

Eclipse Guvnor Tools Reference Guide



Version: 5.1.0.trunk

1. Introduction	1
1.1. Guvnor Tools Key Features	1
1.2. How to start with Guvnor Tools	1
1.3. Other relevant resources on the topic	2
2. Functionality Overview	3
2.1. Guvnor Perspective	3
2.2. Guvnor Connection Wizard	3
2.3. Guvnor Repositories View	6
2.4. Local Copies of Guvnor Files	9
2.5. Actions for Local Guvnor Resources	11
2.5.1. Update Action	12
2.5.2. Add Action	12
2.5.3. Commit Action	13
2.5.4. Show History Action	14
2.5.5. Compare with Version Action	14
2.5.6. Switch to Version Action	16
2.5.7. Delete Action	16
2.5.8. Disconnect Action	17
2.6. Guvnor Resource History View	17
2.7. Importing Guvnor Repository Resources	19
3. Guvnor Preferences	23
3.1. Guvnor Repository Connection Preferences	24
3.2. Local Guvnor Repository Resource Decoration Preferences	24
4. Conclusion	25

Introduction

The purpose of this document is to describe briefly the functionality present in the [Eclipse Guvnor Tools](#) (EGT) for Drools 5. While not intended as a comprehensive reference, there should be enough detail included for early adopters using these tools.

The [Guvnor repository](#) is not intended as a Source Code Management (SCM) solution, and the [EGT](#) are not intended to be Eclipse “team provider” extensions or replacements. Rather, 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](#) is then to enable access to resources held by the Guvnor repository, so they can be used in development. Thus, limited capabilities for reading, writing, adding, and removing Guvnor repository resources are provided in the [EGT](#).

1.1. Guvnor Tools Key Features

The following table lists all valuable features of the [Guvnor Tools](#).

Table 1.1. Key Functionality of Guvnor Tools

Feature	Benefit	Chapter
Guvnor Repository View	The purpose of the view is to enable access to Guvnor repository resources in a standard tree format	Guvnor Repositories View
Guvnor Connection Wizard	The wizard helps to create a connection to a Guvnor repository.	Guvnor Connection Wizard
Guvnor Resource History View	This view shows revisions of specific resources available in the repository.	Guvnor Resource History View
Resources from Guvnor Wizard	The wizard helps to get local copies of Guvnor repository resources.	Importing Guvnor Repository Resources
A number of actions for working with files	The Guvnor actions (available through the Guvnor context menu on files) are provided for working with files, both those associated with Guvnor repository master copies and those not associated.	Actions for Local Guvnor Resources

1.2. How to start with Guvnor Tools

[Guvnor Tools](#) is a part of the [JBoss Tools](#) project. Thus to get started with [Guvnor Tools](#), you should have the following:

- Eclipse 3.4.x with Jboss Tools bundle of Eclipse plugins installed. How to install JBoss Tools onto Eclipse you can find in the "[JBoss Tools Installation](#)" [../GettingStartedGuide/html_single/index.html#JBossToolsInstall] section.
- JBoss Server 4.2 or higher with Guvnor repository deployed. Thus, you should download [Guvnor CR1](#) [http://download.jboss.org/drools/release/5.0.0.25561.CR1/drools-5.0.0.CR1-guvnor.zip] and extract it to the deploy directory of the server.

Now refer to the "[Functionality Overview](#)" section to find out what you can do with Guvnor and Eclipse synchronisation tool.

1.3. Other relevant resources on the topic

- Guvnor [wiki page](#) [http://www.jboss.org/community/wiki/Guvnor]
- Drools on [JBoss.org](#) [http://www.jboss.org/drools/]
- [JBoss Tools Home Page](#) [http://www.jboss.org/tools/]
- [The latest JBossTools/JBDS documentation builds](#) [http://download.jboss.org/jbosstools/nightly-docs/]
- [JBossTools/JBDS release documentation](#) [http://docs.jboss.org/tools/]

Functionality Overview

This chapter will introduce you to the [Guvnor Repository Exploring perspective](#) and give an overview on all functionality the [Guvnor Tools](#) provides.

2.1. Guvnor Perspective

The [Guvnor Repository Exploring perspective](#) contains two views supplied by [EGT – Repository Explorer](#) and [Version History](#), that will be the center of most interaction with Guvnor, and Eclipse standard views such as [Properties](#) and [Resource Navigator](#) that are also useful.

While each of these views can be opened and positioned independently within an Eclipse workbench, the [Guvnor perspective](#) provides a convenient method of getting a suggested layout. In the Eclipse workbench menu, choose [Window > Open Perspective > Other](#) to get the perspective list:

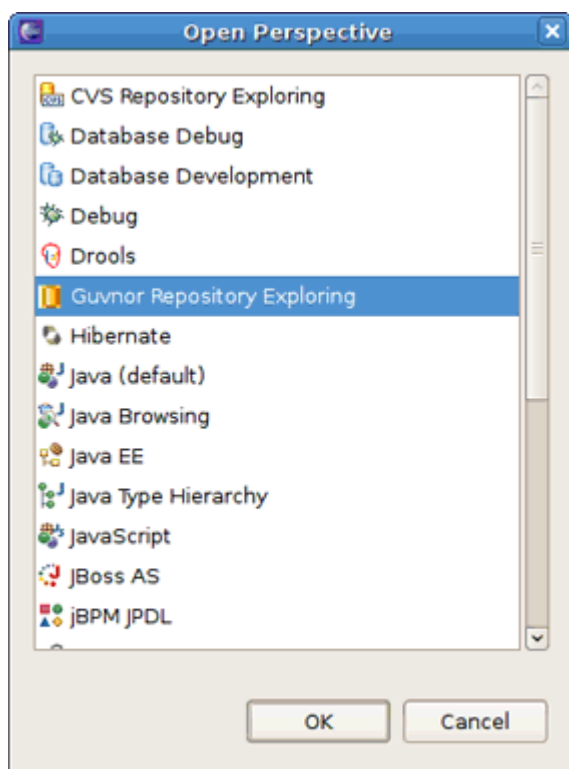


Figure 2.1. Enabling the Guvnor Repository Perspective

And then choose [Guvnor Repository Exploring](#). This opens the [Guvnor perspective](#).

2.2. Guvnor Connection Wizard

After opening the [Guvnor perspective](#), the first task is to make a connection to a Guvnor repository. This is handled by the [Guvnor Connection wizard](#). This wizard appears in a number of places

within the [EGT](#) (as detailed below), but in this section we will cover only the two most basic entry points.

The [Guvnor Connection wizard](#) can be started in the following ways:

- using the Eclipse menu *File > New > Other > Guvnor > Guvnor repository location*

