

GOVERNANCE MODEL

Ensuring that the Taverna project is developed by the community in a structured and transparent way requires a certain degree of governance. The following rules and processes have been adopted to strike a balance between encouraging anyone to make a contribution at any time and maintaining a high level of quality in the software.

1.0 ROLES AND RESPONSIBILITIES

There are quite a few ways to participate in the Taverna project and community, and not all of them involve contributing source code to the project. Simply participating on mailing lists, or filing bug reports or enhancement requests is an incredibly valuable form of participation.

If one were to break down the forms of participation in the Taverna project into a set of roles, the result would look something like this: Users, Contributors, Committers, Maintainers, and Project Lead.

1.1 Users

Users are the people who download and use the Taverna workflow engine. Users are using the software, reporting bugs, making feature requests and suggestions. This is by far the most important category of people. Without users, there is no reason for the project.

How to become one: Download the Taverna software and use it to run workflows.

1.2 Contributors

Contributors are individuals who contribute to the Taverna software source code but do not have write access to the source tree. Contributions can be in the form of source code patches, new code, or bug reports, but could also include web site content like articles, FAQs, or screenshots.

How to become one: Contribute in any of the ways described above: either code, examples, web site updates, tests, bugs, and patches

Integration of a Contributors' submissions is at the discretion of the project maintainer, but this is an iterative, communicative process. Note that for code to be integrated, a completed Contribution Licence Agreement (CLA) is required from each contributor or organisation. See the CLA information below.

1.3 Committers

Committers have access to the source tree, either for the individual modules they are working on, or in some cases global write permissions everywhere in the source tree.

A committer must complete and send in a CLA form to commit code.

Rules for how to commit, once you have commit access, will vary by module. Be sure to ask before you start making changes!

How to become one: Submit some patches via email, and ask the maintainer of the code you have patched for commit access. The maintainer will seek consensus before granting Committer access, but their decisions are final.

1.4 Maintainers

Software developers at the University of Manchester are responsible for merging contributors' patches, bug fixes, and new code from the development branch of the source tree into the stable branch. Maintainers are responsible for making sure that these contributions do not break the build.

The Maintainer is also responsible for checking that everyone who contributes code has submitted a CLA. See the CLA information below.

A Maintainer is responsible for their module (by having access to the source tree), and for granting check-in privileges to contributors. They also act as the "gatekeeper" of the module, helping to ensure quality across the build.

1.5 Project Lead

The Project Lead is responsible for the overall direction and vision. In addition to making sure that releases are planned correctly and completed on schedule the Project Lead decides who receives commit access to the source code repository. This ensures that the quality of the software and the associated documentation remains high and the conceptual integrity of the project is maintained.

The Project Lead for Taverna is a group of developers based at the University of Manchester which is led by Professor Carole Goble, project investigator (PI) of the myGrid project. The Taverna Workflow Workbench was originally developed as part of the myGrid project, a collaborative project between five UK universities and the EBI.

2.0 CONTRIBUTING PATCHES

If you are not an official member of a project team or do not have commit access to the source code repository then you can still make a contribution by submitting a patch. A patch is simply a file that contains the differences between the files you are modifying and the new versions you have created.

A patch can be submitted by emailing the file to the taverna-hackers mailing list, or a message that you wish to submit a patch and one of our maintainers will contact you. If the patch relates to a particular bug or requested enhancement then please provide this information. If the patch file is very large, then a link to a download-able copy of the patch file should be provided. Please ask on the taverna-hackers mailing list if you are unsure.

2.1 Contributor's Agreement

In order to accept a patch we require that you first complete either an Individual or Corporate Contributor's Agreement acknowledging certain terms and conditions for its use. Once this agreement has been completed and a patch is accepted then the differences will be applied to the original source code or documentation by the project team and committed to the source code repository on your behalf.

Please print this form out, fill in all the necessary detail, and return it. You may also want to read the FAQ.

Via fax:

Number is +44 (0) 161 275 6204

Via mail to:

Alan Williams or Stuart Owen
School of Computer Science,
The University of Manchester,
Oxford Road,
Manchester,
M13 9PL
UK

2.2 Mailing Lists

There are two mailing lists available for the Taverna project, these are hosted by sourceforge and the links below allow you to subscribe and view the mail list archives. Please be aware that both developer and user lists are public. We suggest that users should join the user list at least - this allows us to notify you of any critical issues or upcoming changes and allows you to have a say in the future development of Taverna. Anyone seriously contemplating using the Taverna APIs in their own code would be well advised to join both lists.

