



# **JBoss Enterprise SOA Platform 5**

## **Eclipse Guvnor Tools Reference Guide**

This guide is for developers .

Edition 5.3.1



# JBoss Enterprise SOA Platform 5 Eclipse Guvnor Tools Reference Guide

---

This guide is for developers .

Edition 5.3.1

David Le Sage

Red Hat Engineering Content Services

dlesage@redhat.com

## Legal Notice

Copyright © 2013 Red Hat, Inc..

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## Abstract

This is a guide to using the JBDS Eclipse Guvnor plug-in..

## Table of Contents

<b>PREFACE</b> .....	<b>2</b>
<b>CHAPTER 1. PREFACE</b> .....	<b>3</b>
1.1. BUSINESS INTEGRATION	3
1.2. WHAT IS A SERVICE-ORIENTED ARCHITECTURE?	3
1.3. KEY POINTS OF A SERVICE-ORIENTED ARCHITECTURE	3
1.4. WHAT IS THE JBOSS ENTERPRISE SOA PLATFORM?	4
1.5. THE SERVICE-ORIENTED ARCHITECTURE PARADIGM	4
1.6. CORE AND COMPONENTS	4
1.7. COMPONENTS OF THE JBOSS ENTERPRISE SOA PLATFORM	5
1.8. JBOSS ENTERPRISE SOA PLATFORM FEATURES	5
1.9. FEATURES OF THE JBOSS ENTERPRISE SOA PLATFORM'S JBOSSSES COMPONENT	5
1.10. TASK MANAGEMENT	6
1.11. INTEGRATION USE CASE	6
1.12. UTILISING THE JBOSS ENTERPRISE SOA PLATFORM IN A BUSINESS ENVIRONMENT	7
<b>CHAPTER 2. INTRODUCTION</b> .....	<b>8</b>
2.1. INTENDED AUDIENCE	8
2.2. AIM OF THE GUIDE	8
2.3. REQUIREMENTS AND INSTALLATION	8
2.4. GUVNOR TOOLS	9
<b>CHAPTER 3. TASKS</b> .....	<b>10</b>
3.1. CREATING A NEW GUVNOR CONNECTION	10
3.2. GETTING LOCAL COPIES OF GUVNOR FILES	10
3.3. MANAGING GUVNOR RESOURCES	11
<b>CHAPTER 4. REFERENCE</b> .....	<b>13</b>
4.1. GUVNOR REPOSITORY EXPLORING PERSPECTIVE	13
4.2. STARTING THE GUVNOR CONNECTION WIZARD	13
4.3. CONFIGURING THE GUVNOR CONNECTION WIZARD	14
4.4. RESOURCE FROM GUVNOR WIZARD	14
4.5. GUVNOR REPOSITORIES VIEW	15
4.6. GUVNOR RESOURCE HISTORY VIEW	15
4.7. CONFIGURING THE GUVNOR RESOURCE HISTORY VIEW	16
4.8. GUVNOR PREFERENCES	16
4.9. GUVNOR REPOSITORY CONNECTION PREFERENCES	16
4.10. LOCAL GUVNOR REPOSITORY RESOURCE DECORATION PREFERENCES	17
<b>APPENDIX A. REVISION HISTORY</b> .....	<b>18</b>

## **PREFACE**

# CHAPTER 1. PREFACE

## 1.1. BUSINESS INTEGRATION

In order to provide a dynamic and competitive business infrastructure, it is crucial to have a service-oriented architecture in place that enables your disparate applications and data sources to communicate with each other with minimum overhead.

The JBoss Enterprise SOA Platform is a framework capable of orchestrating business services without the need to constantly reprogram them to fit changes in business processes. By using its business rules and message transformation and routing capabilities, JBoss Enterprise SOA Platform enables you to manipulate business data from multiple sources.

[Report a bug](#)

## 1.2. WHAT IS A SERVICE-ORIENTED ARCHITECTURE?

### Introduction

A *Service Oriented Architecture* (SOA) is not a single program or technology. Think of it, rather, as a software design paradigm.

