

# Basic python job submission II. Frames

Next, we will submit a job with frames, or, in Qube!-speak, "agenda items".

```
#!/usr/bin/env python3

# As in the last example, we will need the os, sys, and qb modules:
import os, sys
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 exhausted all other possibilities
    import qb

def main():
    # The parameters here are the same as before, with exceptions noted
    job = {}
    job['name'] = 'python test job - echo the frame number'

    # This time, we will request 4 instances (previously known as subjobs).
    # By requesting 4 instances, assuming there are 4 open slots on the farm,
    # up to 4 agenda items will be processed simultaneously.
    job['cpus'] = 4

    # In the last example, we used the prototype 'cmdline' which implied a single
    # command being run on the farm. This time, we will use the 'cmdrange' prototype
    # which tells Qube that we are running a command per agenda item.
    job['prototype'] = 'cmdrange'

    package = {}

    # Just like the last example, we create a package parameter called 'cmdline'.
    # This is the command that will be run for every agenda item. QB_FRAME_NUMBER,
    # however, is unique to cmdrange. The text QB_FRAME_NUMBER will be replaced with
    # the actual frame number at run time.
    package['cmdline'] = 'echo QB_FRAME_NUMBER'

    job['package'] = package

    # Now we must create our agenda list. This is an absolutely essential part of
    # submitting jobs with agenda items (i.e. frames).
    # First we define a range. The range is in typical number range format where:
    # 1-5 means frames 1,2,3,4,5
    # 1,3,5 means frames 1,3, and 5
```

```
# 1-5,7 means frames 1,2,3,4,5,7
# 1-10x3 means frames 1,4,7,10
agendaRange = '0-60x10' # will evaluate to 0,10,20,30,40,50,60

# Using the given range, we will create an agenda list using qb.genframes
agenda = qb.genframes(agendaRange)

# Now that we have a properly formatted agenda, assign it to the job
job['agenda'] = agenda

# As before, we create a list of 1 job, then submit the list. Again, we
# could submit just the single job w/o the list, but submitting a list is
# good form.
listOfJobsToSubmit = []
listOfJobsToSubmit.append(job)
listOfSubmittedJobs = qb.submit(listOfJobsToSubmit)
for job in listOfSubmittedJobs:
    print(job['id'])
if __name__ == "__main__":
    main()
```

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

Continue to [Basic python job submission III. SimpleCmd Compatibility and Application Specific Parameters](#)