| Task Repository Connectors | | |
| --- | --- | --- |
|  |  |  |
| Preferences |  | Updating This Document |

# Task Repository Connectors

Mylyn allows you to collaborate on tasks via a shared task repository, also known as bug tracking systems. In order to collaborate, you need to have a **Connector** to your particular repository.

See [Mylyn Extensions](http://wiki.eclipse.org/Mylyn/Extensions) for a list of available connectors.

## Bugzilla Connector

See [Bugzilla Connector Troubleshooting](http://docs.google.com/Mylyn/FAQ/Bugzilla-Connector.html#Bugzilla_Connector).

## Trac Connector

See [Trac Connector Troubleshooting](http://docs.google.com/Mylyn/FAQ/Trac-Connector.html#Trac_Connector).

## Generic Web Templates Connector

The generic web repository connector is NOT part of the default Mylyn install. You can install it from the incubator update site. See [Mylyn download page](http://www.eclipse.org/mylyn/downloads/) for more details.

The web connector allow to retrieve tasks from repositories that don't have rich connectors, but can show list of tasks on the web UI. Out of the box connector provides configuration templates for the following issue tracking systems:

* Google Code Hosting (code.google.com)
* IssueZilla (java.net, dev2dev, tigris.org)
* GForge (objectweb.org)
* SourceForge (sf.net), see [Using Sourceforge with Mylyn](http://wiki.eclipse.org/Using_Sourceforge_with_Mylyn)
* Mantis (www.futureware.biz/mantis)
* ChangeLogic (changelogic.araneaframework.org)
* OTRS (otrs.org)
* phpBB
* vBulletin

Lists of issues can be extracted from existing web pages using simple parsing configuration. Configuration can be also parametrized to make it easier to customize it for a specific project.

The parameters used for configuring project properties are typically substituted into the URLs used to access the repository. Substitution and matching rules can be edited under the *Advanced Configuration* section on both the *Repository Settings* page and the *Edit Query* page.

See [FAQ](http://docs.google.com/Mylyn/FAQ/Web-Templates-Connector.html#Web_Templates_Connector) for the troubleshooting tips.

**For example**, consider the configuration steps for GlassFish project at java.net:

**1.** Create new Generic web-based repository (in the Task Repository view). GlassFish is using IssueZilla and has a preconfigured template that can be selected by server url [*https://glassfish.dev.java.net/issues*](https://glassfish.dev.java.net/issues). You can also specify all fields manually in the *Advanced Configuration* section. For GlassFish the following settings are required:

* Task URL: ${serverUrl}/show\_bug.cgi?id=
* New Task URL: ${serverUrl}/enter\_bug.cgi?issue\_type=DEFECT
* Query URL: ${serverUrl}/buglist.cgi?component=glassfish&issue\_status=NEW&issue\_status=STARTED&issue\_status=REOPENED&order=Issue+Number
* Query Pattern: <a href="show\_bug.cgi\?id\=(.+?)">.+?<span class="summary">(.+?)</span>

**Note:** *Query Pattern* field should be a regexp with 1st matching group on *Issue ID* and 2nd on *Issue Description*. Alternatively, you could use named matching groups: ({Id}.+?), ({Description}.+?), ({Status}.+?), ({Owner}.+?) and ({Type}.+?), then they can appear in query regexp in an arbitrary order. The second option requires build 2.0.0v20070717 or later. **Note:** the above fields are using parameter substitution ${..}. Variables serverUrl, userId and password are substituted from the values of corresponding fields of the repository preference page. In addition you can specify any arbitrary parameters and their values that will be also substituted into the template fields. **Note:** the SourceForge template included with connector assume that *single repository is used for all projects*. User should create multiple queries, and set project parameters at the query level. Because web connector don't support actions like "open repository task" there is really no need to create separate repositories per project and if you think about it that is how it work for connectors for Bugzilla and JIRA. However, it is still possible to setup separate repository per project using repository url like http://sourceforge.net/tracker/?group\_id=172199 and accordingly updating derived urls is the advanced repository settings.

**For the web repository that require user to login, use advanced configuration in following way.**  *This configuration is for GForge, you might need to change it for other repositories*:

* Login Request URL - an address that form is using to submit login request: ${serverUrl}/account/login.php?return\_to=&form\_loginname=${userId}&form\_pw=${password}&login=Login **(POST)**
* Login Form URL - an address where login form is located *(only needed if server need a login token in the parameters of the Login Request URL)*: ${serverUrl}/account/login.php
* Login Token Pattern - pattern to extract value of the loginToken parameter from the form page *(only needed if server need a login token in the parameters of the Login Request URL and Login Form URL is specified)*: session\_ser=(.+?)

**2.** Create a new query for the GlassFish task repository created above (either from popup context menu in the Task List view or using a "New..." wizard from File -> New... -> Other... menu).

* *Query URL* and *Query Pattern* in the *Repository Preferences* are used as default query parameters and can be overwritten in *Advanced Configuration* section in *Query Preferences*. Custom parameter values can also be overridden here as well as new parameters for substitution into the specific query.
* In the *Advanced Configuration* section of the "New Query" dialog, there is a "Preview" button. You can use it to test your query pattern.



## XPlanner Connector

The XPlanner connector is currently in the Experimental state. It supports XPlanner version .7.

**Query**

XPlanner Connector supports creating queries from your XPlanner repository. You can select to create a query for your own in progress tasks (the default), or select projects or iterations or user stories from a list. You can also select ones assigned to you, or all. Currently you can choose to group the results by tasks or user stories. If you group the results by tasks, this will result in a single query with the applicable tasks underneath it. If you choose to group the results by user stories, this will result in multiple queries added to the task list, each one corresponding to a user story, with the applicable tasks underneath it:



**Task Editor**

A rich editor for editing XPlanner tasks is available. It currently doesn't support all of the attributes you can change in XPlanner itself, but allows you to modify the basic ones necessary for XP development -- e.g. summary, description, estimated time. The attributes section of the editor displays an XPlanner-like view of the meaningful attributes, some of which are not editable.



**User Story Editor**

If you open an XPlanner element by id, using the "Open Repository Task by Key/ID" dialog, and that id is a user story id, you will see the read-only user story editor. This editor is just useful for seeing the major details of a user story -- no items in it can be changed, since there is no really good mapping between a user story and a Mylyn task.



[Category:Draft Documentation](http://wiki.eclipse.org/Category:Draft_Documentation)

|  |  |  |
| --- | --- | --- |
| Preferences |  | Updating This Document |