As you may already know, a *hardware bus* is a physical connector that ties together multiple systems and subsystems. If you use one, instead of having a large number of point-to-point connectors between pairs of systems, you can simply connect each system to the central bus. An *enterprise service bus* (ESB) does exactly the same thing in software.

The ESB sits in the architectural layer above a messaging system. This messaging system facilitates *asynchronous communications* between services through the ESB. In fact, when you are using an ESB, everything is, conceptually, either a *service* (which, in this context, is your application software) or a *message* being sent between services. The services are listed as connection addresses (known as *end-points references*.)

It is important to note that, in this context, a "service" is not necessarily always a web service. Other types of applications, using such transports as File Transfer Protocol and the Java Message Service, can also be "services."



### NOTE

At this point, you may be wondering if an enterprise service bus is the same thing as a service-oriented architecture. The answer is, "Not exactly." An ESB does not form a service-oriented architecture of itself. Rather, it provides many of the tools that can be used to build one. In particular, it facilitates the *loose-coupling* and *asynchronous message passing* needed by a SOA. Always think of a SOA as being more than just software: it is a series of principles, patterns and best practices.

[Report a bug](#)

## 1.3. KEY POINTS OF A SERVICE-ORIENTED ARCHITECTURE

These are the key components of a service-oriented architecture:

1. the *messages* being exchanged
2. the *agents* that act as service requesters and providers
3. the *shared transport mechanisms* that allow the messages to flow back and forth.

[Report a bug](#)

## 1.4. WHAT IS THE JBOSS ENTERPRISE SOA PLATFORM?

The JBoss Enterprise SOA Platform is a framework for developing enterprise application integration (EAI) and service-oriented architecture (SOA) solutions. It is made up of an enterprise service bus (JBoss ESB) and some business process automation infrastructure. It allows you to build, deploy, integrate and orchestrate business services.

[Report a bug](#)

## 1.5. THE SERVICE-ORIENTED ARCHITECTURE PARADIGM

The service-oriented architecture (SOA) consists of three roles: requester, provider, and broker.

### Service Provider

A service provider allows access to services, creates a description of a service and publishes it to the service broker.

### Service Requester

A service requester is responsible for discovering a service by searching through the service descriptions given by the service broker. A requester is also responsible for binding to services provided by the service provider.

### Service Broker

A service broker hosts a registry of service descriptions. It is responsible for linking a requester to a service provider.

[Report a bug](#)

## 1.6. CORE AND COMPONENTS

The JBoss Enterprise SOA Platform provides a comprehensive server for your data integration needs. On a basic level, it is capable of updating business rules and routing messages through an Enterprise Service Bus.

The heart of the JBoss Enterprise SOA Platform is the Enterprise Service Bus. JBoss (ESB) creates an environment for sending and receiving messages. It is able to apply “actions” to messages to transform them and route them between services.

There are a number of components that make up the JBoss Enterprise SOA Platform. Along with the ESB, there is a registry (jUDDI), transformation engine (Smooks), message queue (HornetQ) and BPEL engine (Riftsaw).



---

[Report a bug](#)

## 1.7. COMPONENTS OF THE JBOSS ENTERPRISE SOA PLATFORM

- A full Java EE-compliant application server (the JBoss Enterprise Application Platform)
- an enterprise service bus (JBoss ESB)
- a business process management system (jBPM)
- a business rules engine (JBoss Rules)
- support for the optional JBoss Enterprise Data Services (EDS) product.

[Report a bug](#)

## 1.8. JBOSS ENTERPRISE SOA PLATFORM FEATURES

### The JBoss Enterprise Service Bus (ESB)

The ESB sends messages between services and transforms them so that they can be processed by different types of systems.

### Business Process Execution Language (BPEL)

You can use web services to orchestrate business rules using this language. It is included with SOA for the simple execution of business process instructions.

### Java Universal Description, Discovery and Integration (jUDDI)

This is the default service registry in SOA. It is where all the information pertaining to services on the ESB are stored.

### Smooks

This transformation engine can be used in conjunction with SOA to process messages. It can also be used to split messages and send them to the correct destination.

### JBoss Rules

This is the rules engine that is packaged with SOA. It can infer data from the messages it receives to determine which actions need to be performed.

