoute="tomcatRptSvr">
   ```

Step 5: Start the Cluster Nodes

After completing the previous steps, start the cluster nodes in the following order:

1. Start the HTTP load balancer (`hostA_lb`).
2. Start the ClaimCenter application server on `hostB_app1`.
3. Start the ClaimCenter application server on `hostC_app2`.
4. Start the report server (`hostD_rpt`).

If you start the report server before the ClaimCenter servers, then the report console log displays an `AxisFault (HTTP error 503)` error message. This does not affect the reporting functionality. In most cases, it is likely that the ClaimCenter servers are already running as you must start these servers as a prerequisite for building the report server `.war` file (`sree.war`).

Step 6: Test Your Clustering Configuration

After you have completed the previous steps, verify your configuration by performing the following tests.

To test the configuration

1. Log into ClaimCenter as a report administrator. Specify the load balancer server with **no** port number in the login URL. For example:

Correct	`http://hostA_lb/cc`
Incorrect	`http://hostB_app1:8080/cc`
Incorrect	`http://hostB_app1:13510/cc`

2. Synchronize the ClaimCenter report tree against the InetSoft report server (**Administration** → **Report Admin**).
3. Verify that a user with the report administration permission can create a new report permission.
4. Verify that a user with the new report permission can run a report.
5. Verify that a user with the report administration permission can log into the InetSoft Enterprise Manager portal from `http://hostA_lb/sree`.

Example Load Balancer httpd.conf File

The following is an example of a load balancer `httpd.conf` file.

```
ThreadsPerChild 250
MaxRequestsPerChild  0
ServerRoot "C:/ApacheServer2.2"
Listen 80
LoadModule actions_module modules/mod_actions.so
LoadModule alias_module modules/mod_alias.so
LoadModule asis_module modules/mod_asis.so
LoadModule auth_basic_module modules/mod_auth_basic.so
LoadModule authn_default_module modules/mod_authn_default.so
LoadModule authn_file_module modules/mod_authn_file.so
LoadModule authz_default_module modules/mod_authz_default.so
LoadModule authz_groupfile_module modules/mod_authz_groupfile.so
LoadModule authz_host_module modules/mod_authz_host.so
LoadModule authz_user_module modules/mod_authz_user.so
LoadModule autoindex_module modules/mod_autoindex.so
LoadModule cgi_module modules/mod_cgi.so
LoadModule dir_module modules/mod_dir.so
LoadModule env_module modules/mod_env.so
LoadModule imagemap_module modules/mod_imagemap.so
LoadModule include_module modules/mod_include.so
LoadModule isapi_module modules/mod_isapi.so
LoadModule log_config_module modules/mod_log_config.so
LoadModule mime_module modules/mod_mime.so
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
LoadModule proxy_connect_module modules/mod_proxy_connect.so
LoadModule proxy_http_module modules/mod_proxy_http.so
LoadModule negotiation_module modules/mod_negotiation.so
LoadModule setenvif_module modules/mod_setenvif.so
LoadModule userdir_module modules/mod_userdir.so
ServerAdmin admin@MyCompany.com
ServerName ThisServer.MyCompany.com:80
DocumentRoot "C:/ApacheServer2.2/htdocs"
<Directory />
  Options FollowSymLinks
  AllowOverride None
  Order deny,allow
  Deny from all
  Satisfy all
</Directory>
<Directory "C:/ApacheServer2.2/htdocs">
  Options Indexes FollowSymLinks
  AllowOverride None
  Order allow,deny
  Allow from all
</Directory>
<IfModule dir_module>
  DirectoryIndex index.html
</IfModule>
<FilesMatch "^\.ht">
  Order allow,deny
  Deny from all
</FilesMatch>
ErrorLog logs/error.log
LogLevel warn
<IfModule log_config_module>
  LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
  LogFormat "%h %l %u %t \"%r\" %>s %b" common
  <IfModule logio_module>
    LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" %I %O" combinedio
  </IfModule>
  CustomLog logs/access.log common
</IfModule>
<IfModule alias_module>
  ScriptAlias /cgi-bin/ "C:/ApacheServer2.2/cgi-bin/"
</IfModule>
<Directory "C:/ApacheServer2.2/cgi-bin">
  AllowOverride None
  Options None
  Order allow,deny
  Allow from all
</Directory>
DefaultType text/plain
<IfModule mime_module>
  TypesConfig conf/mime.types
```

```
  AddType application/x-compress .Z
  AddType application/x-gzip .gz .tgz
</IfModule>
<IfModule ssl_module>
  SSLRandomSeed startup builtin
  SSLRandomSeed connect builtin
</IfModule>
<VirtualHost _default_:80>

  #Allow access to all
  <Proxy *>
    Order Deny,Allow
    Allow from all
  </Proxy>

  # Do not allow access to the root directory of the application server
  <Directory />
    Order Deny,Allow
    Deny from all
  </Directory>

  #Allow access to cc and its subdirectories to all
  <Directory /cc >
    Order Deny,Allow
    Allow from all
  </Directory>

  #Allow access to sree and its subdirectories to all
  <Directory /sree >
    Order Deny,Allow
    Allow from all
  </Directory>

  #The Virtual Host load-balances requests to ClaimCenter to hostB_app1 & hostC_app2 on port 13510
  <Proxy balancer://ClaimCenter1>BalancerMember http://host:13510 route=tomcatApp1</Proxy>
  ProxyPass         /cc  balancer://ClaimCenter1/cc stickysession=JSESSIONID nofailover=Off
  ProxyPassReverse  /cc  http://hostB_app1:13510/cc

  #The Virtual Host load-balances requests to Reporting to struma on port 7170
  <Proxy balancer://sree1>BalancerMember http://hostD_rpt:7170 route=tomcatRptSvr</Proxy>
  ProxyPass         /sree  balancer://sree1/sree stickysession=JSESSIONID nofailover=Off
  ProxyPassReverse  /sree  http://hostD_rpt:7170/sree

  #Logs redirected to appropriate location
  LogLevel debug
  ErrorLog logs/ReportingInvestigation.log

