

Minutes from 13/08/2007 (Tuesday)

Attendees: David, Ken, Jon, Hill

Next Meeting: 20/08/2008 (Tuesday) @ 1110 hours

- The file monitor has been change to implement the API develop by Ken/Jill
- Aim to store file into database
- David asked? How will the groups be able to use it?
- Ken: see if you can reference methods from API
- David: how to use API to address some of the tickets? where do they start
- Hill: take the basic function from the API, otherwise, extend the APi
- David: are the plugins going to have similar structure than current file monitor
- Ken: yes
- David: so is a new plugging using the methos from API? but how do you know where to use them? how do you go to a file monitor??
- David: the methods seems not to be to generic
- David: make 2 plugins that have roughly a different behaviour
- David: is this going to work in practise?
- David: how did u design the API?
- David: what/when the database or the models need to be changes to accommodate a different application? Advance plugins may require more (you have a function to get info from files, but later, you what to get more information or different information from the files... how to we do that with this??)
- Ken: currently cannot.
- David: Let say you can get the tables and models, is API good enough to cover extra tables. How do you write into different tables depending on what your application is? If you have a more advance plugging you may write more metadata that you don't do with basic plugins. It has to be obvious to the developer what/where to add the code needed.
- David: where is the actual file monitor?
- Ken: used by the earth daemon
- David: so your API define default behaviour in somewhere. But when writing plugins you may want to override this default behaviour (beside extending it)
- David: they seem to be fairly high level function

- David: if I want to do an extension, where do I do the extension??
- Hill: if you dint have something just extend API
- David: shouldn't extend the API that often/much
- David: writing a new plugging should just extend/call what is in the API
- Ken: we can look through ticket and extend the API to what it may be needed in the future
- David: API should have lower level function to either use or overwrite. That's why we need to look at the tickets
- David: We need to get to the point to be able to demonstrate some of the tickets as plugins.
- David: **we need to look at (future task) how to take existing tickets and how the system works (to tickets, one with extension function)**
- David: **what we need to have is a clear structure of how other groups can use this design to extend the API and provide support for specifics plugins**
- David: example, 2 plugins using different naming convection....
- David: go trough old tickets and see where those functions should go
- David: **end of week have a clear direction , putting a design together**
- David: is there a concepts of plugins to plugins???
- Ken: no at the moment. Haven't been consider
- David: is the file monitor a single API ? should it be broken down
- David: experiment
- David: next iteration: is this system going to work?
- David: **so far if we want to rewrite a function from the API we will need to rewrite (duplicate) all the code, but maybe there are pieces that does not to be changed. By findin the default behaviour that will give us the hooks.**
- David: we may need to get the hole group to work with.
- David: the basic systems seems to be OK but need decomposition.
- David: need to get all ready in next 2 days to be able to bring the other groups on board