[Report a bug](#)

## 1.9. FEATURES OF THE JBOSS ENTERPRISE SOA PLATFORM'S JBOSS ESB COMPONENT

The JBoss Enterprise SOA Platform's JBossESB component supports:

- Multiple transports and protocols
- A listener-action model (so that you can loosely-couple services together)

- Content-based routing (through the JBoss Rules engine, XPath, Regex and Smooks)
- Integration with the JBoss Business Process Manager (jBPM) in order to provide service orchestration functionality
- Integration with JBoss Rules in order to provide business rules development functionality.
- Integration with a BPEL engine.

Furthermore, the ESB allows you to integrate legacy systems in new deployments and have them communicate either synchronously or asynchronously.

In addition, the enterprise service bus provides an infrastructure and set of tools that can:

- Be configured to work with a wide variety of transport mechanisms (such as e-mail and JMS),
- Be used as a general-purpose object repository,
- Allow you to implement pluggable data transformation mechanisms,
- Support logging of interactions.



### IMPORTANT

There are two trees within the source code: **`org.jboss.internal.soa.esb`** and **`org.jboss.soa.esb`**. Use the contents of the **`org.jboss.internal.soa.esb`** package sparingly because they are subject to change without notice. By contrast, everything within the **`org.jboss.soa.esb`** package is covered by Red Hat's deprecation policy.

[Report a bug](#)

## 1.10. TASK MANAGEMENT

JBoss SOA simplifies tasks by designating tasks to be performed universally across all systems it affects. This means that the user does not have to configure the task to run separately on each terminal. Users can connect systems easily by using web services.

Businesses can save time and money by using JBoss SOA to delegate their transactions once across their networks instead of multiple times for each machine. This also decreases the chance of errors occurring.

[Report a bug](#)

## 1.11. INTEGRATION USE CASE

Acme Equity is a large financial service. The company possesses many databases and systems. Some are older, COBOL-based legacy systems and some are databases obtained through the acquisition of smaller companies in recent years. It is challenging and expensive to integrate these databases as business rules frequently change. The company wants to develop a new series of client-facing e-commerce websites, but these may not synchronise well with the existing systems as they currently stand.

The company wants an inexpensive solution but one that will adhere to the strict regulations and security requirements of the financial sector. What the company does not want to do is to have to write and maintain “glue code” to connect their legacy databases and systems.

The JBoss Enterprise SOA Platform was selected as a middleware layer to integrate these legacy systems with the new customer websites. It provides a bridge between front-end and back-end systems. Business rules implemented with the JBoss Enterprise SOA Platform can be updated quickly and easily.

As a result, older systems can now synchronise with newer ones due to the unifying methods of SOA. There are no bottlenecks, even with tens of thousands of transactions per month. Various integration types, such as XML, JMS and FTP, are used to move data between systems. Any one of a number of enterprise-standard messaging systems can be plugged into JBoss Enterprise SOA Platform providing further flexibility.

An additional benefit is that the system can now be scaled upwards easily as more servers and databases are added to the existing infrastructure.

[Report a bug](#)

## **1.12. UTILISING THE JBOSS ENTERPRISE SOA PLATFORM IN A BUSINESS ENVIRONMENT**

Cost reduction can be achieved due to the implementation of services that can quickly communicate with each other with less chance of error messages occurring. Through enhanced productivity and sourcing options, ongoing costs can be reduced.

Information and business processes can be shared faster because of the increased connectivity. This is enhanced by web services, which can be used to connect clients easily.

Legacy systems can be used in conjunction with the web services to allow different systems to “speak” the same language. This reduces the amount of upgrades and custom code required to make systems synchronise.

[Report a bug](#)

## CHAPTER 2. INTRODUCTION

### 2.1. INTENDED AUDIENCE

This book is aimed at developers who wish to learn how to use Eclipse Guvnor Tools. It contains information about installation, tasks, references and resources.

[Report a bug](#)

### 2.2. AIM OF THE GUIDE

This book aims to teach users how to quickly and efficiently utilize Eclipse Guvnor Tools. It contains information about how to perform basic tasks such as creating connections as well as information about tools and configurations. There are tasks to perform with the Connection wizard and details about how to configure preferences.

