BS Errors to ESD

Very brief overview of dataflow

- Taken from this talk:
  - http://indico.cern.ch/getFile.py/access?contribId=45&sessionId=14&resId=0&materialId=slides&confId=22936
- Reporting errors in the BS:
  - Several types of hardware error can be recorded by the DAQ as a bit pattern in the ByteStream.
    - Header errors:
    - Trailer errors
    - Errors in hits, or errors taking the place of a hit.
- Need to keep track of these for offline reconstruction, and for HLT
- Need to keep track of which modules are giving errors, in order to flag them for offline reconstruction (and show them in monitoring)

SCT_ByteStreamErrorsSvc

- Offline
  - Stores a vector of IdHashes for each type of error,
  - Filled by SCT_RodDecoder, and can be retrieved by any downstream algorithm.
  - Interface is compatible with SCT_ConditionsSummarySvc (though could/will be improved).
- HLT
  - LVL2 uses SCT_RodDecoder but since it works in terms of RoIs not events:
    - A once-per-event record of errors may not be useful.
    - Also, HLT doesn't necessarily need to know which modules gave the errors, just needs the counts.
  - Two approaches:
    - In SCT_RodDecoder::fillCollections(), count the different types of errors and encode into an int, which is returned as an IssueSeverity along with StatusCode::RECOVERABLE as the return value of fillCollections() .
    - Add new functionality to SCT_ByteStreanErrorsSvc so it can provide counts of errors rather than vectors of IdHashes, and a function to reset these counts.

VERY Brief overview of Error content

- Packages:
  - InnerDetector/InDetConditions/InDetByteStreamErrors - transient containers
  - InnerDetector/InDetConditions/InDetConditionsAthenaPool - persistent containers
- Store one entry per error, with the IdHash and the type of error (represented by an int):
  - DataVector< std::pair< IdHash, int> >