</VirtualHost>
```

chapter 7

Using ClaimCenter Reports

Guidewire provides a number of predefined reports in the base configuration of ClaimCenter Standard Reporting. A user with sufficient permission can view these reports from within the ClaimCenter **Report** tab. In addition, a user with InetSoft administrative permission can view the ClaimCenter reports from within the InetSoft Enterprise Manager.

This topic includes:

Running ClaimCenter Reports

After completing the Style Report installation and configuration and its integration with Guidewire ClaimCenter, you can run ClaimCenter reports.

> **Note:** If the report server is unavailable, you see an error message to that effect. If the report database is unavailable, you see either xxx in report data fields or nothing. If either of the conditions happens, contact your system administrator for resolution.

To run reports from ClaimCenter

1. Log into ClaimCenter as a user with report viewing permission.
2. Select the **Report** tab.
3. Select a report folder, then a report. Almost all of the ClaimCenter reports have an initial parameter page. To select all the parameters, simply click **OK**. Otherwise, select the desired parameters, then click **OK**.

Scheduling ClaimCenter Reports

It is possible to use InetSoft Scheduler to schedule the generation of a specific report or reports. Using InetSoft Scheduler, you can specify the time and date to generate the report. For example, you can schedule InetSoft Style Report to generate a specific report at 3:00 a.m. on a daily basis. Refer to the InetSoft Style *Report Administration Reference Guide* for details, especially the section on the Scheduler.

Configuring Drill-Down Reports

Many ClaimCenter reports provide hyperlinks on the claim number that you can click to access claim information directly within ClaimCenter itself. If this feature is active, clicking the link takes you directly to the relevant claim summary, financial or activities page. This capability requires additional configuration. For details, see "Working with the Report Designer" on page 127.

Be aware that passing a typelist as a parameter to a drill-down report can cause issues if one of its typecode names contains a comma. This is a limitation of the InetSoft reporting software. If a typecode name does contain a comma, InetSoft thinks it is an array and sends multiple values.

For example, the sample data for the **Loss Cause** typelist contains the following typecode:

Typecode	Name
`fall`	Fall, slip, or trip injury

The problem arises if you try to pass the **Loss Cause** as a parameter to a drill-down report. If the **Name** value for the **Loss Cause** typelist does not contain a comma, there is no issue. However, if the **Name** value does contain a comma, Inetsoft thinks it is an array and sends three values:

- fall
- slip
- or trip injury

For the specific case of the **Loss Cause** typelist, Guidewire includes a column in table `cctl_LossCause` that contains the **Loss Cause** name values without commas. ClaimCenter passes these values as the typecode parameter to the drill-down report. It is these values that the drill-down report uses for filtering. If you want to use a typecode that contains a comma as a parameter to a drill-down report, then you need to implement a similar solution for that typelist.

IMPORTANT Be aware that passing a typecode that contains one or more commas as a parameter to a drill-down report can cause problems.

Viewing the Guidewire Reports

In the default configuration, Guidewire ClaimCenter provides a number of sample reports, organized into the following main groups:

- Claim Reports
- Claim Health Metrics Reports
- Dashboard Reports
- Financial Reports
- Special Investigation Unit (SIU) Reports

The Dashboard folder. A user who logs into ClaimCenter with the `View Dashboard` permission and no reporting permission sees the ClaimCenter **Dashboard** tab with its standard reports. (ClaimCenter enforces the permission to view the Dashboard at the Organization level of the user.) However, a user who logs into ClaimCenter with the `Show Reports and Dashboard` permission (and reporting enabled) sees a **Reports** tab with a **Dashboard** folder. Within the **Dashboard** folder, a user with the required permission can see the same set of ClaimCenter reports recreated through the InetSoft Report Designer.

Claim Reports

Guidewire provides the following Claim reports in the base configuration:

- Activity List
- Claim Catastrophe Detail
- Claim Detail
- Claim Detail by Underwriter
- Claim Injury Detail
- Claims Aging
- Claims by Catastrophe
- Claims by Catastrophe Overview
- Claims by Injury Type
- Claims Processing Effectiveness
- Claims without Updates
- Current Overdue Activities
- First Payment Productivity
- Litigation Claim Detail
- Litigation Rate Trend
- Litigation Summary
- Loss Run
- Open Claims by LOB
- Overdue Activities Rate
- Overdue Activities
- Past 30 Day Claim Summary
- Reopened Claims List
- Subrogated Claims

Activity List

The *Activity List* report provides activity detail information. This report provides information on the following:

Name	Description
Actual Days	Actual number of days between assigned date and completion date
Late Days	Number of days that the activity was completed late
Scheduled to Complete Days	Number of days between assigned date and target completion date

Claim Catastrophe Detail

The *Claim Catastrophe Detail* report provides detailed information on each claim that involves a catastrophe. This report provides information on the following:

Name	Description
Days from Loss to Reported	Number of days between Loss Date and Reported Date
Days from Reported to Closed	Number of days between the claim Reported Date and the date on which the claim was closed
Days from Closed to Reopened	Number of days between the closed date of the claim and the reopened date of the claim (for those claims that have been reopened)
Days Open	Number of days that the claim is open
Days Reopen	Number of days that the claim was in reopened state
Days from Litigation	Number of days between the time the report was run to the date on which the matter was filed. (If there are multiple matters on the claim, the report uses the earliest file date.)

Name	Description
Claim Cost Paid	Amount paid on the claim due to cost of indemnifying the claim
Expenses Paid	Amount paid due to the claim processing expenses
Total Paid	Amount of claim cost and expenses paid
Remaining Claim Cost Reserves	Amount of claim cost reserves
Remaining Expense Reserves	Amount of expense reserves
Recovery	Amount recovered
Gross Total Incurred	Amount of payments, plus remaining reserves
Net Total Incurred	Amount of payments plus remaining reserves minus recoveries

Claim Detail

The *Claim Detail* report provides detailed information on each claim. This report provides information on the following:

Name	Description
Claim Cost Paid	Amount paid on the claim due to cost of indemnifying the claim
Days from Closed to Reopened	Number of days between the closed date of the claim and the reopened date of the claim (for those claims that have been reopened)
Days from Litigation	Number of days between the time the report was run to the date on which the matter was filed. (If there are multiple matters on the claim, the report uses the earliest file date.)
Days from Loss to Reported	Number of days between Loss Date and Reported Date
Days from Reported to Closed	Number of days between the claim Reported Date and the date on which the claim was closed
Days Open	Number of days that the claim is open
Days Reopen	Number of days that the claim was in reopened state
Expenses Paid	Amount paid due to the claim processing expenses
Gross Total Incurred	Amount of payments, plus remaining reserves
Net Total Incurred	Amount of payments plus remaining reserves minus recoveries
Open Claim Cost Reserves	Amount of claim cost reserves
Open Expense Reserves	Amount of expense reserves
Recovery	Amount recovered
Total Paid	Amount of claim cost and expenses paid

Claim Detail by Underwriter

The *Claim Detail by Underwriter* report displays claim detail information by underwriter company and underwriter group. This report provides information on the following:

Name	Description
Claim Cost Paid	Amount paid on the claim due to cost of indemnifying the claim
Days from Closed to Reopened	Number of days between the closed date of the claim and the reopened date of the claim (for those claims that have been reopened)
Days from Litigation	Number of days between the time the report was run to the date on which the matter was filed. (If there are multiple matters on the claim, the report uses the earliest file date.)
Days from Loss to Reported	Number of days between Loss Date and Reported Date
Days from Reported to Closed	Number of days between the claim Reported Date and the date on which the claim was closed

Name	Description
Days Open	Number of days that the claim is open
Days Reopen	Number of days that the claim was in reopened state
Expenses Paid	Amount paid due to the claim processing expenses
Gross Total Incurred	Amount of payments, plus remaining reserves
Net Total Incurred	Amount of payments plus remaining reserves minus recoveries
Open Claim Cost Reserves	Amount of claim cost reserves
Open Expense Reserves	Amount of expense reserves
Recovery	Amount recovered
Total Paid	Amount of claim cost and expenses paid

Claim Injury Detail

The *Claim Injury Detail* report provides detailed information on each claim, including the injury type. This report provides information on the following:

Name	Description
Days from Loss to Reported	Number of days between Loss Date and Reported Date
Days from Reported to Closed	Number of days between the claim Reported Date and the date on which the claim was closed
Days from Closed to Reopened	Number of days between the closed date of the claim and the reopened date of the claim (for those claims that have been reopened)
Days Open	Number of days that the claim is open
Days Reopen	Number of days that the claim was in reopened state
Days from Litigation	Number of days between the time the report was run to the date on which the matter was filed. (If there are multiple matters on the claim, the report uses the earliest file date.)
Claim Cost Paid	Amount paid on the claim due to cost of indemnifying the claim
Expenses Paid	Amount paid due to the claim processing expenses
Total Paid	Amount of claim cost and expenses paid
Open Claim Cost Reserves	Amount of claim cost reserves
Open Expense Reserves	Amount of expense reserves
Recovery	Amount recovered
Gross Total Incurred	Amount of payments, plus remaining reserves
Net Total Incurred	Amount of payments plus remaining reserves minus recoveries

Claims Aging

The *Claims Aging* report provides a count of claims (by Group, Adjuster, LOB, Loss Type, and Loss Cause) that are open. The claim must have been reported within the specified time period, defined as:

- last 1-90 days
- last 91-180 days
- last 181-270 days
- last 271-365 days

This report displays information on the number of claims that are open relative to their “reported on” date periods. From this report, the user can drill down (through Group) to the *Claim Detail* report. The *Claims Aging* report provides information on the following:

Name	Description
Open Claim Count	Number of open claims for each time period

Claims by Catastrophe

The *Claims by Catastrophe* report provides a count of claims (by Group, LOB, and Loss Cause) of the number of claims related to a catastrophe. (You associate a claim with a catastrophe through the ClaimCenter interface.)

Claims by Catastrophe Overview

The *Claims by Catastrophe Overview* report provides information in both tabular and chart form on each claim that involves a catastrophe. This report categorize Catastrophe by Loss State, Loss Cause, and Loss Type. From the report, you can drill down to more data through any of the following:

- Loss State
- Loss Cause
- Loss Type
- The charts

Claims by Injury Type

The *Claims by Injury Type* report provides a count of open claims (by Group, Adjuster, LOB, and Loss Cause) for the reported type of injury involved. From this report, the user can drill down (through the claim number) to the *Claim Injury Detail* report.

Claims Processing Effectiveness

The *Claim Processing Effectiveness* report provides a trend of the average days to close a claim (by Group, Adjuster, LOB, Loss Type, and Loss Cause) for claims. The claim must have been closed within the specified time period, defined as:

- last 1-90 days
- last 91-180 days
- last 181-270 days
- last 271-365 days

One use for this report is to analyze the average days to close a claim across the different time periods. (This allows you to assess whether or not the organization is improving the ability to process claims.) From this report, the user can drill down (through Group) to the *Claim Detail* report. The *Claim Processing Effectiveness* report provides information on the following:

Name	Description
Average Days to Close	Calculated from number of days between the claim reported date and the date the claims were closed, divided by the number of claims that were closed for each time period

Claims without Updates

The *Claims without Updates* report provides the list of all claims that have not yet been updated, based on a user-entered date. This report provides information on the following:

Name	Description
Reported Date	Date claim was reported
Update Date	Date claim was last updated

Current Overdue Activities

The *Current Overdue Activities* report provides a list of activities on open claims (by Group, Adjuster, LOB, and Loss Cause) that are overdue. This report provides information on the following:

Name	Description
Days Overdue	Number of Days than an activity is overdue

First Payment Productivity

The *First Payment Productivity* report provides a trend of average duration to first payment for claims that were opened within the specified time period, defined as:

- last 1-90 days
- last 91-180 days
- last 81-270 days
- last 271-365 days

One use for this report information is to analyze the average days to first payment across time periods (by Group, Adjuster, LOB, Loss Type, and Loss Cause). This helps to assess whether or not the organization is improving the average duration to first payment. From this report, the user can drill down (through Group) to the *Claim Detail* report. The *First Payment Productivity* report provides information on the following:

Name	Description
Average Days to First Payment	Number of days between the claim reported date and the date of the first payment transaction, divided by the number of claims that were reported during that time period.

Litigation Claim Detail

The *Litigation Claim Detail* report provides detailed information on the litigation status of an open claim. The *Litigation Claim Detail* report provides information on the following:

Name	Description
Days from Loss to Reported	Number of days between Loss Date and Reported Date
Days from Reported to Closed	Number of days between the claim Reported Date and the date on which the claim was closed
Days from Closed to Reopened	Number of days between the closed date of the claim and the reopened date of the claim (for those claims that have been reopened)
Days Open	Number of days that the claim is open
Days Reopen	Number of days that the claim was in reopened state

Name	Description
Days from Litigation	Number of days between the time the report was run to the date on which the matter was filed. (If there are multiple matters on the claim, the report uses the earliest file date.)
Claim Cost Paid	Amount paid on the claim due to cost of indemnifying the claim
Expenses Paid	Amount paid due to the claim processing expenses
Total Paid	Amount of claim cost and expenses paid
Open Claim Cost Reserves	Amount of claim cost reserves
Open Expense Reserves	Amount of expense reserves
Recovery	Amount recovered

Litigation Rate Trend

The *Litigation Rate Trend* report provides analysis of claims that are litigated and opened within the specified time period, defined as:

- last 1-90 days
- last 91-180 days
- last 181-270 days
- last 271-365 days

One use for this report is to asses how litigation trends over different time periods. From this report, the user can drill down (through Group) to the *Litigation Summary* report. The *Litigation Rate Trend* report provides information on the following:

Name	Description
Litigation Rate	Number of claims that are litigated for claims opened within each time period, divided by number of claims that were opened within each time period

Litigation Summary

The *Litigation Summary* report provides a summary of Litigated Claims by loss year. From this report, the user can drill down (through Group) to the *Litigation Claim Detail* report. This report provides information on the following:

Name	Description
Open Litigated Claims	Number of litigated claims that remain open
Closed Litigated Claims	Number of litigated claims that are closed

Loss Run

The *Loss Run* report provides loss information for each claim associated with a given policy.

Open Claims by LOB

The *Open Claims by LOB* report provides a pie chart of the number of open claims by Line of Business (LOB). From this report, the user can drill down to the *Claim Detail* report. This report provides information on the following:

Name	Description
Claims	Number of claims that are open, by Line of Business

Overdue Activities

The *Overdue Activities* report provides a historical list of activities that have become overdue. This report provides information on the following:

Name	Description
Days Overdue	Number of Days that an activity is overdue

Overdue Activities Rate

The *Overdue Activities Rate* report provides the rate at which activities became overdue (by Group, Adjuster, LOB, and Loss Cause) within the specified time period, defined as:

- last 1-90 days
- last 91-180 days
- last 181-270 days
- last 271-365 days

This report provides information on the following:

Name	Description
Overdue Activity Rate	Rate (in percent) at which activities became overdue in the specified time period.

Past 30 Day Claim Summary

The *Past 30 Day Claim Summary* report provides a summary of claim and activity metrics (by Group, Adjuster, LOB, Loss Type, and Loss Cause) for the last 30 days. From this report, the user can drill down (through Group) to the *Claim Detail* report. This report provides information on the following:

Name	Description
Newly Opened Claims	Number of claims reported within the last 30 days
Newly Closed Claims	Number of claims closed within the last 30 days
Average Days to Close	Number of days to close divided by number of closed claims
Newly Reopened Claims	Number of claims reopened within the last 30 days
Newly Open Activities	Number of activities opened within the last 30 days
Newly Closed Activities	Number of activities closed within the last 30 days
New Exposures	Number of exposures created within the last 30 days
New Notice Only Claims	Number of notice-only claims within the last 30 days
New Litigation	Number of claims that have new matters within the last 30 days

Reopened Claims List

The *Reopened Claims* report provides claim detail information for those claims that have been reopened. This report provides information on the following:

Name	Description
Claim Cost Paid	Amount paid on the claim due to cost of indemnifying the claim
Days from Closed to Reopened	Number of days between the closed date of the claim and the reopened date of the claim for those claims that have been reopened

Name	Description
Days from Litigation	Number of days between the time the report was run to the date on which the matter was filed. (If there are multiple matters on the claim, the report uses the earliest file date.)
Days from Loss to Reported	Number of days between Loss Date and Reported Date
Days from Reported to Closed	Number of days between the claim Reported Date and the date on which the claim was closed
Days Open	Number of days that the claim is open
Days Reopen	Number of days that the claim was in reopened state
Expenses Paid	Amount paid due to the claim processing expenses
Gross Total Incurred	Amount of payments, plus remaining reserves
Net Total Incurred	Amount of payments plus remaining reserves minus recoveries
Open Claim Cost Reserves	Amount of claim cost reserves
Open Expense Reserves	Amount of expense reserves
Recovery	Amount recovered
Total Paid	Amount of claim cost and expenses paid

Subrogated Claims

The *Subrogated Claims* report provides a list (by Group, Adjuster, LOB, Loss Cause) of subrogated claims. From this report, the user can drill down (through the claim number) to the *Claim Detail* report.

Claim Health Metrics Reports

Claim Health Metrics allow you to see how your team, claims, or exposures are performing based on metrics defined by your business requirements. These reports provide the following benefits:

- **Managers and supervisors can immediately see the status of closed claims and exposures and determine which ones were problematic.** This allows them to analyze the data and possibly adjust their business practices or redistribute their subordinates work.
- **Reports are organized by overall average, group, group and tier, and exposure by tier.** Setting tiers is optional and allows you to have different target values for a particular metric within a specific policy type. Tiers are a way to have further granularity within the policy type.
- **Managers and supervisors can also see the status of open claims and exposures.** This means that they have the opportunity to be proactive in working with the claim or exposure.
- **Adjusters can also see their own open claims and exposures.** Viewing their reports (My Claim and Exposure Alerts) allows them to focus their efforts on claims that might be problematic and be proactive in working on them.
- **You can drill down on a report to see additional information.** This additional granularity allows you to quickly see specific details quickly. This includes drilling down to see the specific claim or exposure in ClaimCenter. Managers and supervisors drill down to see how specific teams or adjusters are handling claims and which claims or exposures were or still are problematic. Adjusters drill down to quickly see the areas of the claim that they can be proactive in working on.
- **You can use filters to selectively see report data.** Reports are filtered by the groups that the user manages or owns. This means that when you drill down to see additional information, you see only the groups or claims that you or your team are responsible for. You can also filter by date. For example, you can filter by month to date, or by the previous month (for managers or supervisors). You can also filter by group (such as your top three or five or by a specific group). Adjusters can filter their reports by the past day or week.

Guidewire ships ClaimCenter Standard Reporting with the following Claim Health Metrics reports. Select a link to go to the individual report and the metrics that further define it.

- Claim Overall Avg

- Claim by Group
- Claim by Group and Tier
- Exposure by Tier
- Open Alert Supervisor Red
- Open Alert Supervisor Yellow
- My Claim and Exposure Alerts Red
- My Claim and Exposure Alerts Yellow

Reports Organization

The reports are organized as follows:

Report Name	Users	Status of Claim or Exposure
Claim Overall Average	Supervisors or managers	closed claims
Claim by Group	Supervisors or managers	closed claims
Claim by Group and Tier	Supervisors or managers	closed claims
Exposure by Tier	Supervisors or managers	closed exposures
Open Alert Supervisor Red	Supervisors or managers	open claims or exposures
Open Alert Supervisor Yellow	Supervisors or managers	open claims or exposures
My Claim and Exposure Alerts Red	Adjusters	open claims or exposures
My Claim and Exposure Alerts Yellow	Adjusters	open claims or exposures

How the Metrics are Calculated

These metrics are average metrics calculated using the filters of groups and time (such as month to date or past seven days). As previously mentioned, managers see only groups that they are responsible for. Adjusters see only the claims or exposures they are responsible for.

The reports contains pie or bar charts which can be divided by color. These charts represent the different metrics that the report is based on. The pie or bar chart counts the number of claims that fall into the Red, Yellow, and Green categories for each metric. There are two fields from the metrics table that are used to calculate the number of claims that fall into the Red, Yellow, and Green categories. There are `ReachYellowDate` and `ReachRedDate`. If you have administration permissions, you can see the metrics table in ClaimCenter. Navigate to the **Administration** tab → **Metrics and Thresholds** → **Claim Metric Limits** (or **Exposure Metric Limits**).

- **Time Dependent** For time dependent metrics, the dates for when the metric is expected to turn yellow or red are populated when the metric is still open. When the metric is completed, ClaimCenter populates the dates that correspond with those thresholds that have been met.
- **Time Independent** For time independent metrics, these dates are populated once the thresholds are met.

The following logic governs how the claims are counted for each metric.

For *closed* claim aggregate reporting:

- If both dates are `null`, then count the metric as green.
- If the `Problemdate` is populated, then count the metric as red.
- If the `Problemdate` is `null`, and the `WarningDate` is populated, then count the metric as yellow.

For *open* claim aggregate reporting:

- A metric is yellow if:
  - The date evaluated (yesterday or last seven days) is greater than or equal to the `ReachYellowTime`, and
  - The `ReachRedTime` is null or less than the `ReachRedTime`

- A metric is red if the date evaluated (yesterday or last seven days) is equal to or greater than the `ReachRedTime`.

Claim Overall Avg

The *Claim Overall Avg* report provides the averages for the metrics across all the different groups of closed claims. It is filtered by the groups that the user managers or owns. Drilling down on the pie charts displays a table organized by:

- group
- team member
- claim number, and
- the specifics of that pie chart (such as days open or paid loss costs as a percentage of total paid).

Further clicking on a claim number opens the claim so you can view specific details of that claim. The date range is either the current month to date or the previous month.

The Claim Overall Avg report contains the following metrics which you can click to get additional details. Clicking on a specific color in the pie chart displays the claims that fall under that indicator.

Name	Description
Overall Claim Metrics	
Days Open	Average days open.
Initial Contact with Insured (Days)	Average time to initial contact.
Claim Activity	
Percentage of Escalated Activities	Average days open.
Claim Financials	
Time to First Loss Payment (Days)	Average time to first loss payment.
Number of Reserve Changes	Average number of reserve changes.
Net Total Incurred	Average net total incurred.
Paid Loss Costs as Percentage of Total Paid	Average paid loss costs as a percentage of the total paid.

Claim by Group

The *Claim by Group* report provides detail information about closed claims by group. This report is filtered to show only the groups managed or supervised by the user. You can filter based on date, and by group. Choices for group include your top three or five groups or you can select a specific group. You also have the option in each metric to view the full report. Doing so displays all your groups. You can further drill down by clicking a number in the **Number of Claims** column. From this view, you can access the claim in question.

This report provides information on the following metrics:

Name	Description
Overall Claim Metrics	
Days Open	This report shows the averages, days that the claim was open, and number of claims per group.
Initial Contact with Insured	This report shows the average, days to initial contact with the insured, and number of claims per group.
Claim Activity	
Percentage of Escalated Activities	This report shows the average and number of claims based on percentage of escalated activities per group.
Claim Financials	

Name	Description
Time to First Loss Payment	This report shows the average and number of claims per group related to time to first loss payment.
Number of Reserve Changes	This report shows the average and number of claims relating to number of reserve changes.
Net Total Incurred	This report shows the average and number of claims relating to net total incurred.
Paid Loss Costs as Percentage of Total Paid	This report shows the average and number of claims relating to paid loss costs as a percentage of the total paid.

Claim by Group and Tier