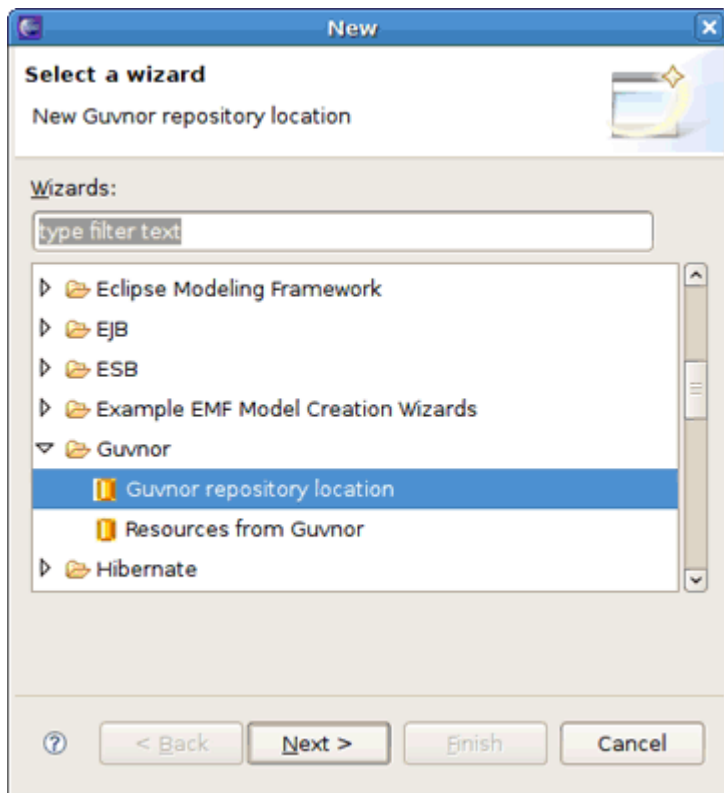


Figure 2.2. New Guvnor Repository Location

- in the [Guvnor Repositories view](#) using the drop-down menu

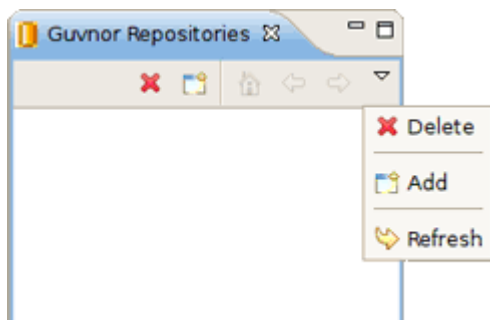



Figure 2.3. Adding New Guvnor Connection

- using the menu button ()

Choosing either of these will start the [Guvnor Connection wizard](#).



Figure 2.4. Guvnor Connection Wizard

Default values appear in the *Location*, *Port*, and *Repository* fields (See the [“Guvnor Preferences”](#) section below for details about how to change these default values.) Of course, any of these fields can be edited by typing in the corresponding text box. Drag-and-drop or paste into the *Location* field of a typical Guvnor repository URL such as:

<http://localhost:8080/drools-guvnor/org.drools.guvnor.Guvnor/webdav>

Results in the URL being parsed into the respective fields as well. The authentication information (user name and password) can optionally be stored in the Eclipse workbench's key-ring file based on the selection of [Save user name and password](#).



Note:

If the authentication information is not stored in the key-ring, then the [EGT](#) uses a session authentication, what means that the credentials supplied are used only for the lifetime of the Eclipse workbench instance.

If authentication information is not stored in the key-ring or the authentication information (key-ring or session) is not valid, the [EGT](#) will prompt for authentication information when it has to access the Guvnor repository.

If authentication fails, the [EGT](#) will retry once and then issue an authentication failure error.



Tip:

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

Note that the [EGT](#) calls the Guvnor repository at various times, such as when determining if resource updates are available. Thus if you use session authentication, the authentication dialog will appear at different times during the Eclipse workbench session, depending on what actions you take. For ease of use, we recommend saving the authentication information in the Eclipse key-ring.



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 unsynchronized and you will be unexpectedly prompted for authentication in Eclipse. This is nuisance, but your usual credentials should apply in this case.

2.3. Guvnor Repositories View

The [Guvnor Repositories view](#) contains tree structures for Guvnor repository contents.

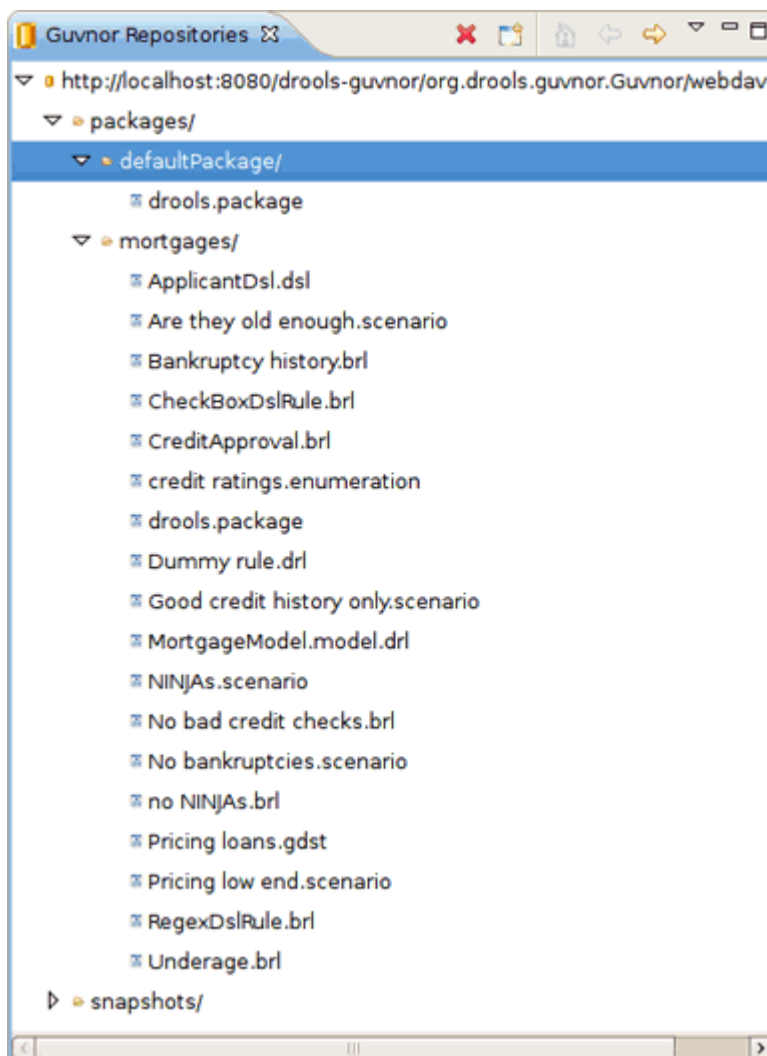






Figure 2.5. Guvnor Repositories View

You can perform the following actions under the resources in the [Guvnor Repositories view](#):

- create a new Guvnor repository connection. How to do this is describe above in the ["Guvnor Connection Wizard"](#) section.
- remove a Guvnor repository connection. Use the Delete button () in the tool-bar or the [Delete](#) option in the context menu to remove a repository connection.
- refresh Guvnor repository resorces. Use the [Refresh](#) context menu item to reload a tree content for the selected node.
- make use of "drill-into" functionality. It's represented by a number of tool-bar/context menu items such as [Go Home](#) (

, [Go](#) [Back](#) (

) and [Go](#) [Into](#) (

).

Drill-down is useful when working with deeply nested tree structures and when you wish to concentrate on only branch of the tree. For example, drilling into the *"defaultPackage"* node changes the tree view to:

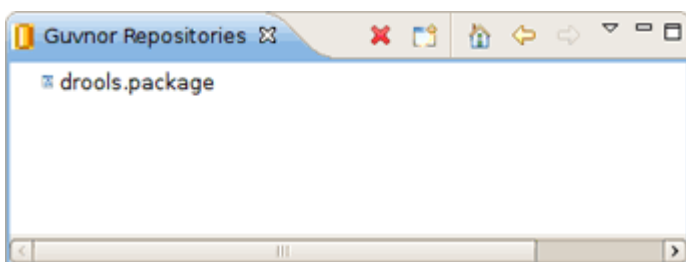



Figure 2.6. Going Into the "defaultPackage"

Clicking on the [Go Home](#) button (

) or selecting [Go Home](#) in the context menu returns the tree to the top-level structure shown in the previous picture above.