[Report a bug](#)

### 2.3. REQUIREMENTS AND INSTALLATION

1. To get started with Guvnor Tools, you should have the following:
  - JBoss Developer Studio, which can be downloaded through the [RedHat customer portal](#).
  - JBoss EAP with Guvnor repository deployed.
2. To deploy JBoss EAP with the Guvnor repository, download [JBoss Enterprise BRMS](#). When downloading you may choose either web-based Guvnor to deploy to your JBoss Server or Standalone Guvnor with JBoss AS.
3. If you use JBoss AS 5 and Guvnor Tools doesn't work when deployed to the server, perform the following steps:
  1. Remove the JBoss Security libraries (**authorization-spi-2.0.2.CR6.jar**, **identity-\***, **acl-spi-2.0.2.CR6.jar**) from drools WAR file (not the app server itself). The JBoss Security libraries are provided in AS 5.
  2. Remove the JAXP libraries (**xercesImpl-2.8.0.jar**, **xml-apis-2.\***)
  3. Remove the EL libraries (**el-\***)

When these steps have been performed, Guvnor should operate as normal. During deployment, you will see the following exception in the log which indicates that the experimental JBoss Security based ACL integration has defined some EJB3/JPA features in Guvnor that have not been configured. It looks like this:

```
ERROR [AbstractKernelController] Error installing to Start:
name=persistence.unit:unitName=#ACL state=Create
java.lang.RuntimeException: Specification violation [EJB3 JPA
6.2.1.2] - You have not defined a non-jta-data-source
for a RESOURCE_LOCAL enabled persistence context named: ACL
```

[Report a bug](#)

## 2.4. GUVNOR TOOLS

The Guvnor repository is a location where certain artifacts (such as rules and SOA policy definitions) are controlled (“governed”) by policies defined by the deployment environment. The purpose of the EGT (Eclipse Guvnor Tools) is to enable access to resources held by the Guvnor repository. Limited capabilities for reading, writing, adding and removing Guvnor repository resources are provided in the EGT.

[Report a bug](#)

## CHAPTER 3. TASKS

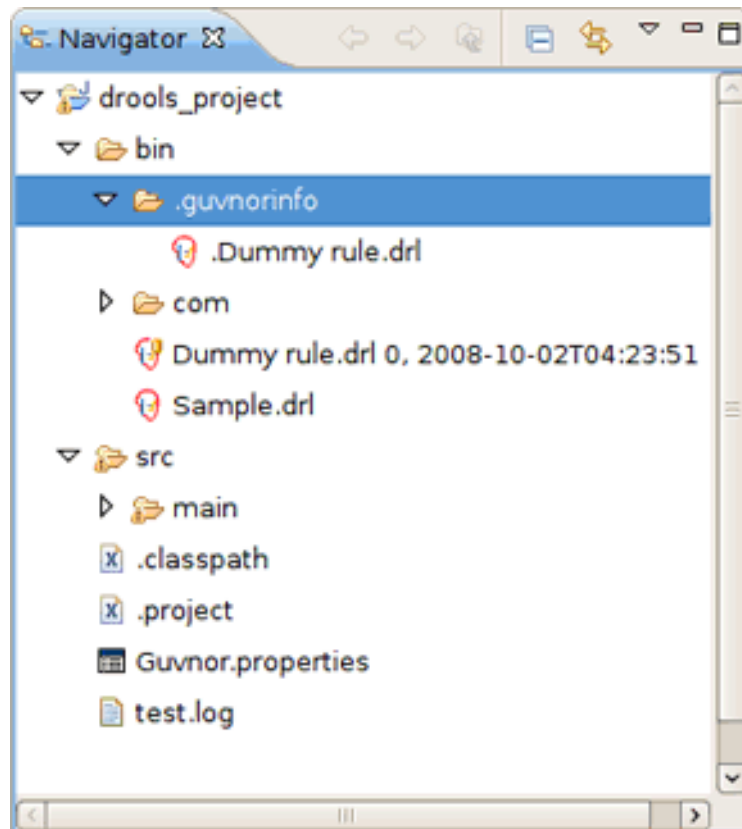
### 3.1. CREATING A NEW GUVNOR CONNECTION

