Basic dependencies

The next step is to understand basic dependencies.

In this example, we use the job's dependency field to set up a job-based dependency. Job based dependencies are those that will wait for the entire dependent job to finish before the current job starts. There are also subjob (instance) and agenda (frame) based dependencies.

jobSubmit_dependency-example.py

```python
#!/usr/bin/python

import os, sys

# The next few lines attempt to import the Qube API. If the path to the qb module
# is not in $PATH or $PYTHONPATH, we will attempt to find it by looking in known
# locations
try:
    import qb
except ImportError:
    if os.environ.get("QBDIR"):
        qbdir_api = os.path.join(os.environ.get("QBDIR"),"api","python")
    for api_path in (qbdir_api,
                    "/Applications/pfx/qube/api/python/",
                    "/usr/local/pfx/qube/api/python/",
                    "C:\\Program Files\\pfx\\qube\\api\\python",
                    "C:\\Program Files (x86)\\pfx\\qube\\api\\python"):
        if api_path not in sys.path and os.path.exists(api_path):
            sys.path.insert(0,api_path)
            try:
                import qb
            except:
                continue
            break
    # this should throw an exception if we've exhuasted all other possibilities
    import qb

def main():
    # We're going to submit two jobs in the example - the second will wait
    # for the first

    # Set up the basic job properties as we've done before.  We're going to do
    # a "sleep" job, so we'll limit this test to osx or linux
    job = {}
    job['name'] = 'python parent job'
    job['cpus'] = 2
    job['prototype'] = 'cmdrange'
    job['requirements'] = 'host.os=linux or host.os=osx'

    # For the package, we'll just do a simple sleep
    package = {}
    package['cmdline'] = 'sleep 5'
    job['package'] = package

    # Create a typical agenda, as we've done in previous examples.
    agendaRange = '0-60x10'
    agenda = qb.genframes(agendaRange)
    job['agenda'] = agenda
```

```python
    # Now we submit a single job (as a single element list).
    # This will be the parent job
    listOfJobsToSubmit = []
    listOfJobsToSubmit.append(job)
    listOfSubmittedJobs = qb.submit(listOfJobsToSubmit)
    parentJobID = listOfSubmittedJobs[0]['id']
    print 'parent: %d' % parentJobID


    #==================================================================#

    # Now we'll create a second job that will be dependent on the first
    # the basic setup is the same as before
    job = {}
    job['name'] = 'python child job'
    job['cpus'] = 1
    job['prototype'] = 'cmdrange'
    job['requirements'] = 'host.os=linux or host.os=osx'

    # To create the dependency on the parent, we simply create a
    # 'dependency' attribute.  See link in this doc's text for more information
    job['dependency'] = 'link-done-job-%d' % parentJobID

    # Continuing on, we create a typical "sleep" job for testing
    package = {}
    package['cmdline'] = 'sleep 20'
    job['package'] = package

    # As before, we submit the single job as a single element list
    listOfJobsToSubmit = []
    listOfJobsToSubmit.append(job)
    listOfSubmittedJobs = qb.submit(listOfJobsToSubmit)

    for job in listOfSubmittedJobs:
        print 'child: %d' % job['id']

if __name__ == "__main__":
    main()
```

```
        sys.exit(0)
```

Running this script will create 2 jobs. The first will run immediately, the second will be in a blocked state until the first job completes, then it will automatically start running.

Continue to Advanced Dependencies