Tools

Tool How To's

Have a simulation/modeling tool that you want to deploy for others to use? This hub makes it easy. You don't have to build the tool for multiple platforms. You don't have to distribute binaries or tar balls. Just get your tool installed and running within our environment, and others can access your tool via the Web.

Existing Graphical User Interface that runs under Linux/X11

If you already have a tool with a graphical user interface built with Java, Qt, MATLAB, or anything else that runs under Linux/X11, you can probably deploy it as-is within this hub in a matter of hours. There are a few caveats:

- **If your tool relies on a graphics card for acceleration, the graphics performance will be slow.** Tools running within the virtual machines of this hub use software rendering for all graphics. It will probably work, but graphics may be clunky.
- **If your tool needs to pull data from external sites (other web sites and databases), you will have to let the hub team know.** All such connections are blocked by default, in order to safeguard against malicious tools. Once your tool has been approved and white-listed, it will have controlled access to external sites.
- **Your tool will be running in a remote server environment with its own file system.** The usual *File > Open...* operation will look for files on the server--not on the user's desktop. Users can upload files via sftp, WebDAV, and the hub's own importfile command. You might consider modifying your tool so that *File > Open...* launches importfile, so your users can upload files from their desktop on demand.

Command-line interface that runs under Linux

If you don't have a graphical user interface for your tool, you can create one in short order by using our Rappture Toolkit. Rappture integrates with a wide variety of programming languages, including C/C++, Fortran, Java, MATLAB, R, Python, Perl, Ruby, and Tcl/Tk. Check out these videos to get started:

- Introducing the Rappture Toolkit

- What's Under the Hood?
- More Rappture Objects
- Advanced Visualization

Can your tool take advantage of cluster resources or parallel execution? We have additional computing resources within the hub that you can tap into via the submit command. Write a wrapper program in your favorite language, and have it gather the input values, prepare input files, and submit your runs to our remote computing resources.

Tools for Microsoft Windows or Macintosh Systems

Sorry, your options here are limited. All tools deployed on this hub must run in a Linux/X11 environment. In some cases, tools written for Microsoft Windows will run under Linux via Wine emulation. If that works in your case, then you can deploy your tool after setting up the appropriate Wine configuration files. If not, then we won't be able to host your tool.

Follow These Steps

- ## Create a project area for your tool

  Watch this video to get a sense of the tool upload process, then register your tool project . This lets us know that you're starting a new tool. We'll grant you access to our remote development environment, and we'll create a project area with a Subversion source code repository where you can store your code.

- ## Upload your code and build/test in the workspace

  A workspace is a remote Linux desktop--a window into our remote development environment that you can access from any Web browser. Watch this video to learn how to use workspaces, then go to the workspace page and click *Launch Tool*. You can use sftp, WebDAV, or importfile to upload your code into the workspace. After that, you can build and test your code as you would in any other Linux environment. Once your tool runs properly in the workspace, it is ready to deploy.

- ## Check in your code and tell us you're ready for deployment

  The project area that you created in Step 1 comes with a Subversion source code repository. After having tested your code in the workspace, commit it to the Subversion repository. If you've never used Subversion before, you should watch this tutorial. Once your code is checked in, you can click a link in your project that says *My code is committed*, as described in this video. We'll install your code and get it ready for final testing. Click another link to say that you approve the tool, we'll make it available for others to use.

-

-

Keep an eye on your tool

Each published tool has its own page listing the name of the tool, the authors, and a brief description. This page also shows the number of people who have accessed the tools, along with any questions they have posted, and ideas for improvement. Keep an eye on this page and watch your tool's popularity grow. Help answer the questions, use the feedback to make improvements, and publish improved versions of your tool. The more you support your tool, the more your user base will grow.