1. Open a new connection.
2. Start the **Guvnor Connection wizard** in one of the following ways:
  - o Select **File** → **New** → **Other** → **Guvnor** → **Guvnor repository location** within Eclipse
  - o Go to the **Guvnor Repositories** view using the drop-down menu
  - o Use the menu button

[Report a bug](#)

### 3.2. GETTING LOCAL COPIES OF GUVNOR FILES

1. To get local copies of Guvnor repository resources, you can drag-and-drop them from the Guvnor Repositories View. You can also perform this action in the Guvnor Wizard.
2. Local copies of Guvnor repository files will be created. The EGT will set an association between the local copy and the master file in the repository. You should not alter this information. The association allows for operations such as update and commit in synchronization with the master copy held in the Guvnor repository.



**Figure 3.1. Diagram**

The EGT decorates local resources associated with Guvnor repository master copies. This decoration appears in Eclipse views conforming to the Eclipse Common Navigator framework,

such as the Eclipse **Resource Navigator** and the Java **Package Explorer**. The **Dummy rule.drl** file with the decoration in the **Resource Navigator**. The Guvnor icon decorator is on the top right of the file image, and the Guvnor revision details are appended to the file name. You can change the location of these in the **Local Guvnor Repository Resource Decoration Preferences**.

3. You can see that the **Dummy rule.drl** file is associated with a Guvnor repository resource, and the local copy is based on revision 0, with a **02-10-2008, 4:21:53** date/time stamp. However, the file **Sample.drl** is not associated with a Guvnor repository file.
4. The EGT adds a property page to the standard Eclipse properties dialog. Use it to view the specific Guvnor repository, the location within the repository, the version (date/time stamp) and the revision number.

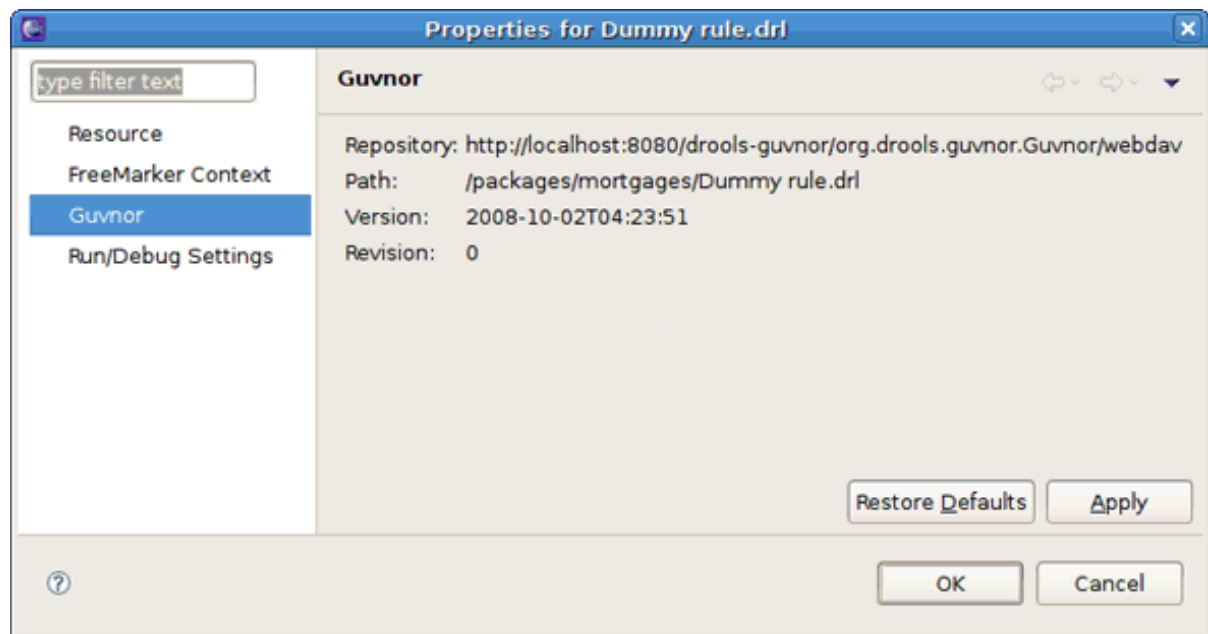


Figure 3.2. Properties

[Report a bug](#)