The *Claim by Group and Tier* report provides closed claim detail information based on group and tier. The report provides a comparison of how the group fares against its peers (other groups within the current result set). Groups are filtered by permissions. This means that you only see the groups that you are in charge of. You can filter based on date, and by group. Choices for group include your top three or five groups or you can select a specific group. You also have the option in each metric to view the full report. Doing so displays all your groups. You can further drill down by clicking a number in the **Number of Claims** column. From this view, you can access the claim in question.

This report provides information on the following metrics:

Name	Description
Overall Claim Metrics	
Days Open	This report shows the averages, days that the claim was open, and number of claims per group and tier.
Initial Contact with Insured (Days)	This report shows the average, days to initial contact with the insured, and number of claims per group and tier.
Claim Activity	
Percentage of Escalated Activities	This report shows the average and number of claims based on percentage of escalated activities per group and tier.
Claim Financials	
Time to First Loss Payment	This report shows the average and number of claims per group and tier related to time to first loss payment.
Number of Reserve Changes	This report shows the average, tier, and number of claims relating to number of reserve changes.
Net Total Incurred	This report shows the average, claim tier, and number of claims relating to net total incurred.
Paid Loss Costs as Percentage of Total Paid	This report shows the average, tier, and number of claims relating to paid loss costs as a percentage of the total paid.

Exposure by Tier

The *Exposure by Tier* report provides closed exposure detail information based on exposure tier, average, and number of exposures. It is filtered by date (month to date or previous month) and exposure tier. Use the drop down menu to select the type of exposure tier. You can select your top three, top five, or a specific exposure tier, including exposure group, or policy type.

Clicking on a specific color in the bar chart displays a list of those exposures that fall under that indicator. Select any exposure under the **Exposure Detail** column and ClaimCenter opens the **Exposures** detail screen. You can also select under each metric (such as **Days Open**) a specific exposure (under the **Exposure Tier** column). These drill down reports list any coverages and exposure groups that fall under that indicator. You can optionally choose to view the full report which displays all the exposure tiers.

The *Exposure by Tier* report contains the following metrics (drill down reports).

Name	Description
Overall Exposure Metrics	
Days Open	This report shows the averages, days the exposure was open, and number of closed exposures by exposure tier.
Initial Contact with Claimant	This report shows the average, days to initial contact with claimant, and number of exposures by exposure tier per claimant.
Exposure Financials	
Time to First Loss Payment	This report shows the average and number of exposures by exposure tier related to time to first loss payment.
Net Total Incurred	This report shows the average, claim tier, and number of exposures by exposure tier relating to net total incurred.

Open Alert Supervisor Red

The *Open Alert Supervisor Red* report provides information about which open claims have metric thresholds that have exceeded red. Since the date filter is either the past day or past week, supervisors and managers have the opportunity to immediately respond to a problematic claim or exposure. Since the report lists the claim owner and claim description, supervisors and managers can quickly come up to speed on the type of problematic claim they need to address. Selecting a claim from the **Claim Number** column opens that claim's **Claim Health Metrics** screen.

The *Open Alert Supervisor Red* report provides information on the following:

Name	Description
Open Claim Alerts	
Claims Exceeded Red	This report shows the claims that exceeded red. It also lists the claim owner and a brief description of the claim.
Open Exposure Alerts	
Exposures Exceeded Red	This report shows the open exposures that exceeded red. It also lists the claim owner, exposure owner, and a brief description of the claim.

Open Alert Supervisor Yellow

The *Open Alert Supervisor Yellow* report provides information about which open claims have metric thresholds that have exceeded yellow. Since the date filter is either the past day or past week, supervisors and managers have the opportunity to immediately respond to a claim or exposure before it becomes critical. Selecting a claim from the **Claim Number** column opens that claim's **Claim Health Metrics** screen.

The *Open Alert Supervisor Yellow* report provides information on the following:

Name	Description
Open Claim Alerts	
Claims Exceeded Yellow	This report shows the claims that exceeded yellow. It also lists the claim owner and a brief description of the claim.
Open Exposure Alerts	
Exposures Exceeded Yellow	This report shows the open exposures that exceeded yellow. It also lists the claim owner, exposure owner, and a brief description of the claim.

My Claim and Exposure Alerts Red

The *My Claim and Exposure Alerts Red* report provides open claim and exposure detail information for adjusters. Since the date filter is either the past day or past week, adjusters can immediately respond to a claim or exposure. Selecting a claim from the **Claim Number** column opens that claim's **Claim Health Metrics** screen.

This report provides information on the following:

Name	Description
Open Claim Alerts	
Claims Exceeded Red	This report shows the number of claims that exceeded red.
Open Exposure Alerts	
Exposures Exceeded Red	This report shows the number of exposures that exceeded red.

My Claim and Exposure Alerts Yellow

The *My Claim and Exposure Alerts Yellow* report provides open claim and exposure detail information for adjusters. Since the date filter is either the past day or past week, adjusters can immediately respond to a claim or exposure and thus be proactive in its resolution. Selecting a claim from the **Claim Number** column opens that claim's **Claim Health Metrics** screen.

This report provides information on the following:

Name	Description
Open Claim Alerts	
Claims Exceeded Yellow	This report shows the number of claims that exceeded yellow.
Open Exposure Alerts	
Exposures Exceeded Yellow	This report shows the number of exposures that exceeded yellow.

Dashboard Reports

Guidewire delivers ClaimCenter Standard Reporting with the following Dashboard reports:

- Claim Currency - Open Claim Financials Dashboard
- Open Claim Count
- Open Claim Financials Dashboard
- Past 30 Day Claim Dashboard
- Past 30 Day Financials

Note: A user that accesses these reports (in the **Dashboard** folder) sees only data relating to groups for which that individual is listed as either the manager or supervisor.

Claim Currency - Open Claim Financials Dashboard

The *Claim Currency - Open Claim Financials Dashboard* report provides a financial summary in claim-based currency for all currently open claims, organized by Group, LOB, and Loss Type. You must first select the claim-based currency and the report will be filtered to show only claims with that particular claim-based currency.

Note: You must enable ClaimCenter for multicurrency. See "Enabling Multicurrency and Defining the Default Currency" on page 172 in the *Application Guide* for details.

This report provides information on the following:

Name	Description
Claim Cost Paid on Open Claims	Amount of claim cost paid on open claims
Currency	Claim currency that the user selected.
Expenses Paid on Open Claims	Amount of expenses paid on open claims
Net Total Incurred	Total incurred calculated as Total Paid, plus Open Reserves minus recoveries
Open Claim Count	Number of claims that are open
Open Exposures	Number of open exposures
Open Reserves	Amount of open reserves
Total Paid	Amount of total payments

Open Claim Count

The *Open Claim Count* report provides information on open claims organized by Group, LOB, and Loss Type, using the following:

Name	Description
TI > Limit	Number of Claims that are greater than Total Incurred Limit
Claims	Number of Claims
Exposures	Number of Exposures
Flagged	Number of claims that have been flagged
Litigated	Number of claims that have a litigated status
Notice-Only	Number of claims that are flagged as “notice-only”

Open Claim Financials Dashboard

The *Open Claim Financials Dashboard* report provides a financial summary for all currently open claims, organized by Group, LOB, Loss Cause, and Coverage. From this report, the user can drill down (through each category) to the *Transaction Detail* report. This report provides information on the following:

Name	Description
Average TI/Claim	Gross Total Incurred divided by Open Claim Count
Claim Cost Paid on Open Claims	Amount of claim cost paid on open claims
Expenses Paid on Open Claims	Amount of expenses paid on open Claims
Net Total Incurred	Total incurred calculated as Total Paid, plus Open Reserves minus recoveries
Open Claim Count	Number of Claims that are open
Open Exposures	Number of open exposures
Open Reserves	Amount of open reserves
Total Paid	Amount of total payments

Past 30 Day Claim Dashboard

The *Past 30 Day Claim Dashboard* report provides a summary (by Group, LOB, Loss Cause, and Coverage) of claim and activity metrics for the last 30 days. From this report, the user can drill down (through Group, LOB, Loss Cause, and Coverage) to the *Claim Detail* report. This report provides information on the following:

Name	Description
Average Days to Close	Number of days to close divided by number of closed claims

Name	Description
New Exposures	Number of exposures created within the last 30 days
New Litigation	Number of claims that have new matters within the last 30 days
New Notice Only Claims	Number of notice-only claims within the last 30 days
Newly Closed Activities	Number of activities closed within the last 30 days
Newly Closed Claims	Number of claims closed within the last 30 days
Newly Open Activities	Number of activities opened within the last 30 days
Newly Opened Claims	Number of claims reported within the last 30 days
Newly Reopened Claims	Number of claims reopened within the last 30 days

Past 30 Day Financials

The *Past 30 Day Financials* report provides information (organized by Group, LOB, Loss Cause, and Coverage) on financial information. The report looks at claims over the last 30 days for both the closed and total payments. From this report, the user can drill down (through Group, LOB, and so on) to the *Transaction Detail* report. This report provides information on the following:

Name	Description
Average Days to Close	Number of days to close divided by number of closed claims
New Exposures	Number of exposures created within the last 30 days
New Litigation	Number of claims that have new matters within the last 30 days
New Notice Only Claims	Number of notice-only claims within the last 30 days
Newly Closed Activities	Number of activities closed within the last 30 days
Newly Closed Claims	Number of claims closed within the last 30 days
Newly Open Activities	Number of activities opened within the last 30 days
Newly Opened Claims	Number of claims reported within the last 30 days
Newly Reopened Claims	Number of claims reopened within the last 30 days

Financial Reports

Guidewire ships ClaimCenter Standard Reporting with the following financial reports:

- Average Claim Cost Trend
- Average Expenses Trend
- Catastrophe Financials
- Catastrophe Severity
- Current Catastrophe Financials
- Large Loss Financials
- Monthly Claim Costs Paid
- Monthly Expenses Paid
- Monthly Expenses and Claim Cost
- Monthly Financial Transactions
- Monthly Recoveries
- Open Claim Financials
- Transaction Detail

Average Claim Cost Trend

The *Average Claim Cost Trend* report provides a trend of average claim cost (by Group, Adjuster, LOB, Loss Type, and Loss Cause). The claim must have been closed within the specified time period, defined as:

- last 1-90 days
- last 91-180 days
- last 181-270 days

- last 271-365 days

From this report, the user can drill down (through Group) to the *Transaction Detail* report. The *Average Claim Cost Trend* report provides information on the following:

Name	Description
Average Claim Cost Paid	Claim Cost Paid divided by Claims with Claims Cost Payments

Average Expenses Trend

The *Average Expenses Trend* report provides a trend of average expenses (by Group, Adjuster, LOB, Loss Type, and Loss Cause). The claim must have been closed within the specified time period, defined as:

- last 1-90 days
- last 91-180 days
- last 181-270 days
- last 271-365 days

From this report, the user can drill down (through Group) to the *Transaction Detail* report. The *Average Expenses Trend* report provides information using the following

Name	Description
Average Expenses Paid	Expenses Paid divided by Claims with Expense Payments

Catastrophe Financials

The *Catastrophe Financials* report provides financial metrics for catastrophe claims (both open and closed) by Group, Lines of Business and Loss Cause. From this report, you can drill down to the following more detailed reports:

- Through Group - Catastrophe Financials Detail at Claim Owner Level report
- Through Catastrophe - Catastrophe Financials Detail at Claim Owner Level report

From this report, you can drill down to more detailed catastrophe information by Claim Owner on both the Catastrophe and Claim Group fields.

Catastrophe Severity

The *Catastrophe Severity* report provides a severity rating for closed catastrophe claims by Group, Adjuster, Lines of Business, and Loss Cause. The report defines severity as the ratio of the Paid Amount to the total number of closed claims.

From this report, you can drill down to more detailed severity information by Claim Owner on both the Catastrophe and Claim Group fields.

Current Catastrophe Financials

The *Current Catastrophe Financials* report provides financial metrics (by Group, Adjuster, LOB, and Loss Cause) for catastrophe claims that are currently open. From this report, the user can drill down (through Catastrophe) to the *Claim Catastrophe Detail* report. This report provides information on the following:

Name	Description
Gross Total Incurred	Total incurred calculated as Total Paid, plus Open Reserves for each catastrophe (This report does not account for recoveries.)
Open Claim Count	Number of Claims that are open for each catastrophe
Open Reserves	Amount of open reserves for each catastrophe

Name	Description
Total Paid	Amount of total payments for each catastrophe

Large Loss Financials

The *Large Loss Financials* report provides financial metrics for claims in which the Gross Total Incurred is greater than a user-entered amount. From this report, the user can drill down (through the claim number) to the **Summary** screen. This report provides information on the following:

Name	Description
Gross Total Incurred	Total incurred calculated as Total Paid, plus Open Reserves for each catastrophe (This report does not account for recoveries.)
Open Claim Count	Number of Claims that are open for each catastrophe
Open Reserves	Amount of open reserves for each catastrophe
Total Paid	Amount of total payments for each catastrophe

Monthly Claim Costs Paid

The *Monthly Claim Costs Paid* report provides a monthly breakdown (by Group, Adjuster, LOB, and Loss Cause) of the claim costs paid. From this report, the user can drill down (through Group) to the *Transaction Detail* report. This report provides information on the following:

Name	Description
Average Claim Cost Paid	Claim Cost Paid divided by Claims with Claims Cost Payments
Claim Costs Paid	Amount paid on the claim due to cost of indemnifying the claim
Claims with Claim Cost Payments	Number of Claims that have claim cost payments

Monthly Expenses Paid

The *Monthly Expenses Paid* report provides a monthly breakdown (by Group, Adjuster, LOB, and Loss Cause) of expenses paid. From this report, the user can drill down (through Group) to the *Transaction Detail* report. This report provides information on the following:

Name	Description
Average Expenses Paid	Expenses Paid divided by Claims with Expense Payments
Claims with Expense Payments	Number of Claims that have expense payments
Expense % of Total Paid	Percentage of paid expenses divided total paid
Expenses Paid	Amount paid due to the claim processing expenses

Monthly Expenses and Claim Cost

The *Monthly Expenses and Claim Cost* report provides a monthly breakdown (by Group, Adjuster, LOB, and Loss Cause) of the expenses and claims cost paid. From this report, the user can drill down (through Group) to the *Transaction Detail* report. This report provides information on the following:

Name	Description
Average Claim Cost Paid	Claim Cost Paid divided by Claims with Claims Cost Payments
Average Expenses Paid	Expenses Paid divided by Claims with Expense Payments
Claim Cost Paid	Amount paid on the claim due to cost of indemnifying the claim
Claims with Claim Cost Payments	Number of Claims that have claim cost payments

Name	Description
Claims with Expense Payments	Number of Claims that have expense payments
Expense % of Total Paid	Percentage of paid expenses divided total paid
Expenses Paid	Amount paid due to the claim processing expenses

Monthly Financial Transactions

The *Monthly Financial Transactions* report provides a monthly breakdown (by Group, Adjuster, LOB, and Loss Cause) of payments and recoveries. From this report, the user can drill down (through Group) to the *Transaction Detail* report. This report provides information on the following:

Name	Description
Average Claim Cost Paid	Claim Cost Paid divided by Claims with Claims Cost Payments
Average Expenses Paid	Expenses Paid divided by Claims with Expense Payments
Average Recovery Amount	Amount of recovery divided by Claims with Recoveries
Claim Cost Paid	Amount paid on the claim due to cost of indemnifying the claim
Claims with Claim Cost Payments	Number of Claims that have claim cost payments
Claims with Expense Payments	Number of Claims that have expense payments
Claims with Recoveries	Number of claims that have recoveries
Expense % of Total Paid	Percentage of paid expenses divided by total paid
Expenses Paid	Amount paid due to the claim processing expenses
Recovery Amount	Amount of recovery

Monthly Recoveries

The *Monthly Recoveries* report provides a monthly breakdown (by Group, Adjuster, LOB, and Loss Cause) of recoveries. From this report, the user can drill down (through Group) to the *Transaction Detail* report. This report provides information on the following:

Name	Description
Average Recovery Amount	Amount of recovery divided by Claims with Recoveries
Claims with Recoveries	Number of claims that have recoveries
Recovery Amount	Amount of recovery

Open Claim Financials

The *Open Claim Financials* report provides a summary of financial metrics for claims that are currently open, analyzed by Group, Adjuster, LOB, Loss Type, and Loss Cause. From this report, the user can drill down (through Group) to the *Transaction Detail* report. The *Open Claim Financials* report provides information on the following:

Name	Description
Average TI/Claim	Gross Total Incurred divided by Open Claim Count
Claim Cost Paid on Open Claims	Amount of claim cost paid on open claims
Expenses Paid on Open Claims	Amount of expenses paid on open Claims
Net Total Incurred	Total incurred calculated as Total Paid, plus Open Reserves minus recoveries
Open Claim Count	Number of Claims that are open
Open Exposures	Number of open exposures
Open Reserves	Amount of open reserves

Name	Description
Total Paid	Amount of total payments

Transaction Detail

The *Transaction Detail* report provides a summary of all transactions associated with a claim. This includes the transaction type and status, the cost type and category, and the check number and amount.

Special Investigation Unit (SIU) Reports

Guidewire ships ClaimCenter Standard Reporting with the following Special Investigation Unit (SIU) reports:

- Matching Payment Address
- Multiple Roled Payees
- Payees with Multiple Claims

Matching Payment Address

The *Matching Payment Address* report provides a list of the payments for which the payment address matches an address of a user on the claim. This report provides information on the user name and address, and the payment address.

Multiple Roled Payees

The *Multiple Roled Payees* report provides a list of payees who have multiple roles on a claim. From this report, the user can drill down (through the claim number) to the *Claim Detail* report.

Payees with Multiple Claims

The *Payees with Multiple Claims* report provides a list of payees that appear across multiple claims. From this report, the user can drill down (through the claim number) to the *Claim Detail* report. This report provides information on the following:

Name	Description
Claim Number	Number of the duplicate claim.
Payee Name	Name of claim payee.
Loss Date	Date of loss as reported on the claim.

chapter 8

Administering Reports

This topic describes some of the common tasks associated with administering ClaimCenter reports.

This topic includes:

Working with ClaimCenter Reporting Permissions

In the base configuration, Guidewire provides the following system permissions that control who can access the ClaimCenter report administration functions and who can view a report.

Permission	Code	Description
Administer reports	`reporting_admin`	Permission to administer report server settings.
View **Report** tab	`reporting_view`	Permission to view the **Report** tab (Standard Reporting must be enabled.)

The reporting_admin Permission

Just as with other ClaimCenter system permissions, you add the reporting permission to a role. Therefore, all users with the same role receive the same report permissions. ClaimCenter uses report permission sets (groups of one or more reports) to control which reports a user can view. Only a user with the `reporting_admin` permission can create report permission sets from the **Report Admin** screen under the **Administration** tab. The report administrator can create and delete report permission sets. Each set can contain any number of reports.

In the base configuration, Guidewire associates the `reporting_admin` permission and the `reporting_view` permission with several different roles (the Superuser role, for example). By default, Guidewire does not associate any reports with these roles. The report administrator must configure these roles by adding report permission sets to them after the initial report synchronization with the report server. As necessary, the report administrator can create additional permission groups and roles and associate them with reports.

The reporting_view Permission

A user with the `reporting_view` permission can view reports only. The report permission groups assigned to the user dictates what reports the user can view. The user accesses the reports from the **Report** tab.

By default, ClaimCenter does not display drill-down reports in the report tree, even if they appear in the report permission group. (Drill-down reports generally have an association with a separate base report. You invoke a drill-down report by clicking a link within the base report.) However, if you assign permission to view a base report that contains a drill-down report, you also need to assign a permission to view the drill-down report.

Base Configuration Report Permission Sets

Guidewire provides the following report permission sets in the base configuration:

Permission set	Assigned to...
`reportmanager`	**Manager** **Reporting Admin** **Superuser** **Reinsurance Manager**
`reportuser`	**Claims Supervisor**
`viewaggclaimmetrics`	**Claims Supervisor** **Manager** **Superuser** **Reinsurance Manager**
`viewownmetricalerts`	**Adjuster** **Claims Supervisor** **Manager** **Superuser** **Reinsurance Manager**
`viewsupmetricalerts`	**Claims Supervisor** **Manager** **Superuser** **Reinsurance Manager**

These report permission sets are empty by default. The report administrator needs to add reports to these permission sets *after* they synchronize ClaimCenter with the report server.

To set report permissions in Guidewire ClaimCenter

For a user to be able to view a report, you must add that report to a report permission set and link that permission set to a specific role. You must then assign that role to the user.

1. Log into ClaimCenter as administrator, and navigate to the **Administration** tab.
2. Select the **Report Admin** page link on the left sidebar.
3. Select the **Report tab**, then click **Sync** to synchronize the application server report templates with the report server templates. Verify that you can see the list of available reports.

   In general, any time that you change a report template on the report server, you must synchronize the application server.
4. Select the **Permission Sets** tab, and click **Add Report Permission Set**.

5. In the **Add Report Permission Set** screen, do the following:
   a. Enter a unique name for this report permission set.
   b. Select the reports to add to this permission set.
   c. Click **Update** to save your changes and create the permission set.
6. Navigate to **Administration** → **Roles**. You must link your new permission set with a role. To link your permission set with a role, either select an existing role or create a new role that you can assign to a user. (If you create a new role, you set the report permissions as you create it.)
7. After selecting a role, do the following:
   a. Open the role for editing.
   b. Scroll to the bottom of the page. ClaimCenter displays a list of report permission sets. Use the **Add** button to move a permission set from the available list to the view list.
   c. Click **Update** to save your changes.
8. Search for the user. Add the report role to the user if the user does not already have it.
9. To test, log into ClaimCenter as that user and navigate to the **Report** tab. Verify that you are able to see the reports associated with this user.

Synchronizing Reports with the InetSoft Server

Only a user with the `reporting_admin` permission can synchronize reports between the InetSoft report server and the ClaimCenter application server. You need to synchronize the reports between the two servers under the following conditions:

- Initially, after you complete your integration between the InetSoft and Guidewire applications. Otherwise, you cannot see reports within ClaimCenter and you cannot assign a report to a user role.
- Periodically, as you add, delete, or modify reports on the report server.

To synchronize reports between the two servers, navigate to the **Administration** → **Report Admin** → **Reports** tab and click **Synch**.

> **Note:** See “Optimizing the Reporting File Structure” on page 117 for a discussion on how your reporting file structure can impact the performance of the synchronization operation.

Problems Deploying a Report

In rare cases, an attempt to *redeploy* a newly deployed report from the InetSoft Report Designer (reporting server) to the ClaimCenter application server fails. The attempt to overwrite an existing report with a report of the same name fails if you do not first synchronize reports between the report server and the application server. If you do not synchronize the reports between the two servers, ClaimCenter does not add the newly deployed report to the report permission sets. In this case, ClaimCenter does not associate a user with the required overwrite permission with the newly deployed report. Thus, the overwrite attempt fails.

Configuring Reports to Access Claim Information

Many ClaimCenter reports provide hyper-text links (on the claim number) that you can click to access claim information directly within ClaimCenter itself. If this feature is active, clicking the link takes you directly to the relevant claim summary, financial or activities page. (This feature is only available in reports that contain claim numbers.)

To configure ClaimCenter reports to access claim information

1. Navigate to the following reporting directory and open `sree.properties` for editing:
   `../cc/conf/sree.properties`
2. Add the following configuration property near the top of the file:
   `APPURL=machine_name:port/app_name`
   For example, enter the following in `sree.properties`:
   `APPURL=MACHINE_NAME:11500/cc600`
3. Click **OK** to save your modification and close the file.

The report templates (`.srt`) files use the value of the `APPURL` property to generate the correct server URL. For example, in the `Claim List.srt` file, you see the following:

```
<onLoad><![CDATA[parameter.serverURL = inetsoft.sree.SreeEnv.getProperty("APPURL");]]></onLoad>
```

Administering Report Printing

InetSoft supports report printing by first generating an Adobe PDF file of the report. To actually print the report, a user must have PDF printing enabled through the browser. In Microsoft Internet Explorer, you do this by enabling a PDF add-on through the **Manage Add-ons** browser administration screen.

Note: Guidewire provides the following instructions as a guide only. For complete instructions, consult the browser documentation.

To enable PDF printing from Internet Explorer

1. Open Microsoft Internet Explorer.
2. Navigate to **Tools** → **Internet Options** → **Programs**.
3. Click **Manage Add-ons**.
4. Select *Add-ons currently loaded in Internet Explorer* from the **Show** drop-down.
5. If you do not see an entry for Adobe PDF Reader, then you must manually download and install the add-on. Consult the browser documentation on this procedure.

   Note: During the add-on installation process, it is possible that you also need to associate the PDF file type with the Adobe Reader add-on.

PDF Print Issues

As discussed, if a user attempts to print a report, then InetSoft first generates a PDF version of the report. It then downloads that file to the local machine for that user. InetSoft first estimates the PDF download time and displays the PDF print dialog after the passage of that estimated time. InetSoft uses a default value of 100 Kb/s for the download rate and the PDF file size to calculate the estimated time. However, there are problems if the PDF file size is large enough and the download speed for that user is significantly higher than 100 Kb/s. *In that case, the print dialog does not open until well after the download of the PDF completes, if at all.*

Workaround 1

If the report is large, 4 MB or greater, then it is possible that clicking the **Print** icon does not produce the **Print Options** dialog. If this is the case, you can export the report as a PDF file yourself, then issue the print command on that file.

Workaround 2

Note: For a discussion of the properties set through `sree.properties`, consult the InetSoft documentation. Specifically, refer to the *InetSoft Administration Reference*.

InetSoft provides the following property that you can set to specify the Internet connection speed for use in the internal estimation of the PDF download time. (The default value is 100000.)

