Interprocess message specifications

V 1.0

The major revision number of this document should match the major revision number of the associated flowchart.

NOTE: As of Feb 1, 2014, we've switched around our job/task terminology so that it (hopefully) makes more sense. **A *Job* is composed of 6k *Tasks*. Users submit *Jobs;* worker nodes process *Tasks*.** This change has been made in both this document and the accompanying diagram document.

Job Queue Message Specification

```
{
  "JobID" : "My Job ID",
  "operations" : {},
  "??" : "??"
}
```

- *Job ID* is a UUID, generated by the job-initiating application, which will be used to identify this job from start to finish. All tasks, results, and reports generated from this Job will be tagged with this ID.
- *operations* is a hash such as we provide to the management ops validator

What else should go in here?

- Something providing field geometry, etc. ?
- A *Final Action*, such as a url to post report to? Email address? Something indicating where results are sent or how the client is notified that the task is done ?

Report Gen Queue Message Specification

```
{
  "job_id" : "My Job ID",
  "task_ids" : ["TaskID_0","TaskID_1","TaskID_...","TaskID_5999"]
}
```

- *Job ID* is the same Job ID as in the Job Queue. This is the name of the queue that report handler's subprocess for this job will poll for task results
- *Task IDs* is an array of unique task ids, each of which is a UUID generated by the GIS Query Node
- *Final Action* (or something similar). See comment in Job Queue section

Task Queue Message Specification

```
---
EZQ:
  result_queue_name: My Job ID
  get_s3_files:
  - bucket: Input_Bucket
    key: filename_1
  - bucket: Input_Bucket
    key: filename_2
  etc.

  put_s3_files:
  - bucket: Result_Bucket
    key: result_file_n
    filename: local_result_file_n
  etc.
...
{
  "task_id" = "Integer ID for this cell/task",
  "boundary" = "",
  "cokey" = "",
  "soil" = "",
  "skel" = "",
  "slope" = 0.0,
  "yields" = [crop1,yield1,crop2,yield2,(...,...)]
}
```

- The initial section between --- and … is a YAML preamble containing EZQ directives.
- *result_queue_name* tells EZQ to place the results of this task into the named queue. This name should be set to the job_id, as that is the name of the results queue created by GIS Query Node for this job
- The *get_s3_files* block in the EZQ preamble lists all the input files needed for this task that have been cached in S3
- The *put_s3_files* block in the EZQ preamble lists all the result files that should be sent back out to S3
- *task_id* is an integer matching one of those in the task_ids array of a Report Gen Queue message
- Everything else is a direct output of Pre-Grid process and represents information the worker must have to complete its task.

Results Queue Message Specification

```
---
EZQ:
  get_s3_files:
  - bucket: Result_Bucket
    key: result_file_n
  etc.
…
{
  "cell_id" : "Integer ID for this cell/task",
  "sci" : 1.234,
  "eros" : 2.345
}
```

- The *get_s3_files* block in the EZQ preamble lists the files that were cached in S3 based on the *put_s3_files* directive from the Task Queue Message. Report Handler will download these files and process them as results.
- *cell_id* is the same as task_id everywhere else in this document. Once we have decided which terminology we want to use, we can standardize this to one of the other.
- The other pieces are the RUSLE2 outputs.

pf_msg Specification

```
"push_file: bucket,filename"
```

This message is placed on STDOUT by the Pre-Grid program, indicating a that the file *filename* placed on local storage should be uploaded to the S3 bucket *bucket*. Job_Breaker will store the location information for each of these files so that it can add the appropriate *get_s3_files* section to the message preamble.

em_msg Specification

```
"{...message...}"
```

This message is placed on STDOUT by the Pre-Grid program, indicating the JSON message *{...message...}* should be placed into the Task Queue. Job_Breaker will automatically prepend a suitable EZQ preamble as detailed in Task Queue Message Specification.