### 3.3. MANAGING GUVNOR RESOURCES

The EGT provides a number of actions (available through the Guvnor context menu on files) for working with files, both those associated with Guvnor repository master copies and those not associated. The actions are:

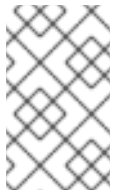
#### Update Action

Available for Guvnor resources that are not synchronized with the Guvnor repository master copies. These resources may not have synchronized due to local changes and/or changes to the master copy. Performing the Update action replaces the local file contents with the current contents from the Guvnor repository master copies.

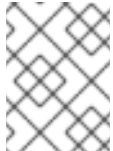
#### Add Action

Available for local files that are not associated with a Guvnor repository master copy. Choosing the Add action launches the Add in Guvnor wizard. The first page of the wizard asks for the selection of the target Guvnor repository and provides the option of creating a new Guvnor repository connection.

Once the target Guvnor repository is chosen, the wizard asks for the folder location to add the selection files. Clicking on the Finish button adds the selected files to the Guvnor repository and creates an association between the local and Guvnor repository files.

**NOTE**

The snapshot folder in the Guvnor repository is read-only for EGT, and hence not visible as a candidate location in this wizard. The Guvnor repository web administration tools must be used to add snapshot content.

**NOTE**

The wizard will not allow existing Guvnor repository files to be overwritten. Another target location must be chosen.

**Commit Action**

Enabled for Guvnor repository associated files that have local changes. The Commit action will write the changes back to the associated Guvnor repository files and update the association for the new revision created.

If a local change is based on an older revision of a file than is currently in the Guvnor repository, the Commit action will ask whether you wish to overwrite the current version in the Guvnor repository with the local content. When such conflicts occur, you should use the Eclipse Guvnor version tools and the Eclipse standard tools to determine the differences and merge content based on the current version.

**Show History Action**

Populates the Guvnor Resource History View with revision history for the selected file.

**Compare with Version Action**

Enabled for singular Guvnor repository associated files. This action first opens a wizard asking for the version for comparison (with the local file contents). Once the revision is selected, the action opens the Eclipse Compare editor (in a read-only mode) which shows the differences in the two versions.

**Switch to Version Action**

Enabled for singular Guvnor repository associated files. It first prompts for the selection of a version. Once the version is selected, the Switch to Version action replaces the local file contents with those from the revision selected.

**Delete Action**

Enabled for one or more Guvnor repository associated files. After the action has been confirmed, the Delete action removes the files in the Guvnor repository and deletes local metadata for the Guvnor repository association.

**Disconnect Action**

Enabled for one or more Guvnor repository associated files. Removes local metadata from the Guvnor repository association.

[Report a bug](#)



## CHAPTER 4. REFERENCE

### 4.1. GUVNOR REPOSITORY EXPLORING PERSPECTIVE

1. In the Eclipse workbench menu, select **Window** → **Open Perspective** → **Other** to view the perspective list.
2. Choose the perspective you wish to use. Eclipse standard views such as **Properties** and **Resource Navigator** are useful.