There are a number of operations that can be performed on Guvnor repository files. Selecting a file in the Guvnor repository causes the Eclipse [Properties view](#) to update with details about that file:

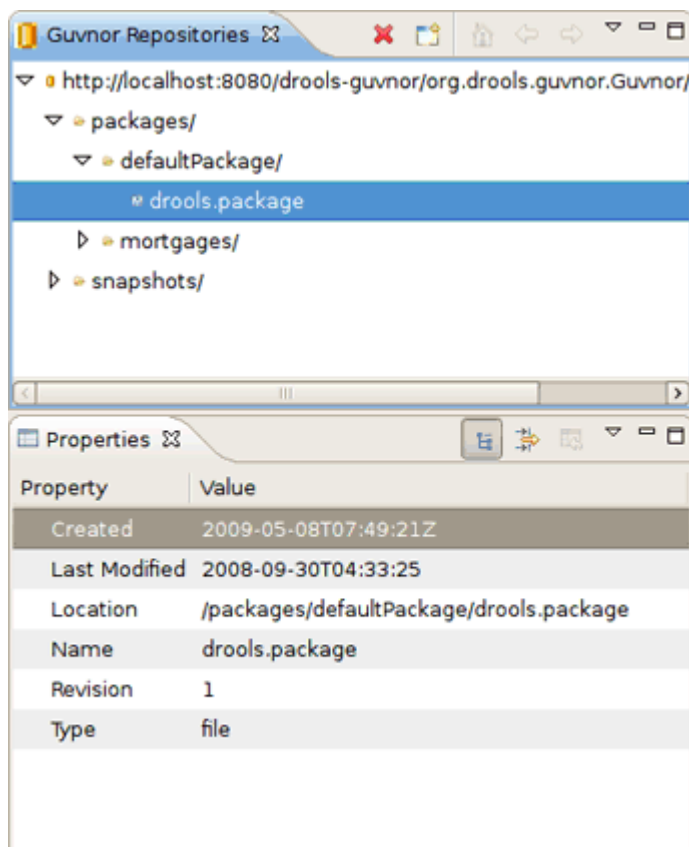


Figure 2.7. Guvnor Repository Files Properties

2.4. Local Copies of Guvnor Files

As mentioned in the ["Introduction"](#), the main purpose of the [EGT](#) is to allow development using resources held in a Guvnor repository. There are two method of getting local copies of Guvnor repository resources:

1. Drag-and-drop from the [Guvnor Repositories view](#)
2. Using the [Import from Guvnor wizard](#), as described further in this document

When local copies of Guvnor repository files are created, the [EGT](#) sets an association between the local copy and the master file in the repository. This information is kept in the (normally) hidden [.guvnorinfo](#) folder in the local project and, like all metadata, should not be changed by end users.

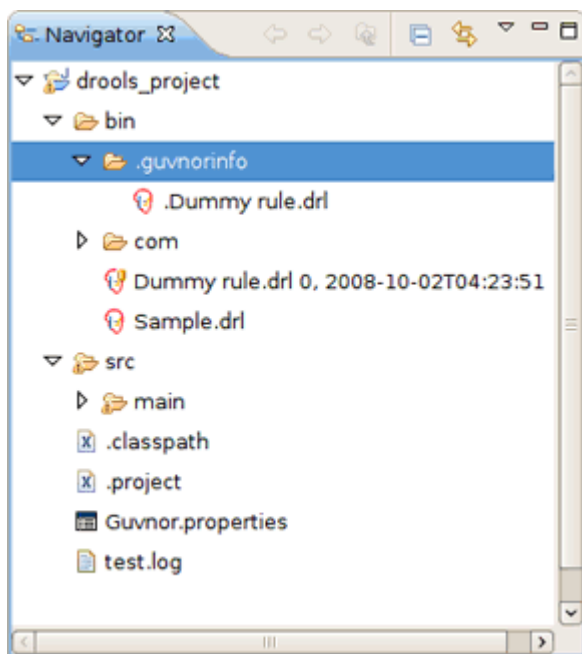


Figure 2.8. .guvnorinfo metadata

The association allows for operations such as update and commit in synchronization with the master copy held in the Guvnor repository.

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](#). On the image above you can see 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. (The presence/location of these can be changed the [Guvnor Preferences](#).)

Here we see that, [Dummy role.drl](#) 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. The file [Sample.drl](#), however, is not associated with a Guvnor repository file. Further details about the association can be found in the standard Eclipse properties page, accessed via the [Properties](#) option in the context menu:

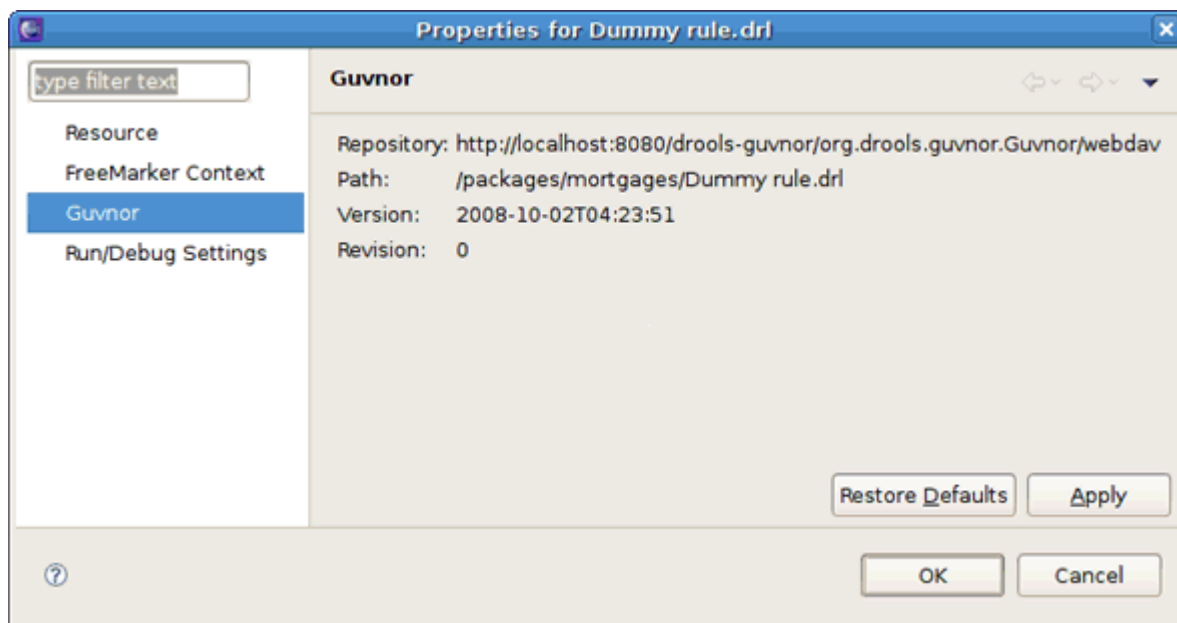


Figure 2.9. Association Details

The [EGT](#) contributes a property page to the standard Eclipse properties dialog, the contents of which are shown above. It displays the specific Guvnor repository, the location within the repository, the version (date/time stamp) and the revision number.

2.5. Actions for Local 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](#)
- [Add](#)
- [Commit](#)
- [Show History](#)
- [Compare with Version](#)
- [Switch to Version](#)
- [Delete](#)
- [Disconnect](#)

Each of these actions is described below.

2.5.1. Update Action

The [Update](#) action is available for one or more Guvnor resources that are not in synchronization with the Guvnor repository master copies. These resources would not be in synchronization because either/both

1. There are local changes to these resources
2. The master copies have been changed in the Guvnor repository

Performing the [Update](#) action replaces the local file contents with the current contents from the Guvnor repository master copies (equivalent to [Switch to version](#) for the latest version).