```
download.speed = 100000
```

Guidewire recommends, at least initially, that you set this value to 1000000. For example:

```
download.speed = 1000000
```

To set this value, you must add it to `sree.properties`, then rebuild and redeploy `sree.war`.

IMPORTANT If a user navigates away from the report page (to another tab or window) before the print dialog opens, then there are also issues with InetSoft displaying the print dialog.

Exporting Reports in Microsoft Excel Format

It is possible to export a report using the Microsoft Excel file format. This can create problems if a report has a data cell with large amounts of data (for example, long text strings). The InetSoft default behaviour is to parse any row in the report with large data cells into multiple Excel rows. This can significantly change the original look of the report.

To force InetSoft to export a large text string into a single cell in Excel, add the following to `sree.properties`:

```
excel.rowheight.max=-1
excel.colwidth.max=-1
```

Working with the InetSoft Enterprise Manager

The following application files control report generation and the interaction between InetSoft StyleReportEE and Guidewire ClaimCenter.

File	Contains	To modify
`Repository.xml`	Repository tree in InetSoft.	Use the SREE Replet Registry.
`Query.xml`	Query logic used by the reports.	Use the Query Builder. Then, deploy a report and select the Query Registry check box.
`Datasource.xml`	Connection information database server.	Use the Query Builder. Then, deploy a report and select the Datasource Registry check box.
`*.srt` files	Reports templates.	Use the Report Designer.
`StyleReport.slr`	Report parameter sheets and beans.	Do **not** attempt to modify this file directly.

Viewing System Logs

Note: Refer to the InetSoft *Style Report Administration Reference Guide* for details on how to configure logging in InetSoft.

ClaimCenter Standard Reporting generates query log files that you can view either through the stand-alone Enterprise Manager or through the ClaimCenter **Report Admin** link. This log is useful, for example, in tracking and troubleshooting problems with the report server and the report database. In some cases, problems associated with the report server do not show in the Guidewire ClaimCenter logs, but do show in the InetSoft logs. Therefore, Guidewire recommends that you review both sets of logs, especially if a problem involves the report server.

By default, all report log messages are written to `sree.log` in the following directory:

```
StyleReportEE/Tomcat/webapps/sree/WEB-INF/classes
```

Guidewire recommends that you set the log-detail level to **Warning**.

To set the log-detail level

1. Open the Enterprise Manager.
2. Select the **Log** folder under **Main Server** to access the log properties page.
3. Click the notepad icon next to **Log Detail Level**.
4. In the **Log Detail Level** dialog, set **Detail Level** to **Warning**.
5. Click **Save** to close this dialog.

Note: As the log file (`sree.log`) can grow quite large, Guidewire recommends that you periodically delete or archive this file.

Viewing Report SQL Queries

It is possible to determine the exact SQL query used in retrieving reporting data from the reporting database by setting the Style Report logging level to **Debug**. Style Report embeds the SQL query used to retrieve the report data with other debug information and prints it to the Style Report **Log** screen and to the system console. You can view this information by navigating to **Main Server** → **Log** → **View**. It can be helpful to clear the log file before executing a new report.

Integrating Style Report with a Third-Party Logger

InetSoft Style Report provides a `Log` API that enables you to replace the default logging mechanism by a user-supplied delegate class. To integrate a third-party logging service to the reporting server, implement a class based on the `SreeLog.Delegate` interface. For complete details, refer to the InetSoft *Style Report Administration Reference Guide*, "Log".

The following abbreviated code example illustrates the use of the `SreeLog.Delegate` interface:

```
public class ThirdPartyLog implements SreeLog.Delegate {
  public void print(String msg) {
    // Implement code to send a logging string to the third-party logger
  }

  public void print(Exception e) {
    // Implement code to send any exceptions to the third-party logger
  }
}
```

After you create the log delegate, you must register it with the reporting server. Enter the full class name in **Log Delegate** field in the Enterprise Manager (**Main Server** → **Log** → **Log Delegate**). Style Report then redirects all calls to `SreeLog` to the delegate.

Reducing the Number of Cached Temporary Files

By default, InetSoft caches temporary files for 48 hours. Over a period of time, this can create a large number of temporary files stored on disk. To reduces the number of files, do one of the following:

- Within the InetSoft Enterprise Manager, navigate to **Performance** → **Cache settings** and set **Maximum Swap File Age** to 108000000, which is 30 minutes. (30 minutes is the minimum value to which you can set this value.)
- Open `sree.properties` for editing, and set the following parameter:

```
replet.cache.max.age=108000000
```

Understanding InetSoft Configuration Options

There are a number of configuration options available in the InetSoft Administration functionality, many of which are critical for integration with Guidewire ClaimCenter. Modifying these parameters incorrectly can invalidate the integration between InetSoft Style Report and Guidewire ClaimCenter. The following table lists critical

parameters that you must not modify, or modify only on instructions provided in this document. To access these parameters, open the Enterprise Manager and navigate to the **Server** tab.

Folder	Parameter	Description
Server	• Repository Servlet URL • License Keys	ClaimCenter references this URL in file `config.xml` to launch the InetSoft Report Viewer from within ClaimCenter. The ability to run the InetSoft application and access the report server requires valid license keys. You do not need to change either of the parameters.
Server → **Security**	Security	**User Defined Security Provider** displays the following value: `com.guidewire.inetsoft.cc.GWSecurityProvider` Do not change this value without instructions from Guidewire. Doing so breaks the user authentication process between Guidewire ClaimCenter and the InetSoft report server. See "Understanding Report Security" on page 37 for details of the authentication process.
Configuration → **Datasource/Lib**	• Datasource XML • Query XML • Style Report library	These are locations of files that contain some of the ClaimCenter report content. You do not need to change these values.
Presentation → **DHTML**	**Report Session Timeout** (text box)	*Set to 60 minutes.* This duplicates the session time-out value set for Guidewire ClaimCenter. Setting these two values the same minimizes the possibility of time-out errors between the two applications. You can also set this value directly in `sree.properties`: `html.session.timeout=3600000` This parameter sets the timeout period for each report viewing session (in milliseconds).
Presentation → **DHTML** → **Windows and Frames**	**Page Frame Name** (text box)	ClaimCenter uses the value `inetframe` to properly display InetSoft within ClaimCenter. Do not change this value.

See also "Working with the InetSoft Enterprise Manager" on page 109 for a description of other options that you can set in the InetSoft Enterprise Manager.

Troubleshooting Guidewire Standard Reporting

This section describes problems that you can encounter during running reports. These include:

- Problems with Report Data
- Problems with Report Auditing
- Problems Contacting the ClaimCenter Reporting Database
- Problems Opening Reports from within ClaimCenter
- Performance Issues Using F5 Proxy as a Load Balancer
- Problems Related to Tomcat
- Problems with the InetSoft Scheduler
- Problems with InetSoft SRSecurityException
- Problems with InetSoft installation on Microsoft Vista

Problems with Report Data

The following table lists issues that you might encounter with report data.

Problem	Resolution
Report returns no data	All reports return data. If one does not, it can be because there is no data available. Verify that your ClaimCenter reporting database contains data, even if it is just sample data.
Report execution appears to hang	If the data set being returned to the report is too big (15,000 or more records), the report can appear to hang. In this case, change your report criteria to reduce the number of records in the report.
Report data contains placeholder data	In some cases the report data can contain placeholder data such as series of X's for field values. This usually indicates a problem with retrieving data from the report database.

Problems with Report Auditing

The default InetSoft configuration enables report auditing by default. In some cases, the InetSoft server does not generate the reports correctly because the report server fails to save the audit record because of database issues. Guidewire recommends, therefore, that you *turn off* auditing on the reporting server.

To disable the InetSoft report auditing, you need to add the following to `CCReporting/cc/conf/sree.properties`:

```
repository.audit.enabled=false
```

Problems Contacting the ClaimCenter Reporting Database

To test that InetSoft Style Report can successfully communicate with the ClaimCenter reporting database, do the following:

1. Open InetSoft Style Report Enterprise Manager and navigate to the **Main Server** screen. This is the initial application screen.
2. Click the active link listed for **Repository Servlet URL** (usually `/sree/Examples`).
3. Log in using a valid username and password.
4. Test that you can access the ClaimCenter reports directly from the Enterprise Manager. If you can, then InetSoft Style Report is successfully interacting with the ClaimCenter reporting database.

Problems Opening Reports from within ClaimCenter

Occasionally, even if the InetSoft ClaimCenter reporting database connection is working, you cannot access reports directly from within ClaimCenter.

- If this is the issue, first verify that your login account has the correct permission to access and view reports.
- If this is not the issue, then verify that the application server URL uses the fully-qualified domain name, rather than `localhost`.

Performance Issues Using F5 Proxy as a Load Balancer

You can experience performance issues if using the F5 proxy as the load balancer in your clustered configuration. This can occur if the following conditions are true:

- The communication between the Internet Explorer browser and the F5 proxy uses Secure Socket Link (SSL).
- The communication between the F5 proxy and the report server does not use SSL.

To work-around this issue, set the following parameter in `sree.properties`:

```
ssl.enabled=true
```

If Style Report detects that the application server is using SSL, it automatically provides the client with the proper communication. However, it is possible to add SSL without the Web application actually knowing about it. This might be case, for example, if you use a proxy to add SSL. Then, you must set property `ssl.enabled` to `true` to inform Style Report. Setting this property to `true` causes the server to change its output to handle the SSL issue.

Problems Related to Tomcat

If you have problems with Tomcat, review the following:

- Check that you have correctly set the `JAVA_HOME` environment variable. Set this in `catalina.bat` (if using Tomcat) or in the system variables *before* you attempt to start the application server.
- Check that the Tomcat shutdown port and the connector port are unique on that host. If you have earlier installations on that host, these port numbers can be taken by earlier Tomcat installs.
- If the server fails to start, try using `<tomcat_home>/bin/catalina.bat run`. This command prints the server log to the console. From this, you may be able to ascertain the cause of the failure.
- The service batch file that comes with Tomcat base configuration does not update the registry properly for Java memory heap settings. Therefore, you may need to manually edit the registry to correctly set the server startup JVM parameters.

Problems with the InetSoft Scheduler

If you have problems with the InetSoft Scheduler, review the following:

- Disable InetSoft Scheduler Auto-start
- Change InetSoft Scheduler RMI Port
- Use Correct InetSoft Scheduler CLASS Path
- Use Correct CLASS Path Separator.
- Configure -Djava.awt.headless on Linux

Disable InetSoft Scheduler Auto-start

Guidewire recommends that you do *not* enable the auto-start functionality of the InetSoft Scheduler. If you enable Scheduler auto-start, the automatic restart attempts to start the Scheduler JVM if you restart the InetSoft Enterprise Manager.

This configuration can interfere with the restart of the reporting server if you do not shut down the Scheduler properly before attempting to restart the reporting server. If the Scheduler is set to start automatically and the reporting server is unable to start the Scheduler during a restart, then the reporting server refuses to start. In this case, you need to find and stop the Scheduler process manually.

In the base configuration, Guidewire disables the InetSoft Scheduler auto-start functionality by setting the following property in `sree.properties` to `false`:

```
schedule.auto.start=false
```

IMPORTANT Guidewire recommends, in general, that you *disable* the InetSoft Scheduler auto-start feature and that you only enable it on a case-by-case basis.

Change InetSoft Scheduler RMI Port

You must change the RMI port of Scheduler from its default of 1099 to something else. As this is a standard port, it can possibly be in use by another RMI application.

Use Correct InetSoft Scheduler CLASS Path

For version 10.1, use the following Scheduler `CLASSPATH`:

```
scheduler.classpath=$(sree.home);
                    $(sree.home)/../lib/axis.jar;
                    $(sree.home)/../lib/wsdl4j.jar;
                    $(sree.home)/../lib;
                    $(sree.home)/../lib/sree_pro.jar;
                    $(sree.home)/../lib/etools.jar;
                    $(sree.home)/../lib/ojdbc14.jar;
                    $(sree.home)/../lib/sqljdbc.jar;
                    $(sree.home)/../lib/gw-sree.jar
```

Use Correct CLASS Path Separator.

Verify that the Scheduler `CLASSPATH` property uses the correct class path separator for your operating system.

- For Microsoft Windows, use a semi-colon (;).
- For the different UNIX versions, use a colon (:).

Configure -Djava.awt.headless on Linux

You only need to do the following if you are running the InetSoft Scheduler on Linux and there is no X11 server running. The following examples assume that you are using a Tomcat report server.

To set `CATALINA_OPTS` in the command shell in which you are starting the report server, enter the following:

```
export CATALINA_OPTS="-Djava.awt.headless=true"
```

To set `JAVA_OPTS` in `<tomcat-root>/bin/catalina.sh`, do the following:

```
JAVA_OPTS="$JAVA_OPTS -Djava.awt.headless=true"
```

You must stop and restart the report server for these changes to take effect.

Problems with InetSoft SRSecurityException

Occasionally, as you attempt to access a report from the ClaimCenter **Report** tab, the browser displays the following exception:

```
inetsoft.sree.security.SRSecurityException: Access not allowed or session timeout error is seen.
```

This can happen if you attempt to access the InetSoft report server from two different browsers. To clear the exception, log out of the second browser, and refresh the first browser.

If this is a recurring issue, you can also disable session persistence across reporting server restarts. For example, if using Apache Tomcat as the report server, open the following file for editing:

```
...\conf\context.xml
```

Remove the comment tag from the `<Manager>` element in the following section:

```
<!-- Uncomment this to disable session persistence across Tomcat restarts -->
<!-- <Manager pathname="" /> -->
```

Thus:

```
<!-- Uncomment this to disable session persistence across Tomcat restarts -->
<Manager pathname="" />
```

Problems with InetSoft installation on Microsoft Vista

There is a limitation to the installer that InetSoft includes with versions 8 and 9 of InetSoft Style Report. This version of the installer is incompatible with the Microsoft Windows Vista operating system. It is unable to create a correct executable. The most recent Guidewire reporting installation files correct this problem. Guidewire now bundles Standard Reporting with a newer version of the installer (version 9.5), which can handle the Vista operating system.

You can, however, launch InetSoft version 8 and 9 on the Vista operating system by calling the Java classes directly. You must set the `CLASSPATH` to point to the application JAR files (`etools.jar`, `sree_pro.jar`,

design_pro.jar). Consult the InetSoft *Report Designer Guide*, specifically the section on “Starting the Designer”. (Section 5.1 in the InetSoft *Report Designer Guide Version 9.0*.)

Alternatively, you can create and run a batch file that sets a CLASSPATH variable with the required JAR files. For example, do something similar to the following in your batch file:

```
CLASSPATH=E:\StyleReportEE8\bin\;
          E:\StyleReportEE8\lib\sree_pro.jar;
          E:\StyleReportEE8\lib\design_pro.jar;
          E:\StyleReportEE8\lib\etools.jar;
          E:\StyleReportEE8\Tomcat\webapps\sree\WEB-INF\classes;
          E:\StyleReportEE8\examples\docExamples\datasource;
          E:\StyleReportEE8\examples\docExamples\design;
          E:\StyleReportEE8\bin\lax.jar;
          E:\StyleReportEE\Tomcat\webapps\sree\WEB-INF\lib\ojdbc14.jar;
          E:\StyleReportEE\Tomcat\webapps\sree\WEB-INF\lib\sqljdbc.jar;
         <PATH_TO_GWSECURITYPROVIDER>

Java -classpath %CLASSPATH% -Xms64m -Xmx512m inetsoft.sree.design.Designer2
```

chapter 9

Optimizing Report Performance

This topic includes:

Optimizing the Reporting File Structure

Periodically, the reporting administrator (or someone with the `reporting_admin` permission) needs to synchronize reports between the InetSoft report server and the ClaimCenter application server. You need to perform this task under the following conditions:

- Initially, after you complete your integration between the InetSoft and Guidewire applications. Otherwise, you do not see reports within ClaimCenter, and you are not able to assign a report to a user role.
- Periodically, as the report administrator adds, deletes, or modifies reports on the report server.

How you configure the reporting file structure can significantly impact the length of time that it takes to load these reports.

> **Note:** To synchronize reports between the two servers, navigate to the **Administration** → **Report Admin** → **Reports** tab and click **Synch**.

Guidewire Recommendations

Guidewire strongly recommends that you organize reports on the report server into folders instead of placing the report files in a *flat* file structure. Otherwise, you can experience performance issues during the synchronization process. This is due to Guidewire ClaimCenter trying to render the page while retrieving a large number of page items (reports).

Guidewire also recommends that you set the `pageSize` parameter in the `ReportsAdmin.pcf` page to fit your business needs. The `pageSize` parameter sets the number of root level nodes that the PCF `RowTree` iterator renders on the page. Therefore, in general, you would set this value to the number of report folders that you want to be

visible on a page. Thus, for ClaimCenter to display a single root report folder on a separate page, you need to set `pageSize` to `1`.

To set the number of report folders

The following procedures describes how to set the `pageSize` property in `ReportsAdmin.pcf`.

1. Open Guidewire Studio.
2. Open the **ReportsAdmin** PCF file. You can use CTRL-N to find the file.
3. Select the `RowTree reportNode` page element.
4. Navigate to the `pageSize` property at the bottom of the screen and set its value.

Structuring Reporting Views

Guidewire makes the following general recommendations for structuring SQL views in reporting:

1. Persist data through the use of *Materialized Views* (Oracle) and *Indexed Views* (MS SQL).
2. Streamline queries that use the views. This means that you only select what is necessary for that view.
3. Make use of indexed columns for query joins.
4. In a query, do not have one view join to another view. The database server is not able to optimize this type of query.

Streaming, Caching, and Clustering

Attempting to retrieve a large set of reports from InetSoft can impact reporting performance. If this is the case, Guidewire recommends that you consider the following performance optimization techniques.

Optimization	Description
Streaming	Streaming allows ClaimCenter to initially display the first portion of large result data sets without waiting for the rest. For example, if the results include several pages, you can configure InetSoft so that the first pages are *streamed* to ClaimCenter. Thus, the user can start checking results while InetSoft is still returning the remainder of the results.
Caching	Caching allows the storing of data in memory or disk so that some result can be reused later for other reports. This technique is most useful if the data does not change much or if the changes do not need to be reflected immediately. An obvious issue with this optimization technique is that excessive caching can lead to using stale data.
Clustering	Clustering (meaning having several application server nodes) is a straightforward addition of reporting nodes to spread load between multiple servers. See “Reporting in a Clustered Environment” on page 75 for information about using ClaimCenter Standard Reporting in a clustered server environment.

Note: For a discussion of these optimization techniques—streaming, caching, and clustering—refer to the InetSoft *Style Report Administration Reference Guide*.

Load Types

Depending on the type of report, it is possible that you can have a heavier load on either the report database or the application server:

- *Standard (base application) reports* generally place a heavier load on the report database.
- *Customized (or ad-hoc) reports* can place a heavier load on the application server.

Performance Tuning

For more information on Guidewire application tuning and InetSoft report tuning:

- For information on performance tuning of the Guidewire ClaimCenter application server, contact Guidewire Support.
- For information on InetSoft report performance tuning, refer to the InetSoft documentation. Specifically, refer to InetSoft *Administration Reference*, “Server Environment”. This topic also includes information on clustering and report caching.

part IV

Working with Report Development

chapter 10

Installing InetSoft Style Report Enterprise for Development

WARNING You need install InetSoft Style Report as a stand-alone application only if you plan to use the InetSoft Report Designer. Guidewire Reporting does not require a separate InetSoft installation as you access the Enterprise Manager directly through the Guidewire report portal. If you do install Style Report as a stand-alone application, **do not attempt to access or manage the ClaimCenter reports through it because this can cause harm to ClaimCenter**.

Guidewire Standard Reporting for ClaimCenter 6.0.8 works only with the InetSoft Report Designer shipped with the InetSoft Style Report, Enterprise Edition, version 10.1. If you have an earlier version of InetSoft Style Report, it will not work correctly with this ClaimCenter 6.0.8 release.

This topic includes:

Note: For detailed information about InetSoft Style Report Enterprise Edition, refer to the InetSoft documentation. If you need access to the InetSoft documentation, contact Guidewire Support.

Installing the InetSoft Application

IMPORTANT Before starting to install InetSoft, review the installation scenarios outlined in "Guidewire Standard Reporting Installation Scenarios" on page 17.

To install InetSoft Style Report, version 10.1, see the following sections.

- Acquire the InetSoft Application Software
- Install Style Report (Windows)

Again, you need do this only if you need access to the InetSoft Report Designer.

Acquire the InetSoft Application Software

Contact Guidewire Services to acquire the InetSoft installation files.

WARNING The build number of the reporting install files **must** match the build number of the Guidewire ClaimCenter installation. If the build numbers do not match, you will have problems with your reporting installation.

Install Style Report (Windows)

Perform the following steps using the InetSoft Style Report installer.

To install Style Report in a Windows environment

1. Double-click `SREE.exe` to open the Style Report, Enterprise Edition installer.
2. Click **Run**, if you the installer asks you to confirm running the software.
3. Read the **Introduction** page of the installer, then click **Next**.
4. Accept the license agreement, then click **Next**.
5. In the **License Key** box, type the license key that Guidewire provided to you along with Style Report. Click **Next**.

   If you enter an incorrect license key, the installer does not complain. However, InetSoft generates an error message during startup to indicate that there is an incorrect key. If you do not have a license key, see “Step 1: Acquire the InetSoft License Keys” on page 20.
6. Set the installation location to a location other than that used for Guidewire ClaimCenter, and then click **Next**.
7. Accept the default product icons, and then click **Next**.
8. Click **Full** to select the full application for installation, and then click **Next**.
9. Choose the JVM to use. There can be multiple choices listed, depending on the other software installed on your computer. Choose the same JDK version as that required by your installation of Guidewire ClaimCenter, and then click **Next**.
10. Enter the host name and port number of the bundled Apache Tomcat report server.
    - Enter the fully qualified domain name for the report server host (for example, `rptserver.mycompany.com`).
    - Enter a port number that does not conflict with a port in use by a Guidewire application.
11. Review the installation information for accuracy, then click **Install**.

    The Installing Style Report, Enterprise Edition progress bar opens.
12. Review the release information, then click **Next**.
13. Click **Done** to complete the installation.

Configuring InetSoft Style Report

If you have difficulty launching the Style Report Enterprise Manager, add the following line:

```
// Example: set JAVA_HOME=c:\j2sdk
set JAVA_HOME=java_2_sdk
```

to file:

```
StyleReportEE/Tomcat/bin/catalina.bat
```

This sets the location of the Java 2 SDK. (Use the same version of the Java SDK that you used during ClaimCenter installation.)

Add this line in `catalina.bat` just before the following line:

```
set JAVA_OPTS=-Xmx512m
```

To configure the Style Report Enterprise Manager parameters

IMPORTANT You only use the Windows **Start** menu to open the stand-alone InetSoft Style Report application.

1. From the Windows **Start** menu, navigate to **Style Report Enterprise Edition** → **Enterprise Manager.** As the server starts, the Enterprise Manager page appears in a browser window.
2. In the **Enterprise Manager**, log in using the following account information, then click **Login**:

   Administrator: admin

   Password: admin

   Note: Use the provided login information for the initial login. Guidewire **strongly** recommends that you change this user name and password going forward. Refer to the InetSoft documentation for details.
3. On the Main Server page, notice the server status in the **Status** box. If the status does not display **Running**, then click **Restart**.
4. In the navigation tree on the left, click **Deployment**, then select the checkbox to the left of **Enable live report deployment.**
5. In the navigation tree, click **Presentation.**
   a. Enter the appropriate date format for your locale (for example, `MM/dd/yyyy`) in the **Date Format** field.
   b. Click **Apply** before continuing.

      InetSoft does not automatically save your work if you navigate away from this page.
6. Navigate to **Presentation** → **DHTML** → **Windows and Frames.**
   c. Enter `inetframe` in the **Page Frame Name** text box. Enter the parameter exactly as written because it is case sensitive.
   d. If **Create New Window when Drilldown** is checked, uncheck it. This ensures that drill-down reports open in the same window as the initial report.
   e. Click **Apply** before continuing.
7. In the navigation tree, navigate to **Server** → **Security.**
   a. Select the **Default Security Provider** radio button. If you run InetSoft Style Report as a stand-alone application, then you can use the default security provider.
   b. Click **Apply** before continuing.

Launching the Reporting Tools

If you install InetSoft Style Report as a separate, stand-alone application, you need to launch it separately from Guidewire ClaimCenter.

To launch the InetSoft Reporting tools (Windows)

1. Navigate to **Start** → **Programs** → **Style Report Enterprise Edition.**
2. Select one of the following:
   - **Enterprise Manager**
   - **Report Designer**

To launch the InetSoft Enterprise Manager on UNIX

1. Navigate to the following directory:
   ```
   StyleReportEE/bin
   ```
2. Run the following script to open the Enterprise Manager:
   ```
   em.sh
   ```
   Note: Guidewire does not support the use of the InetSoft Report Designer on the Unix operating system.

To verify that the report server is running

Note: This procedure works only in a non-clustered environment. In a clustered environment, the status displays **Stopped**.

1. Log into the InetSoft Enterprise Manager.
2. Open the **Main Server** folder.
3. In the list of server properties, find **Status**. Verify that it is set to **Running**.

To stop the InetSoft Enterprise Manager

1. Close the InetSoft Enterprise Manager by clicking **Log Out** in the upper right-hand corner of the Enterprise Manager screen.
2. Close down the Tomcat server instance by clicking the **Close** box in the upper right-hand corner of the window in which the server runs. If prompted, click **End Now.**

chapter 11

Working with the Report Designer

To use the InetSoft Report Designer, you must have a stand-alone version of the InetSoft application. You cannot use the version of InetSoft that Guidewire bundles with Guidewire ClaimCenter. For detailed information about InetSoft Style Report Enterprise Edition, refer to the InetSoft documentation. If you need access to the InetSoft documentation, contact Guidewire Support.

This topic includes:

Setting Report Designer Configuration Parameters

To access and use the InetSoft Report Designer correctly, you need to configure the Report Designer with the correct path to the necessary connection and class files.

- See "Installing InetSoft Style Report Enterprise for Development" on page 123 for instructions on how to install InetSoft Style Report.
- See "Launching the Reporting Tools" on page 126 for details on how to launch and open the InetSoft Report Designer.

To configure the Report Designer parameters

The following procedure details setting up the Report Designer to work with Apache Tomcat. Instructions for WebSphere and WebLogic are similar.

1. Open the Report Designer. From the Windows **Start** menu, navigate to **Style Report Enterprise Edition** → **Report Designer.**

2. If this is the first time that you are running Report Designer:
   a. The setup wizard asks you to enter your license key. Type your license key, and then click **OK**. If you do not have a license key, see "Step 1: Acquire the InetSoft License Keys" on page 20.
   b. The wizard indicates that the application will create data source registry files. Click **OK**.
   c. The wizard asks if you want to set up data sources now. Click **No**.
3. On the **File** menu, select **Configure**.
4. On the **Registry** tab, select the **Local Repository** checkbox.
5. Click **Select Registry Directory**, browse to the following directory, and click **Open** to select it:
   `StyleReportEE/Tomcat/webapps/sree/WEB-INF/classes`
6. On the **Classpath** tab, do the following:
   a. Click **Add Directory** and add the following:
      `StyleReportEE/Tomcat/webapps/sree/WEB-INF/lib`.
   b. Click **Add JAR**. Navigate to the folder you just added, and add one of the following JAR files as appropriate for your database.

Database	JAR
SQL Server 2008	sqljdbc4.jar
Oracle	ojdbc14.jar

   If the JAR file for your database does not appear in the selection list, then you need to acquire a copy and manually add it to the location before continuing:
   `.../sree/WEB-INF/lib`
7. Click **OK**. You see a message stating that the changes will take effect the next time the Report Designer starts.
8. Exit the Report Designer.
   You must stop and restart the Report Designer for the application to recognize your changes.

Configuring the Data Source

You must configure the data source that the Report Designer uses. If the ClaimCenter data source does not exist, you must create it before preceding.

IMPORTANT Guidewire recommends that you *not* use the ClaimCenter production database or the ClaimCenter mirror database for report development.

To configure the Report Designer data source

1. Find the following reporting application files:
   - `datasource.xml`
   - `query.xml`

   Guidewire provides a version for both Microsoft SQL Server and Oracle. You will find these files in one of the following directories:
   report-home`/cc/conf/oracle`
   report-home`/cc/conf/sql`
2. Copy each one to the following directory:
   `../StyleReport/Tomcat/webapps/sree/WEB-INF/classes`

This overwrites the default InetSoft version of each file in that directory.

3. Open the Report Designer.
4. Open the Data Modeler (**Query** → **Data Modeler**).
5. Select the icon for your ClaimCenter database. The right-hand pane displays connection information.
6. Make the following changes in the right-hand pane:
   a. Modify the entry in the **JDBC URL** field and make it specific for your database host machine. For example, if using Microsoft SQL Sever, use some variant of the following (replacing *HOSTNAME* with the correct name):

      `jdbc:sqlserver://HOSTNAME/SQLSERVER2008`.

   b. Enter the user and password information for the database.
   c. Enter the name of the database in the **Change Default DB** field.
7. On the toolbar, click **Save (**).
8. On the toolbar, click **Test Data Source** ( ). If successful, the Data Modeler displays the message **Test is successful!** Click **OK** to close this message.
9. Expand the ClaimCenter data source root node. Verify that you see a large number of queries listed. Do not proceed until you verify that the Data Modeler can communicate to the data source correctly.
10. Click **Close** to exit the Data Modeler.
11. Exit Report Designer.

Deploying Your Reports

If you unzip and install the ClaimCenter report templates as part of a general InetSoft installation process, you install the ClaimCenter reports into the correct directory. However, if you create a new report or wish to use an updated version of a report, then you must deploy it in a separate procedure.

For example, insurance company A1 goes live with ClaimCenter, which also includes ClaimCenter Standard Reporting. The initial deployment of the reporting package copies all of the needed files into their correct location. However, after this date, to add or update reports in this environment, you must append any new or updated files to those that were copied (installed) in the initial implementation.

To do that, you must manually deploy each new or updated report using the InetSoft Report Designer. In addition, if there are any new or updated SQL views, you must load these into the reporting database. Instructions for each incremental deployment will vary based on which of the following files you need to install:

- SQL views
- `datasource.xml`
- `query.xml`
- `stylereport.srl`
- `*.srt` (report template) files

The following procedure describes the process of deploying additional or updated reports to a ClaimCenter environment that is already running reports.

To deploy additional or updated reports to a currently running environment

Use the following procedure to deploy new or updated reports. This ensures that you append new reports to the report tree and that you do not overwrite existing reports. Also, it preserves any custom reports added after the original out-of-the-box ClaimCenter implementation.

1. Unzip the new reports zip file into a temporary directory on the report server.
2. Load any new or updated SQL views to the report database, if necessary.
3. Open the InetSoft **Report Designer**.
4. Navigate to the **File** → **Configure** → **Registry** tab.
5. Note the current **Registry Directory**. (You will need this value at the end of the deployment in step 15.)
6. Select **Registry Directory**, set it to the temporary directory that you used in step 1, then click **OK** to close the dialog box.
7. Open the first report (**File** → **Open**).
8. Click **Deploy** to open the **Deploy Report** dialog box.
9. Set the correct **Server URL**, **User**, and **Password**.
10. Do one of the following:
    - *If deploying a new report*, ensure that the **Overwrite existing report** checkbox is **not** selected. (It is not likely that there is a report with this name.)

      If there is an existing report with this name, it is most likely a custom report that was not in the base ClaimCenter configuration. In this instance, rename the new report so as to not conflict with the existing report name.
    - *If deploying an updated report*, select the **Overwrite existing report** checkbox.
11. Fill out the rest of the entries:
    - **Report** - Enter appropriate values in the **Folder** and **Report Name** fields.
    - **Copy Files** - Select the following check boxes: **Query Registry** and **Report Library**.
12. Click **Deploy**.
13. Close the report
14. Open the next report and repeat the process
15. After all of the reports have been deployed, reset the **Registry Directory** to its original value.

Report Naming Guidelines

If you create custom reports outside of the standard ClaimCenter implementation, Guidewire recommends that you create a naming convention that includes all objects referenced by the custom report.

Suppose that an insurance company called A1 creates a custom report named *Accounts in California*. Guidewire recommends that you create the report and all the new objects on which the report relies using a naming convention unique to A1. The following table lists several examples.

Object	Name
Report	`A1_AccountsInCA.srt`
Query	`A1_CaliforniaAccounts`
SQL view	`A1_ccrv_ca_account`

Adhering to the recommended naming scheme can eliminate any conflicts between custom reports and incremental report deployments from Guidewire.

chapter 12

Report Localization

This topic discusses how to configure ClaimCenter Standard Reporting to display a different language other than that of the base configuration reports. Guidewire ClaimCenter uses U.S. English as the default language, both in Standard Reporting and in the application itself. To display reports in a different language, you must configure ClaimCenter Standard Reporting separately from Guidewire ClaimCenter.

This topic includes:

Note: For information on how to configure the Guidewire ClaimCenter application to display another language, see "Localizing Guidewire ClaimCenter" on page 463 in the *Configuration Guide.*

InetSoft and Localization

Note: For complete details on how to localize the InetSoft application, refer to the *InetSoft Technology Administration Reference*, especially Section 4.8 "Localization/Internationalization".

Localization is the act of translating and adapting the text in the ClaimCenter interface or reports so that the text is appropriate for use in other countries or languages.

InetSoft provides two levels of localization:

- Report portal localization—You use this localization type to specify a locale for the InetSoft portal interface (portal tabs, repository tree, Ad Hoc design interface). This type of localization also localizes report elements

(text elements, chart axis titles, table header columns, and similar items) and data model/query names shown in the **Ad Hoc** tool.

- Report localization—You use this localization type to localize most text elements in a report, as well as data source names, data model names (entity/attribute), and query (column) names.

InetSoft uses the following files to control localization:

Mapping file	Controls...
`srinter.properties`	Localization for interface text (tabs, tree nodes, Ad Hoc controls, and similar items.)
`SreeBundle.properties`	Localization for report text (text in tables, charts, and similar items) and data model/query names

See Also

- For information on localizing Guidewire ClaimCenter, see "Localizing Guidewire ClaimCenter" on page 463 in the *Configuration Guide*.

Registering a Locale in InetSoft Enterprise Manager

The first step in localizing the InetSoft application is to register the language and country codes with the InetSoft Enterprise Manager.

To register language and country codes with the Report server

1. Open the Enterprise Manager from within the ClaimCenter report portal.
2. Navigate to **Configuration** → **Localization**.
3. Click **Add**, then enter add the language and country code for your default language. If your default language is U.S. English, enter `en` for the language code and `US` for the country code.
4. Click **Apply**. Your language and country codes appear in the **Localization** table.
5. Click **Add** again, and enter the codes for your second locale, using the same codes that you used in naming your localization file. For example, if this was Canadian French, you enter `fr` and `CA`.
6. Click **Apply** to save this information to the **Localization** table.

Setting the Report Currency Display

Unless you configure InetSoft to do otherwise, the InetSoft reports display the local client currency instead of the actual transaction and reporting currency. This means that the InetSoft reports displays the default currency symbol of the operating system on the client machine rather than the currency symbol used in the actual reports.

This can lead to a situation in which the local client currency is the U.S. dollar and the default application currency is the Euro. Thus, as the client currency uses the $ sign and the application currency uses the € symbol, the reports would show all the currency numbers in the wrong currency.

IMPORTANT Setting the default reporting currency is different from setting the default ClaimCenter application currency. To set the default ClaimCenter currency, use ClaimCenter configuration parameter `DefaultApplicationCurrency`. See "Application Configuration Parameters" on page 35 in the *Configuration Guide* for information on the ClaimCenter configuration parameters.

To configure a default reporting currency

1. Open `sree.properties` for editing. You can find this file in the following location:

   ```
   ClaimCenterReporting/cc/conf
   ```

2. Find the following two properties. Typically, they exist directly as part of the ClaimCenter-related properties at the top of the file.

   ```
   language
   country
   ```

3. Set these values to a locale (language and country) that uses the desired currency symbol.

The default values are `en` and `English`, which displays a $ sign for the currency symbol used in the English/United States locale.

Setting different currency display values. The following table describes some of the possibilities in configuring localized currency displays.

Language/Country	Sree properties	Symbol
English/United States	`language=en` `country=US`	$
English/Great Britain	`language=en` `country=GB`	£
French/France	`language=fr` `country=FR`	€
German/Germany	`language=de` `country=DE`	€
Russian/Russia	`language=ru` `country=RU`	руб
Japanese/Japan	`language=ja` `country=JP`	¥

Thus, to show the British pound symbol, enter the following for the English/Great Britain.