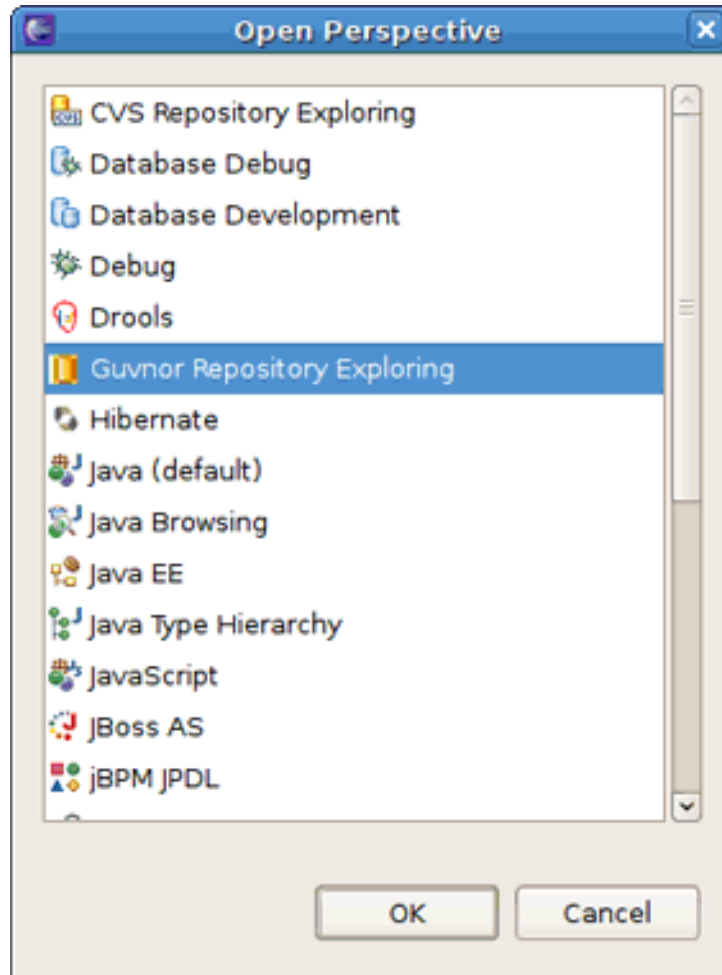


Figure 4.1. Perspective list

[Report a bug](#)

### 4.2. STARTING THE GUVNOR CONNECTION WIZARD

1. To start the **Guvnor Connection wizard**, open Eclipse and select **File** → **New** → **Other** → **Guvnor** → **Guvnor repository location**.
2. Alternatively, open the **Guvnor Repositories** and click the menu button in the drop-down list.

[Report a bug](#)

### 4.3. CONFIGURING THE GUVNOR CONNECTION WIZARD

1. Default values appear in the **Location**, **Port** and **Repository** fields. You can edit these fields using their corresponding text boxes.
2. To input a Guvnor repository URL into the respective fields, drag and drop it into the **Location** field.



#### IMPORTANT

When using Eclipse tooling, you will not be able to access Guvnor data if this Guvnor is deployed on a secured HTTP (HTTPS).

3. The EGT calls the Guvnor repository at various times, such as when it is determining if any resource updates are available. If you use session authentication, the authentication dialog will appear at different times during the Eclipse workbench session, depending on what actions you take. We recommend saving the authentication information in the Eclipse key-ring.

Click on **save username and password** to save your details into the Eclipse workbench's key-ring file.



#### NOTE

If the authentication information is not stored in the key-ring, then the EGT uses a session authentication. This means that the credentials supplied are only used during the lifetime of the Eclipse workbench instance.



#### NOTE

If an authentication failure error occurs, you can retry the same operation and supply different authentication information.



#### NOTE

The Eclipse key-ring file is distinct from key-ring files found in some platforms such as Mac OS X and many forms of Linux. Thus, sometimes if you access a Guvnor repository outside the EGT, the key-ring files might become outdated, and you will be unexpectedly prompted for your credentials in Eclipse. This is nuisance, but in this case you can simply re-enter the appropriate username and password.

[Report a bug](#)

### 4.4. RESOURCE FROM GUVNOR WIZARD

1. This wizard allows you to copy files to your local workspace from the Guvnor repository. Open the wizard by selecting **File** → **Import** → **Guvnor** → **Resource from Guvnor** or **File** → **New** → **Other** → **Guvnor** → **Resource from Guvnor**.
2. You will be prompted to enter the source Guvnor repository. You can also create a new Guvnor repository connection.



3. Once the source Guvnor repository is chosen, the wizard prompts for the resources to be copied.
4. Finally, choose the target location for the files in the local workspace.
5. On completion the wizard copies the selected files from the Guvnor repository to the local workspace. If a file with the same name already exists in the destination, you will be prompted to rename it.