2.5.2. Add Action

The [Add](#) action is available for one or more local files that are not associated with a Guvnor repository master copy. Choosing the [Add](#) action launches the [Add to Guvnor wizard](#):

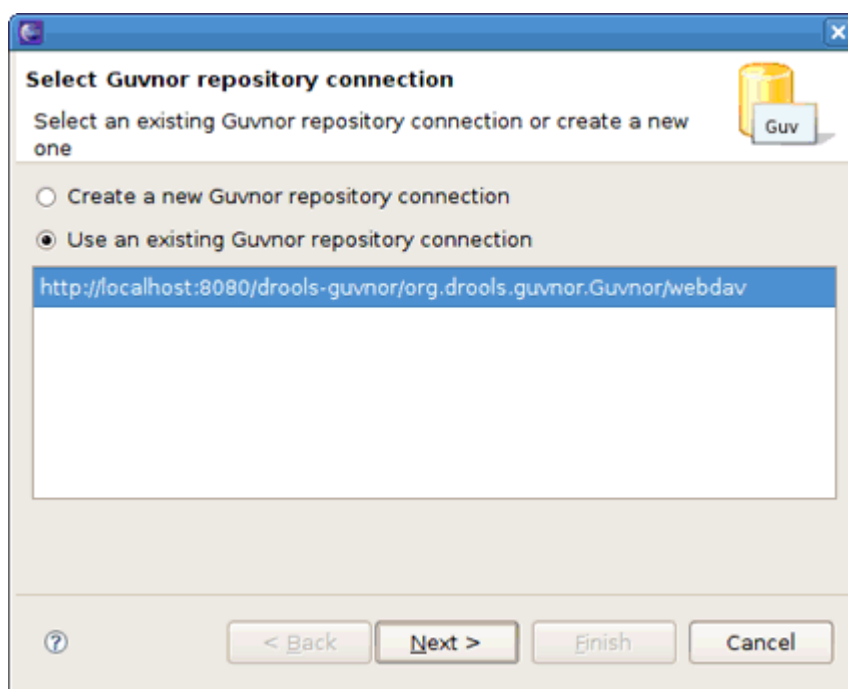


Figure 2.10. Add to Guvnor Wizard

The first page of the wizard asks for the selection of the target Guvnor repository and gives the choice to create a new Guvnor repository connection (in which case the second page is the same as the [Guvnor Connection wizard](#)). Once the target Guvnor repository is chosen, the wizard then asks for the folder location to add the selection files:

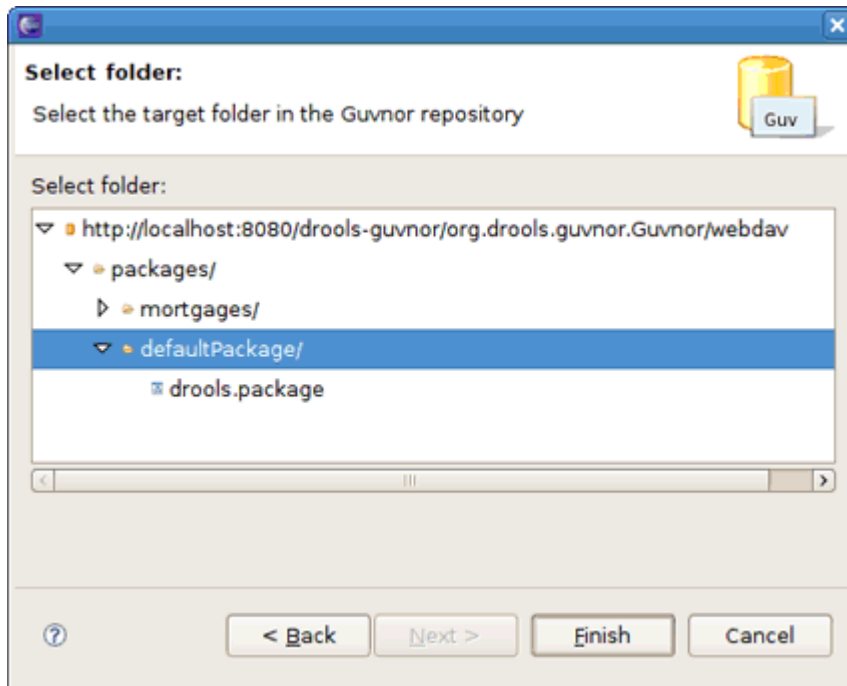


Figure 2.11. Selecting a Target Folder

Here the *"defaultPackage"* folder is selected as the destination location.



Note:

Note that 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.

Clicking on *Finish* adds the selected files to the Guvnor repository and creates an association between the local and Guvnor repository files.



Note:

Note that the wizard will not allow for overwrite of existing Guvnor repository files. Another target location must be chosen.

2.5.3. Commit Action

The *Commit* action is enabled for one or more Guvnor repository associated files that have local changes. The *Commit* action will write the local 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 that is currently in the Guvnor repository (for example, someone else changed the same file), then the *Commit* action will

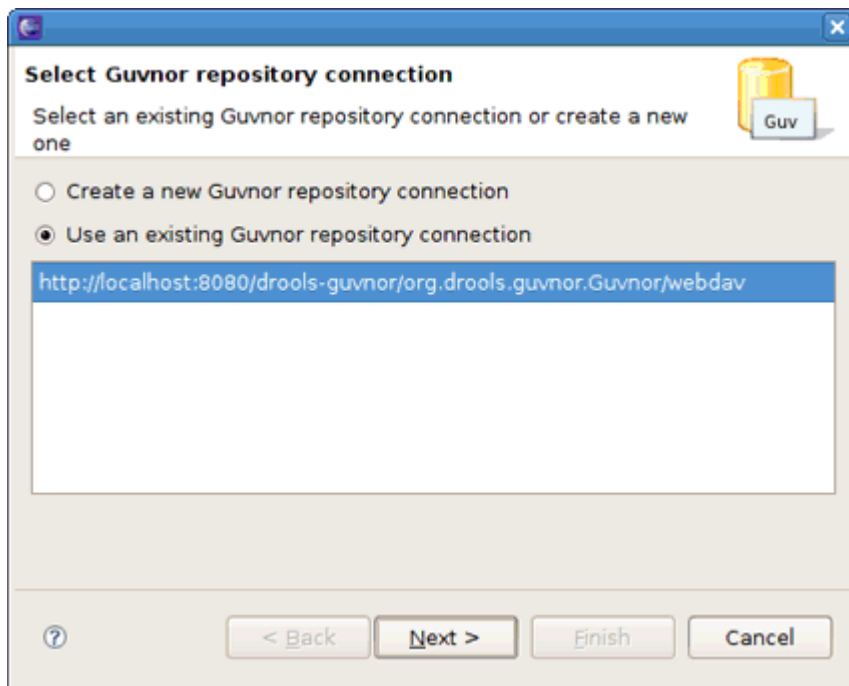


Figure 2.12. Add to Guvnor Wizard

ask whether you wish to overwrite the current version in the Guvnor repository with the local content. When such conflicts occur, however, you should use the Eclipse Guvnor version tools, along with Eclipse standard tools, to determine the differences and merge content based on the current version in the Guvnor repository.

2.5.4. Show History Action

The *Show History* action is enable for one Guvnor repository associated file and causes the [Guvnor Resource History view](#) to be populated with revision history for the selected file.

