GumpCode

'Gump Code'

The main gump packages are documented here:

- PyDoc Root
- Gump PyDoc

Main Packages:

See GumpInternals for how a run (the tree of context) is formed/loaded (objects from XML) and worked upon.

- Core Classes (e.g. GumpRunOptions, GumpSet, GumpRun)
- Object Model (e.g. Project/Module, etc.)
- Runner (perform a run)
- Updater (CVS/SVN/etc.)
- Builder (Ant/Maven/etc.)

1) Create GumpSet

The input parameters (e.g. all or * or avalon-*) are matched against projects in the loaded workspace to determine a list of projects. The dependencies for those projects are sorted to create an order project sequence. The modules for that complete sequence is stored. We now have a full GumpSet with all the details we need to perform a run. This (along with Run Options), and the original workspace tree (of projects/modules) are stored on a GumpRun.

2) Run

The Runner loops throug the build sequence and (when needed) does an update on a module before doing a build of the project. When done (with each) a Run Event is dispatched to all listening worker/actors. The workers/actors do their thing (e.g. update stats, generate documentation, etc.)

Note: Currently context (such as output from exec commands) are stored on the model, not on a wrapper, hence the model/tree is poluted by the run & not re-usable. [Work todo...]

Workers (Actors):

The workers work on the content tree (the tree of results) based off events that come from the runner, e.g. module complete, project completed, etc. These actors process the state of the entity and do their thing (including updating information on that entity in the tree, e.g. timestamps).

- Documenting (Text/XDOCS/XHTML)
- Notification (E-mails)
- Syndication (RSS
- Statistics (to DBM file)
- Results (e.g. XML on/for other servers)
- Repository (Artifacts Repo)
- Timing (Time Keeper, how long did X take)

Random Important:

- Loading from XML to model
- Utility code (e.g. file sync, DOM, exec, etc.)

Random Minor:

- Shared code (e.g. model comparators by XYZ)
- Integration code (e.g. w/ CVS/others)
- Stats & XRef 'gurus' – know it alls
- SVG utilities (Scalable Vector Graphics)
- Unit Tests
- Multithreading