Ghub Tool Directory Structure

When developing tools, use the recommended Hub tool directory structures and save yourself some grief.

Hub Application Space Directory Structure

This is the actual directory structure of your tool's installed home on the Hub. The developer of the tool has read access to all directories except src/.

As the developer of the tool, you can run your tool from the command line (using e.g. middleware/invoke) in these directories.

- If the tool is Installed, it can be run from the dev/ subdirectory;
- if Published, it can be run from the current/subdirectory.

```
/apps/toolname/
 dev/  - symlink to most recent Installed release
 r1/  - release 1
              ...
 rn/  - release n
 current/ - symlink to most recent Approved release

  bin/  - contains binaries (executables)
  data/  - any needed data, optional
  doc/  - documentation, optional
  examples/ - examples of tool use, optional
  middleware/ - contains invoke script
  rappture/ - contains tool.xml (must point to bin directory)
  src/  - contains source code and Makefile
```

User's Tool Development Directory Structure

Each user has a workspace to use for tool development. Setting up your workspace with a test/ area and a repo/ for each Tool you develop minimizes headaches.

Once a tool has been Registered, users can access the Workspace tool to run an xterm, create the directory structure, test and troubleshoot tools, and check out the Subversion repository contents for *toolname*.

Recommended user workspace structure:

```
 ~/apps/toolname/
  test/  - test subdirectory, for testing tool's basic functionality,
without GUI
  repo/  - project directory structure checked out from Subversion rep
ository

   bin/  - contains binaries (executables)
   data/  - any needed data, optional
   doc/  - documentation, optional
   examples/ - examples of tool use, optional
   middleware/ - contains invoke script
   rappture/ - contains tool.xml (must point to bin directory)
   src/  - contains source code and Makefile
```

From the hub team:

“The initial project directory structure is set by contribtool. Upper level directories include src, bin, rappture, data, examples, doc, and middleware. To operate smoothly within the HUB infrastructure it is expected that these directories be left intact although they may remain empty.”

from https://nanohub.org/answers/question/374

User Session Information

When testing and troubleshooting your Tools on the Hub, you can make use of your own user session information. Refer to the following subdirectories of your home directory for session and result data:

```
 ~/data/
  sessions/ - uniquely numbered sessions opened by the user
  results/ - results generated in sessions, identified by session numb
er
```