2.5.5. Compare with Version Action

The *Compare with Version* action is enabled for one Guvnor repository associated file. This action first opens a wizard asking for the version for comparison (with the local file contents):

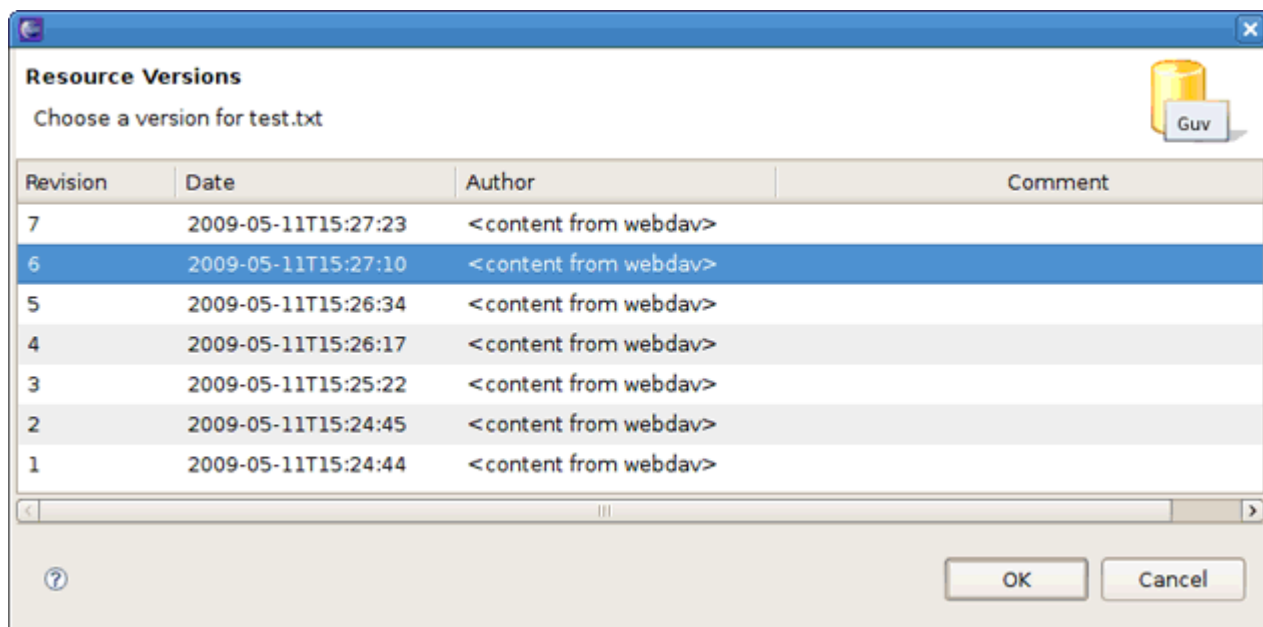


Figure 2.13. Compare with Version Wizard

Once the revision is selected, the action opens the Eclipse [Compare editor](#) (read-only):

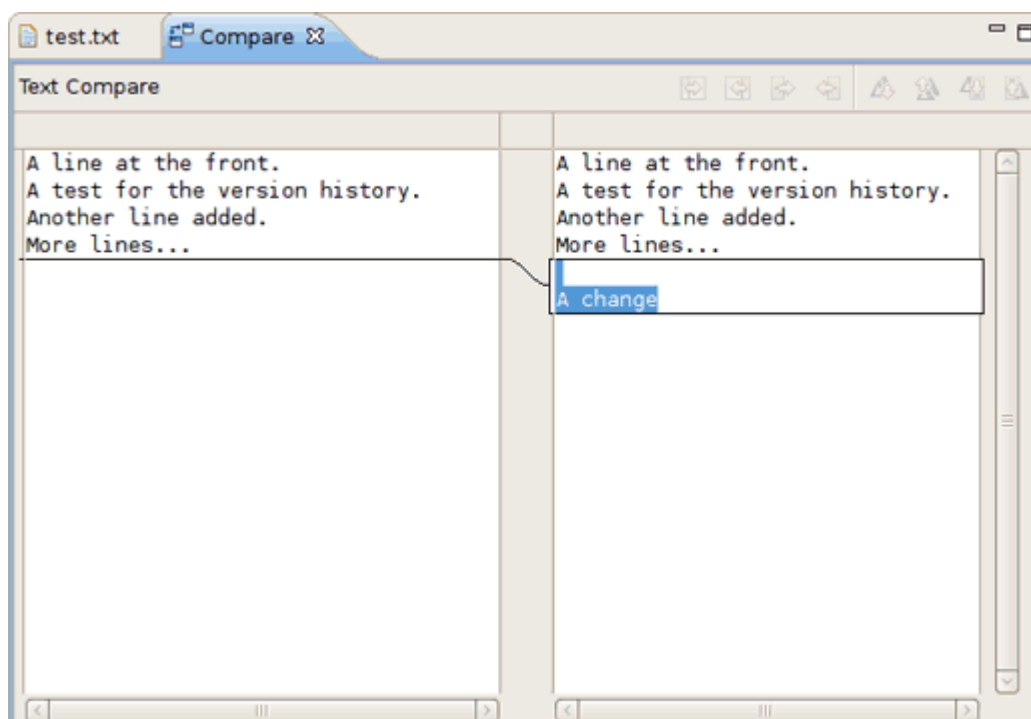


Figure 2.14. Eclipse Compare Editor

This editor uses Eclipse-standard comparison techniques to show the differences in the two versions. In cases where there are no differences, the editor will not open, rather a dialog saying that there are no differences will appear.

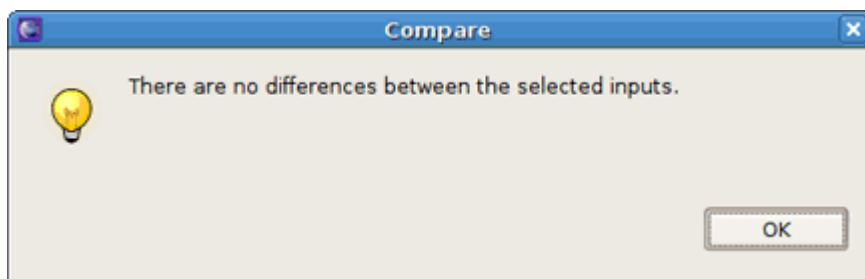


Figure 2.15. Alert Dialog

2.5.6. Switch to Version Action

The *Switch to Version* action is enabled for one Guvnor repository associated file.

First the *Switch to Version* action prompts for selection of version:

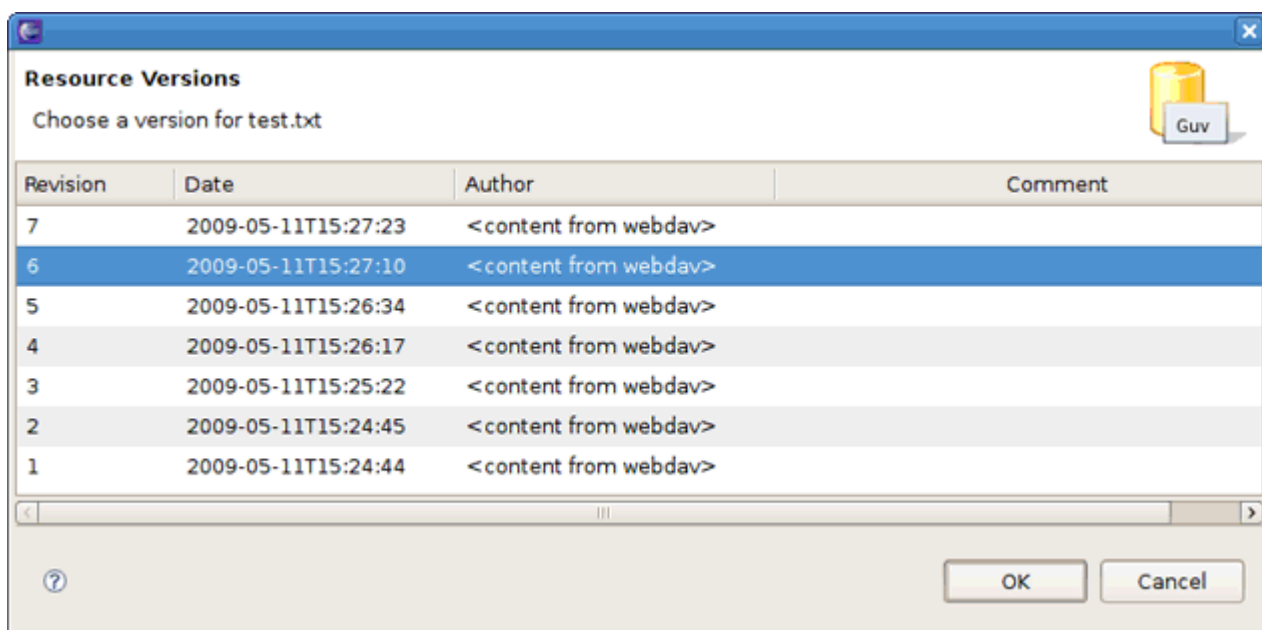


Figure 2.16. Select Version Window

Once the version is selected, the *Switch to Version* action replaces the local file contents with those from the revision selected.

2.5.7. Delete Action

The *Delete* action is enabled for one or more Guvnor repository associated files. After confirmation via a dialog, the *Delete* action removes the files in the Guvnor repository and deletes local metadata for the Guvnor repository association.

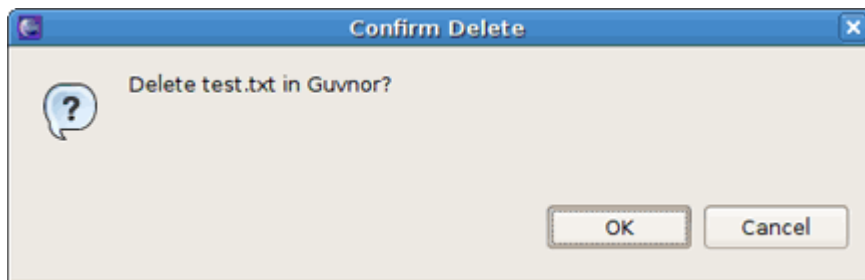


Figure 2.17. Comfirm Delete Dialog

2.5.8. Disconnect Action

The *Disconnect* action is enabled for one or more Guvnor repository associated files, and removes local metadata for the Guvnor repository association.

2.6. Guvnor Resource History View

The *Guvnor Resource History view* should details about revision history for selected files, both local and those in Guvnor repositories. The initial state of this view is shown on the figure below.

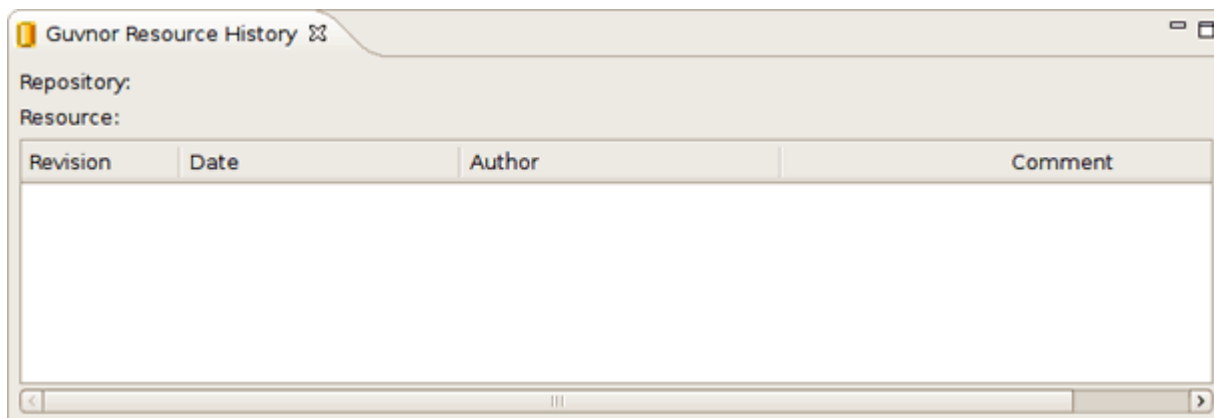
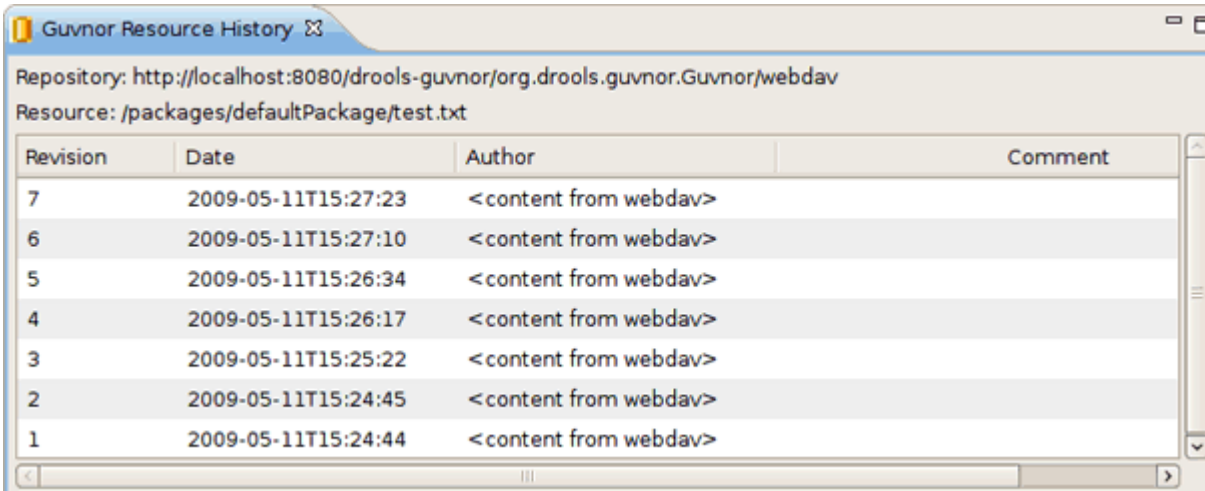


Figure 2.18. Initial State of the Guvnor Resource History View

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*. Once this action is performed, the *Guvnor Resource History view* updates to show the revision history:

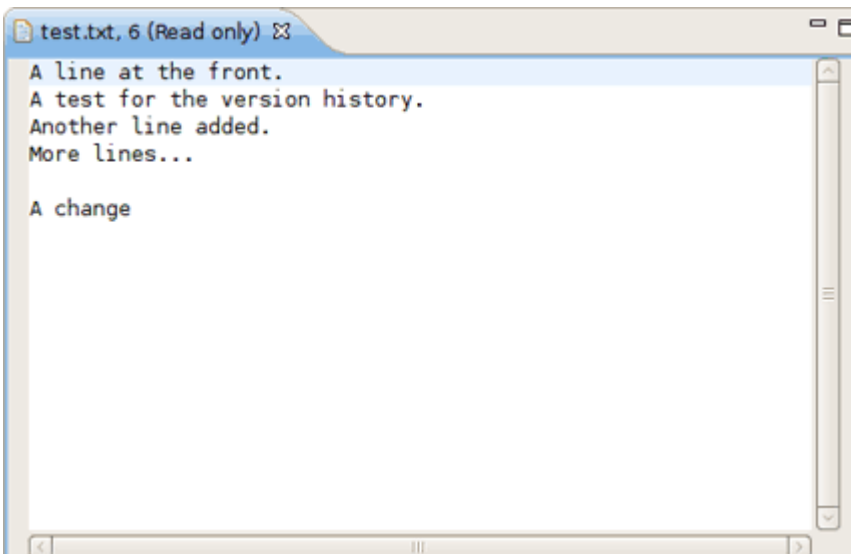


The screenshot shows a window titled "Guvnor Resource History". It displays the repository URL as `http://localhost:8080/drools-guvnor/org.drools.guvnor.Guvnor/webdav` and the resource path as `/packages/defaultPackage/test.txt`. Below this is a table with four columns: Revision, Date, Author, and Comment. The table lists seven revisions, all with the comment "<content from webdav>".