Please do not email the individual developers - you are much less likely to get a response, we're all on the mailing lists!

e-mail contact

Kilburn Building, Department of Computer Science
University of Manchester
Oxford Road, MANCHESTER
M13 9PL
United Kingdom

Telephone: +44 (0)161 275 0661

Fax: +44 (0)161 275 6211

E-Mail: mygrid@cs.man.ac.uk

Taverna Users List
myGrid project wide users mailing list

Archive: <http://taverna-users.markmail.org/>
Join: <https://lists.sourceforge.net/lists/listinfo/taverna-users>

Taverna Hackers List
Intended for those actively developing on Taverna and myGrid
Archive: <http://taverna-hackers.markmail.org/>
Join: <https://lists.sourceforge.net/lists/listinfo/taverna-hackers>

3.0 FEATURE REQUEST AND BUG TRACKING

Submitting bug reports and suggesting new features are important tasks to help improve the quality and functionality of the software. By following the advice given on this page you can help to ensure that all relevant information is captured and entered into the issue tracking system in the correct way. This helps the project team and other contributors to take action in the shortest possible time.

You can browse the current issues and features that have been reported in our issue tracker JIRA. The software running the issue tracker has been provided to us - as an open source project - free of charge by Atlassian.

We recommend that you report possible bugs and feature requests to one of our mailing lists, and our dedicated support hat person will respond and enter the issue into JIRA if required.

Although you might register as a user of our issue tracker, you will by default not have permission to submit new bug reports, the main reason for this is that we received too many spam "reports" and comments, but we might review this policy at a later stage.

- Taverna JIRA bug and feature tracking:
<http://www.mygrid.org.uk/dev/issues/>

myGrid uses JIRA to manage the bugs that we confirmed as being official bugs, as well as other issues and feature requests that we are working on.

Although there is no reason why we can not open a JIRA account for genuine bug reporters, we currently feel that it is better to submit bugs and comments to one of our mailing lists, taverna-users or taverna-hackers (depending on the level of technical complexity).

This way other users can track the activity, and our dedicated maintener person of the week will pick up on the case and see if it should is appropriate to be entered into JIRA or not, and also ensure that any additional details are provided.

3.1 Contribute Code

If you want to help out and contribute some code then the best place to start is by looking at the issue tracking system to find a task that is unassigned or needs

completing. You can then send an email to the Project Lead or post a message on the Taverna Hackers List with a reference to the task, stating that you would like to own it. If there is an area where you want to make a contribution but there are no relevant tasks then use the Taverna Hackers List to post a message asking for one to be created.

Before we can accept any contributions of code you need to have signed a Contributor's Licence Agreement (CLA). This sets out the terms and conditions under which the software you wish to contribute is to be used within the Taverna project. Two agreements are available to choose from depending on whether you represent an individual or a corporate contributor.

3.2 Submitting New Bug Reports

Before submitting a new bug, please search our issue tracker to check the bug hasn't already been reported. If it has, then you can add a comment or vote – though you will need to be registered with JIRA to do so.

When you do submit a bug report, please include the following information:

- The operating system you are using, e.g. RHEL 5, Windows XP, Solaris 10
- The JDK or JRE you are using, e.g. Sun Microsystems JDK 1.5.0_09
- As much of the stack trace (if there is one) as necessary to show the cause of the problem.
- A list of steps to reproduce the bug.
- Name of the software, component or plugin that the bug report refers to
- The version of the software that the bug report refers to
- Details of the plugins that have been installed

The objective is to give enough information for someone with no prior knowledge of the bug to recreate the problem for themselves. Too little information may prevent them understanding when the bug occurs and too much information can create unnecessary confusion.

3.3 Submitting New Feature Requests

When submitting a request for a new feature make sure to keep it as brief and to the point as possible. Each new feature requires its own issue in the issue tracking system so that they can be addressed separately by developers. It is up to the Project Lead to determine when new features are added, according to the development plans they have made. You may point out reasons why features should be added sooner rather than later, such as the ability to compete with an alternative product, but unless you are willing to do the work yourself you may have to wait for development to take place.

4.0 DOCUMENTATION

Taverna documentation is provided at:

<http://taverna.sourceforge.net/doc>

Since documentation needs to be developed in a structured way, to coincide with each product release, it is stored in a version control repository. This means that to make a contribution to a document you need to submit a patch.