[Report a bug](#)

## 4.5. GUVNOR REPOSITORIES VIEW

The **Guvnor Repositories** view contains tree structures representing the contents of Guvnor repositories. You can perform the following actions under the resources in the **Guvnor Repositories** view:

**Table 4.1. Guvnor Repositories View Actions**

Action	Description
Create a new Guvnor repository connection	Use the wizard to create a new connection by selecting <b>File</b> → <b>New</b> → <b>Other</b> → <b>Guvnor</b> → <b>Guvnor repository location</b> in Eclipse.
Remove a Guvnor repository connection	Click <b>Delete</b> (  ) to remove a repository connection.
Refresh Guvnor repository resources.	Use the <b>Refresh</b> context menu item to reload the tree for the selected node.
"Drill-into" functionality	Use icons on the toolbar such as the <b>Home</b> , <b>Go Back</b> and <b>Go Into</b> buttons to navigate deeply nested structures.
<b>Go Home</b> button (  )	Returns the tree to the top-level structure.
Selecting a file in the Guvnor repository	Doing this causes the Eclipse <b>Properties</b> view to display the details of that file.

[Report a bug](#)

## 4.6. GUVNOR RESOURCE HISTORY VIEW

The **Guvnor Resource History** view displays details about the revision history of the files you have selected. The **Guvnor Resource History** view is populated by **Show History** actions in either the

local **Guvnor** context menu or in the context menu for a Guvnor repository file in the Guvnor Repositories View.

[Report a bug](#)

## 4.7. CONFIGURING THE GUVNOR RESOURCE HISTORY VIEW

1. Select **Show History** in the local **Guvnor** context menu. Alternatively, select it in the context menu for a Guvnor repository file in the Guvnor Repositories View.
2. The **Guvnor Resource History** view is refreshed to display the revision history.
3. To open an Eclipse read-only editor with the revision contents, double-click on a revision row or select the **Open (Read only)** option from the context menu.
4. Click on the **Save As...** option when a file is open in a read-only editor to save a local writable copy of the contents. (This will not associate the file created with its Guvnor source.)

[Report a bug](#)

## 4.8. GUVNOR PREFERENCES

The EGT provides a preference page in the **Guvnor** category. The preferences cover two categories:

- Guvnor repository connections
- Local Guvnor repository resource decorations

[Report a bug](#)

## 4.9. GUVNOR REPOSITORY CONNECTION PREFERENCES

There are two preferences that can be set for Guvnor repository connections. These are used when creating new connections:

**Table 4.2. Guvnor Repository Connection Preferences**

Preference	Description
Default Guvnor repository URL template	Simplifies the process of creating additional similar connections by changing part of the field, such as the host name.
Authentication information	Defines whether or not authentication information should be saved in the Eclipse platform key-ring by default. As with the Guvnor repository URL template, this decision can be determined when creating the connection.

---

[Report a bug](#)

## 4.10. LOCAL GUVNOR REPOSITORY RESOURCE DECORATION PREFERENCES

This category of preferences provided by the EGT can be used to define how the decoration of local resources associated with Guvnor repository resources is presented. Since the Guvnor repository is not a substitute for a SCM, and since SCM tools in Eclipse tend to decorate local resources, it is useful to be able to control how the EGT decorates its local resources to avoid conflicts with SCM packages.

In the **File Decoration** section of the preference page, you can choose the location (Top right, Top left, Bottom right, Bottom left) of the decoration icon, or you can choose not to display it. In the **Text** section, you can format the Guvnor metadata which is appended to the file names.

- Whether to show an indicator (>) when the local file has changes not committed back to the Guvnor repository
- Whether to show the revision number
- Whether to show the date/time stamp

Any changes to these preferences take effect immediately upon clicking the **Apply** and then **OK** buttons.

[Report a bug](#)

## APPENDIX A. REVISION HISTORY

**Revision 5.3.1-30.400**

**2013-10-31**

**Rüdiger Landmann**

Rebuild with publican 4.0.0

**Revision 5.3.1-30**

**Wed Feb 20 2013**

**CS Builder Robot**

Built from Content Specification: 7132, Revision: 375314