Revision	Date	Author	Comment
7	2009-05-11T15:27:23	<content from webdav>	<content from webdav>
6	2009-05-11T15:27:10	<content from webdav>	<content from webdav>
5	2009-05-11T15:26:34	<content from webdav>	<content from webdav>
4	2009-05-11T15:26:17	<content from webdav>	<content from webdav>
3	2009-05-11T15:25:22	<content from webdav>	<content from webdav>
2	2009-05-11T15:24:45	<content from webdav>	<content from webdav>
1	2009-05-11T15:24:44	<content from webdav>	<content from webdav>

Figure 2.19. IGuvnor Resource History View

Here we see that the file `test.txt` has three revisions. Double clicking on a revision row (or the context menu `Open (Read only)`) opens an Eclipse read-only editor with the revision contents.



The screenshot shows an Eclipse editor window titled "test.txt, 6 (Read only)". The editor contains the following text:

```
A line at the front.  
A test for the version history.  
Another line added.  
More lines...  
  
A change
```

Figure 2.20. Eclipse Read-only Editor with the Revision Contents



Note:

You can also use the `Save As...` option when a file is open in a read-only editor to save a local writable copy of the contents. Doing so, however, will not associate the file created with its Guvnor source.

2.7. Importing Guvnor Repository Resources

In addition to the single file drag-and-drop from the [Guvnor Repositories view](#), the EGT also includes a wizard for copying one or more files from a Guvnor repository to the local workspace (and setting the association with the Guvnor repository). This wizard is available from the [File > Import > Guvnor > Resource from Guvnor](#) and the [File > New > Other > Guvnor > Resource from Guvnor](#) menu items.



Note:

Note that the wizard is identical but appears in both locations to accommodate users who tend to view this functionality as being in either category.

The first page of the wizard asks for the selection of the source Guvnor repository and gives the choice to create a new Guvnor repository connection (in which case the second page is the same as the [Guvnor Connection wizard](#)).

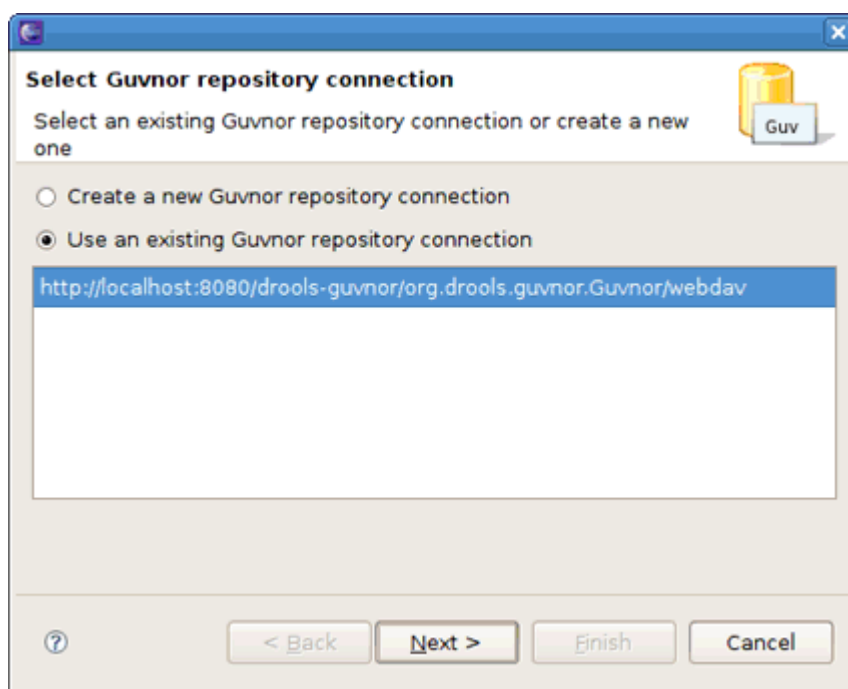


Figure 2.21. Resource from Guvnor Wizard

Once the source Guvnor repository is chosen, the wizard prompts for resource selection:

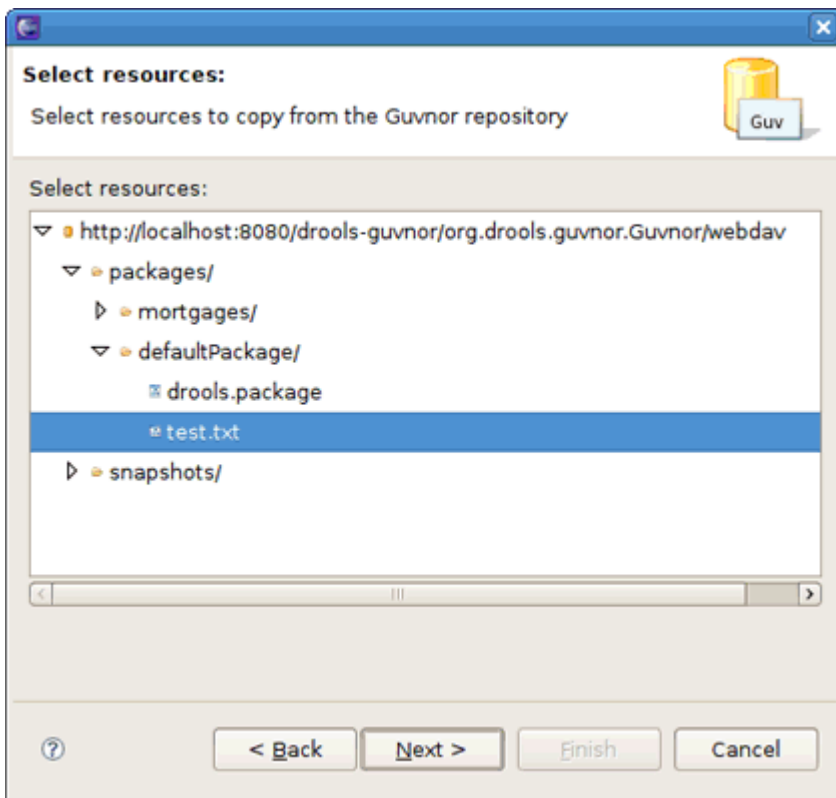


Figure 2.22. Resource Selection

Finally, the target location in the local workspace should be chosen:

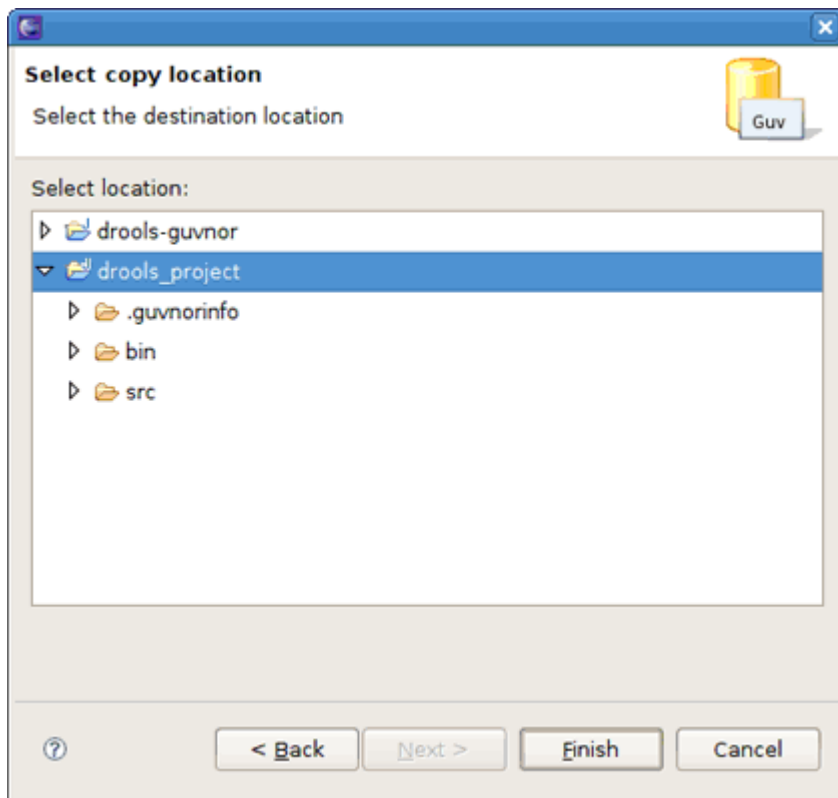


Figure 2.23. Choosing the Target Location

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, the wizard uses the Eclipse standard "prompt for rename" dialog:

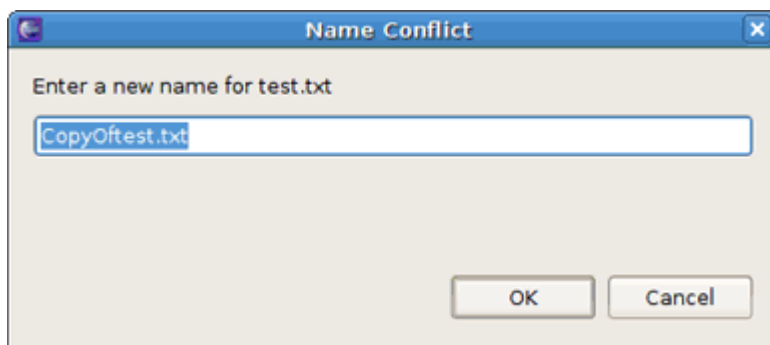


Figure 2.24. Prompt for Rename Dialog

Guvnor Preferences

The EGT provides a preference page in the *Guvnor* category:

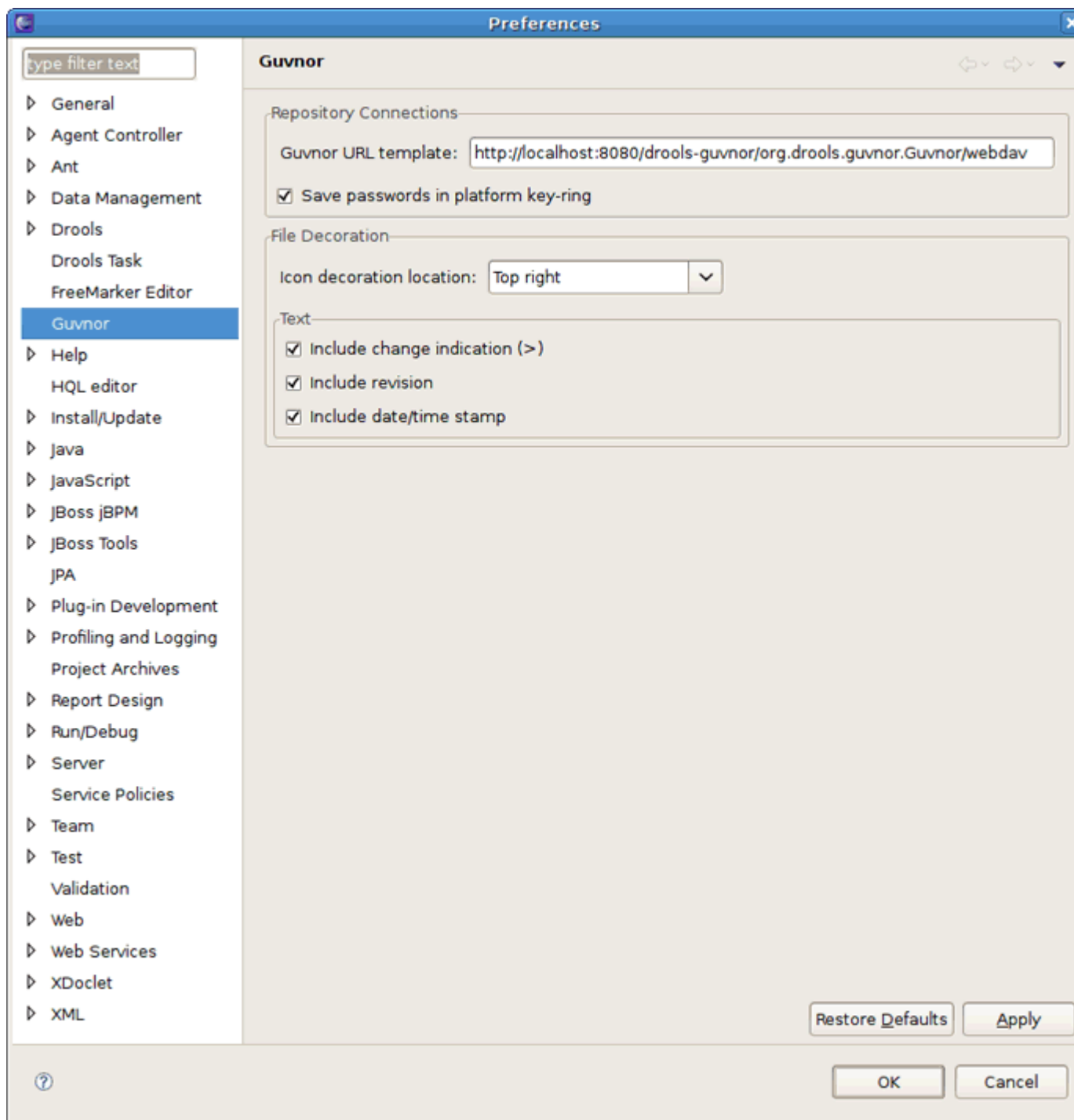


Figure 3.1. Resource Selection

The preferences cover two categories:

- Guvnor repository connections
- Local Guvnor repository resource decorations.

3.1. Guvnor Repository Connection Preferences

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

- The first is a default Guvnor repository URL template, which can make it easier to create multiple similar connections by simply changing part of the field, such as the host name.
- The second is whether saving of authentication information in the Eclipse platform key-ring should be enabled by default.

As with the Guvnor repository URL template, actually whether to save a specific instance of authentication information in the Eclipse platform key-ring can be determined when actually creating the connection. That is, both of these preferences are simply convenience values set to reasonable defaults.

3.2. Local Guvnor Repository Resource Decoration Preferences

The second category of preferences provided by the [EGT](#) deals with 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 just how the [EGT](#) decorate its local resources to avoid messy 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 that 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.

Conclusion

As stated at the beginning of this document, the key goal of the [EGT](#) is to provide a way of interacting with Guvnor repository resources in a local Eclipse workspace. While clearly there is a lot more that could be done, and no doubt there will be aspects of the current tooling that require revision going forward, we feel that the current state of the [EGT](#) is sufficient for the major use cases.

If you have some questions, comments or suggestions on the topic, please feel free to ask in the [Jboss Tools Forum](http://www.jboss.org/index.html?module=bb&op=viewforum&f=201) [<http://www.jboss.org/index.html?module=bb&op=viewforum&f=201>].