ResamplingStage External Interface Description

- Module: ResamplingStage
- Applicable Module Version:
  - $Id: Resample0.cc,v 1.9 2004/02/21 21:49:29 rayg Exp $
    $Id: ResamplingStage.cc,v 1.8 2004/02/21 21:52:38 rayg Exp $
- Document Revision: 20040221
- Document Maintainer: R.K. Garcia <rayg@ssec.wisc.edu>
- Purpose: This module implements spectral wavenumber resampling. It is used to maintain a standard target effective laser wavenumber across a potentially widely varying set of effective laser wavenumbers, which is an artifact of the GIFTS instrument design.
- This prototype module is a standalone executable intended to be deployed as a data processing stage of a large instrument processing pipeline. It is compatible with the POSIX environment, and typically maintained as an OS-specific binary module (e.g. x86 Linux).
- Input and Output (Data Channel):
  - Environment variable **varInputSpectra** specifies a block file descriptor, record size, and record count of an input file which will then be memory-mapped by the ResamplingStage. The format of this environment setting is “%d,%u,%u” % (file_descriptor, doubles_per_record, records).
    - The records in this input map are indexed from 0.
    - The memory map will be mapped with POSIX mmap with PROT_READ and MAP_SHARED.
    - Each record of the input map will consist of doubles_per_record 64-bit IEEE native word order floating point values corresponding to an input spectrum.
  - Environment variable **varOutputSpectra** specifies a block file descriptor, record size, and record count of an output file which will then be memory-mapped by the ResamplingStage. The format of this environment setting is “%d,%u,%u” % (file_descriptor, doubles_per_record, record_count).
    - The records in this input map are indexed from 0.
    - The memory map will be mapped with POSIX mmap with PROT_READ | PROT_WRITE and MAP_SHARED.
    - Each record of the input map will consist of doubles_per_record 64-bit IEEE native word order floating point values corresponding to an input spectrum.
  - Environment variable **refSamplingSettings** specifies a block file descriptor and record count for an input file which will be memory mapped to provide reference database information. The format of the environment setting is “%d,%u” % (file_descriptor, record_count).
    - The records in this database are indexed from 0.
    - The memory map will be mapped with POSIX mmap with PROT_READ and MAP_SHARED.
    - Each record in the reference database will have this structure.
      - ```
        struct ResamplingSettings_t
        /* is native byte order, IEEE floating point and integer */
        {
        ```
        ```

```
                /** source laser wavenumber -- varies per off-axis angle
                 * Units: cm-1
                 */
                double sourceLaserWavenumber;
                /** target laser wavenumber -- generally constant
                 * Units: cm-1
                 */
                double targetLaserWavenumber;
                /** number of points in the input spectrum -- e.g. 1025 for a 2048
                    interferogram
                 * Units:
                 */
                unsigned long inputPointCount;
                /** quality reduction factor reduces the effective size of the resampling
                    matrix at cost of error.
                 * Units:
                 */
                unsigned long qualityReductionFactor;  // not yet used
                /** decimation factor
                 * Units:
                 */
                unsigned long decimationFactor;  // not yet used
            };
```

- Control Channel
  - The module is controlled through its stdin and stdout (file descriptors 0 and 1). A simple set of commands drives it. Each command is a single line of ASCII text, which will be replied to with a single line of ASCII text. The first scannable element on the reply line will be a result code: 0 for “OK”, less than zero for errors, greater than zero for warnings. This will be followed by an explanation string. (In future interface revisions, more formalized methods of exchanging control channel information may be substituted.)
  - “ping”
    Causes the module to respond with an acknowledgement response.
  -
  - “exit”
    Causes the module to shut down and exit.
  - “resample %u %u %u” % (output_buffer, input_buffer, reference_index)
    Causes the module to do a resampling calculation on a input buffer at index input_buffer, sending the result to the indexed output_buffer, using reference information at reference_index.
- Monitoring Channel
  - Currently, the module sends its monitoring information as lines of text to stderr (file descriptor 2). These include messages
    - “resampling completed”
    - “resampling aborted”
    - “adding entry for detector %d to resampling cache”
    - “removing entry for detector %d to resampling cache”
    - “resampling cache flushed”
    - “ERROR - unable to find resampling settings for %d”
  - In future interface revisions, the format and content of these messages may be changed to a more formal reporting mechanism.

- Auditing Channel
  - Currently, the ResamplingStage does not provide any information to an auditing channel.
- Typical usage.
  - The controlling process is responsible for providing the file descriptors for the input and output data and initializing an environment for the ResamplingStage.
  - A set of unit tests written in python, starting with simpletest0.py, are included as examples of creating data and control channels as file descriptors, forking and execing the stage, and passing data through it.
- <<END>>