```
language=en
country=GB
```

To show the Euro symbol, use a country symbol for any country that uses the Euro.

- For example, for French/France, use:

  ```
  language=fr
  country=FR
  ```

- Or, for German/Germany, use:

  ```
  language=de
  country=DE
  ```

Configuring Localized Reports in InetSoft Report Designer

InetSoft supports the use of several different types of tables in constructing reports (SRTs, or Style Report Templates). These tables can be either *List* tables (the default table style) or *Freehand* tables. Refer to the InetSoft documentation for a discussion of these table types. Especially, refer to the *Report Designer Guide*.

Guidewire recommends that you use List tables if you plan to localize your reports. It is possible to apply localization to Freehand tables. However, in practice, it is more difficult to set up localization initially using Freehand tables. It is also more difficult to maintain reports that use Freehand-type tables.

Applying Localization to Tables that Bind To Logical Models

If you intend to use List tables that bind to logical models, there is nothing else that you need to do to have the table use localization. (List tables are the default table style.) However, if you choose to use Freehand tables that bind to logical models, then you must perform the following steps to apply localization:

1. Open the report that you want to localize, and select a Freehand table.
2. Click on the column header, then select **Format** from the right-click menu to display the formatting options at the bottom of the screen.
3. Select the **Data** tab, then select the **Text** radio button.
4. In the **Text** field, enter the localized name (for example, `Policy Number`).
5. From the main **Edit** menu, select **Localization**, then navigate to the table in the **Localization** dialog.
6. In the **Localization** dialog, click on the column that you modified earlier (for example, **Policy Number**).
7. Enter a value in the in **Text ID** field (for example, `Policy Number`).
8. Click **OK** to save your work and close the **Localization** dialog.
9. At the bottom of the screen, select the **Format** tab, then select the **none** radio button.
10. From the main **File** menu, select **Preview.**

    The column now appears as **Policy #** due to the following entry is in the `SreeBundle` file:

    ```
    Policy\ Number=Policy #
    ```

    If you view the **Data** tab, you see that the entry now reads as **Policy #**. The same holds true for **Edit** → **Localization**.

Applying Localization to Tables Based on Queries

Action	Description
Change a List-type table to a Freehand-type table (1)	You can change a List-type table to a Freehand-type table by moving a column in the table. • Select the table. • Click a single time on the right-most column header in the report. • Then, perform a right mouse-click **Column** → **Move Left**. Notice that as you do this, the cell representing the data in the table changes from **XXXX**, for a text field, to the data object (**Item.Amount**, for example). However, if you change the column order using **Bind Data**, then the table remains a List-type table. • Select the table. • Perform a right mouse-click **Bind Data**. • Select the **Columns** tab and highlight the column that you want to move. • Then, use the **Move Up** or **Move Down** arrow to move the column to the desired position.
Change a List-type table to a Freehand-type table (2)	You can also change a List-type table to a Freehand-type table by inserting a column into the table. • First, select the table. • Then, perform a right mouse-click **Column** → **Insert Column**. If you add a column to a table using **Bind Data**, then the table remains a List-type table. • Select the table. • Perform a right mouse-click **Bind Data**. • Then, select the **Columns** tab and click **Create Formula**.
Revert a table to a List-type table	You can revert the table back to a List-type by selecting the table and performing a right mouse-click **Table** → **Restore Default**.

To use localization with a Freehand table, you must give each column header a data type of **Text**:

- Click on a column header, perform a right mouse-click **Format**, and select the **Data** tab.
- Then, select the **Text** radio button and enter in the name of the column in the text area.
- After you set the column data type, you can assign the column a **TextID** to use with localization.
- Select **Edit** → **Localization** from the main menu, then navigate to the column.

Setting the InetSoft Calendar Widget

It is possible to localize the calendar widget that InetSoft uses in the report parameter sheets. To do so, do the following:

- Create a localized `srinter.properties` file for a specific locale. For example, for French (France), you need to create a `srinter_fr_FR.properties` file and place it in the following location:
  ```
  webapps\sree\WEB-INF\classes\inetsoft\util
  ```
- Modify the entries in the file that correspond to calendar widget labels and text so that they conform to the desired language.

After creating your locale-specific `srinter*.properties` file, you need to modify the entries in the file that relate to the calendar widget. These are items similar to the following (for French, France):

- `Mon=Lundi`
- `Tue=Mardi`
- `Weds=Mercredi`
- `...`
- `January=Janvier`
- `February=Février`
- `March=Mars`
- `...`

Generating Localized ClaimCenter Reports

In the base configuration, ClaimCenter reports support U.S. English only. However, it is possible to add support for other locales. This is a multi-step process:

1. First, make your reports localizable using the separate InetSoft Report Designer. See “Configuring Localized Reports in InetSoft Report Designer” on page 135 for details of this process.
2. Then, set up specific localization files and add them to the report server.
3. Finally, configure the language and country codes in the InetSoft Enterprise Manager.

If the application has different locales configured, the report administrator has access to the replets (report templates) in all locales from within the InetSoft Enterprise Manager. ClaimCenter simulates this behavior. From within ClaimCenter:

- The report administrator can synchronize the reports of all locales and assign them to report group permissions as well.
- A report user can change the locale of the display from within ClaimCenter. This also updates the locale for the report. You can change the locale by using the locale link at the top of the screen.

IMPORTANT Guidewire does **not** provide translated or localized versions of the display keys (`display.properties`) file used with Guidewire Standard Reporting. You must provide your own translated version of this file.

To add support for a specific locale

1. Create your localized reports as described in "Configuring Localized Reports in InetSoft Report Designer" on page 135. You do this in the external, stand-alone InetSoft Report Designer.
2. Navigate to *report-home*/cc/conf and make copies of the following existing files:
   - repository.xml: contains the U.S. English version of the replet definition
   - SreeBundle_en_US.properties: contains display strings in U.S. English
3. Name your new files using the standard convention:
   - repository_*LanguageCode_CountryCode*.xml
   - SreeBundle_*LanguageCode_CountryCode*.properties

   The convention for locale file names is to append *LanguageCode_CountryCode* to the end of the file name, before the file extension. This is the standard naming convention that Java uses for locale support. For example, for locale file repository_fr_CA.xml, fr is the language code for French and CA is country code for Canada.
4. Modify your new files as necessary to define your locale.
5. Open sree.properties for editing. (You can find this file in the same directory as repository.xml.) Modify the following items, replacing *xx_YY* with the appropriate language and country codes:
   - replet.repository.file=$(sree.home)/repository_*xx_YY*.xml
   - sree.bundle=SreeBundle (changing the file name from SreeBundle_en_US)

   For example, if using Canadian French and U.S. English, add the following to sree.properties:

   ```
   replet.repository.file=$(sree.home)/repository_fr_CA.xml
   ```
6. Add your language and country codes in the **Localization** tab in the InetSoft Enterprise Manager. For details, see "Registering a Locale in InetSoft Enterprise Manager" on page 134.

IMPORTANT If you modify or add files, you must rebuild sree.war and redeploy it to the report server.

IMPORTANT You must rerun the appropriate cc_xx_views.sql script if you make a change to the data model configuration that affects a typelist used in ClaimCenter Standard Reporting. Specifically, you *must* run this script each and every time that you add a new language locale to your configuration. For more information, see "Step 3: Create the Required Database Reporting Views" on page 21.

Configuring Direct Access for ClaimCenter Localized Reports

Most Guidewire users view reports through the ClaimCenter **Reports** tab. However, it is possible for those with the requisite privilege to view these reports directly through the InetSoft Enterprise Manager by entering a direct URL for the InetSoft application. If you configure your Guidewire installation for localized reports, using a direct access URL does not provide the correct report data.

For localized reports accessed through the **Reports** tab, ClaimCenter passes the Guidewire user locale to InetSoft so that it can correctly generate report typelist names based on the locale. In a direct login, ClaimCenter does not pass the user locale.

- If the report or parameter sheet *does use* a typelist name, then it does not return a value as it is dependent on a locale.
- If a report or parameter sheet *does not* use a typelist name, then it generates the correct data.

To view valid data in localized reports with direct URL login

To correctly access localized report data through direct URL login, then you must define the required locale in the InetSoft Enterprise Manager directly.

1. First, register any locale that you add on the **Localization** page of the Enterprise Manager.
   a. Log into the Enterprise Manager and navigate to the **Configuration** → **Localization**.
   b. In the **Localization** page, click **Add** and enter your locale information. For example:

Language	Country	Locale Label
en	US	English (United States)
fr	FR	French (France)

   This action adds the following entries to `sree.properties`:

```
locale.available=en_US\:fr_FR
```

2. Next, configure the InetSoft Enterprise Manager for the desired locale. You can do this at the portal level, which localizes the Enterprise Manager interface (tab and button names, for example). You can also do this at the report level, which localizes the report elements themselves. In general, however, you need to create mapping files for each desired locale. Consult the InetSoft documentation for details. Specifically, consult the section on *Localization/Internationalization* in the *InetSoft Technology Administration Reference*.

   Note: If you correctly set up and configure multiple locales (the default is United States English), then you see an additional drop-down menu labeled **Locale** on the Enterprise Manager login screen.

3. Select the desired locale from the **Locale** drop-down list in the Enterprise Manager login screen before you attempt to view localized reports.

Scheduled Reports

If you have a scheduled task that involves localized reports, then you also need to set the appropriate locale for that task. You set the locale as you schedule the task in the Enterprise Manager **Schedule** tab. If you do not set a specific locale for a task, then the locale information for the task is `null` and all localized ClaimCenter base configuration reports show blank.

If you do set a locale for a scheduled task, then you see something similar to the following in `schedule.xml` (for example):

```
<Task name="su:Task1" owner="su" enabled="true" delNotRun="false" locale="fr_FR">
```

Localization and the Reporting Database

Microsoft SQL Server requires that you store Unicode multi-byte character data in `NVARCHAR` columns. (A SQL Server database that uses a single-byte format stores character data in `VARCHAR` columns.) If you use SQL Server as the reporting database, then you need to run different scripts to create the reporting views for Unicode multi-byte data.

Installation topic “Step 3: Create the Required Database Reporting Views” on page 21 describes the scripts that you must run to set up the reporting views on the reporting database. Consult that topic for complete details. However, to summarize:

Script location. You can find the required SQL scripts in the following ClaimCenter reporting installation directory:

```
report-home/resources/cc
```

Single-byte data. If you use a SQL Server reporting database that stores character data as single-byte data, then run the following scripts:

```
cc_sqlserver_views.sql
cc_sqlserver_storedprocedures.sql
```

Multi-byte data. If you use a SQL Server reporting database that stores character data as Unicode multi-byte data, run the following scripts:

```
cc_sqlserver_views_unicode.sql
cc_sqlserver_storedprocedures_uncode.sql
```

Note: See "Configuring SQL Server for ClaimCenter" on page 23 in the *Installation Guide* for a discussion of the `<database>` element in `config.xml`. Especially, see the discussion on how to set the `unicodecolumns` attribute on the `<database>` element. You need to set this attribute specifically for SQL Server databases that use a multi-byte format to store character data.

IMPORTANT You only need to run the specialized Unicode multi-byte scripts if you use Microsoft SQL Server as the reporting database.