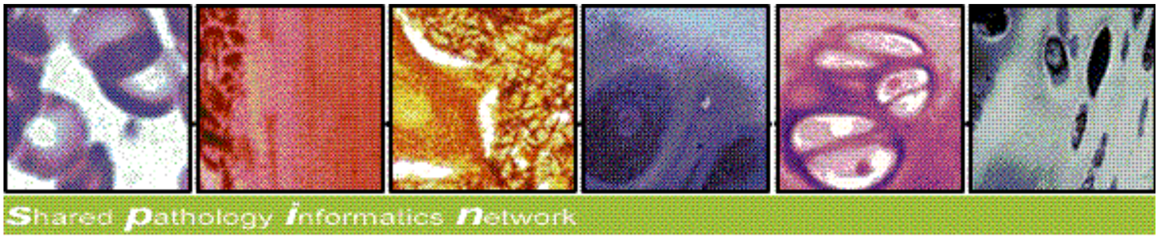
ETL VERSION: 1.3.4.1
DATE: 5/19/11

Title: **VSL ETL Developer Guide**

Author: Andrew McMurry
Contact: Andrew_mcmurry@hms.harvard.edu



Intended Audience:

Developers looking to extend, customize, or contribute to this scrubber utility.
It is assumed the reader has already reviewed the *VSL ETL User Guide.*

ETL Overview:

The ETL program *Extracts* patient information from existing systems, *Transforms* the results into standard HIPAA safe medical vocabularies, and *Loads* the results into a locally controlled SPIN Peer.

The ETL is an upgrade to the original SPIN Submission Suite with four notable improvements:
(1) ETL is highly modular and configurable to new types of clinical data,
(2) ETL allows easy deployment with interfaces to JDBC databases, XML exports, and SOAP
(3) ETL supports a Two Phase Commit protocol to ensure data integrity during synchronization.
(4) ETL can be run on demand or as a scheduled task synchronizing cases within a date range.

Definitions:

ETL: **(1) E**xtract **(2) T**ransform **(3) L**oad
Extract: Extract from existing database, XML file, or custom interface
Transform: Transform results of the extraction (autocoding and de-identification)
Load: Load transformed results into the SPIN peer, DB, or XML file

Required Software:

Java 1.5 or later
MySQL 5 or later

ETL VSL Step Loader Sequence Diagram

ETLStepLoaderConfig.isPerformDelete = false

1. Check accession number in codebook
    a. If Exists:
        i. Set existing CaseUUID from codebook to PathologyCaseVO
        ii. SHA256(xml)
        iii. Check new SHA256(xml) against the saved SHA256(xml) in codebook
            1. Match:
                a. Log.warn(Skip, no change)
            2. Differ:
                a. Log.warn(case has changed resubmitting)
                b. Update in peer
                    i. If ACK
                        1. Update in codebook by setting the new SHA256(xml)
                    ii. If NACK
                        1. Throw new ETLRuntimeException

    b. If NOT exists:
        i. assign *CaseUUID (not specimenUUID or patientUUID)*
        ii. new SHA256(xml)
        iii. submit to peer
            1. SHA256(xml) input
            2. persist
            3. return acknack
        iv. if ACK:
            1. compare codebook SHA256(xml) & peer SHA256(xml)
                a. match:
                    i. save codebook SHA256(xml) to codebook
                b. differ:
                    i. log error
                    ii. throw new ETLRuntimeException
                    note: this checks code level failures, not case input failures hence we stop if we detect code failure.
        v. *If Not ACK:*
            1. Throw new ETLRuntimeException
            note: assumes ETL will validate PathologyCase VO

ETLStepLoaderConfig.isPerformDelete = true

```
1. Check accession number in codebook
    a. If Exists:
        i. Set existing CaseUUID from codebook to PathologyCaseVO
        ii. SHA256(xml)
        iii. Check new SHA256(xml) against the saved SHA256(xml) in
             codebook
            1. Match:
                a. Peer.delete
                b. If ACK
                    i. Log.info
                       (accession Num + caseUUID + hash)
                    ii. delete from codebook
                c. If NACK
                    i. throw new ETLRuntimeException
            2. Differ:
                a. Log.warn
                   (case has changed but you are deleting anyway)
                b. Delete from peer
                    i. If ACK
                        1. Log.info
                           (accession Num + caseUUID +
                           hash)
                        2. Delete from codebook
                    ii. If NACK
                        1. Throw new ETLRuntimeException

    b. If NOT exists:
        i. Log.warn(“case does not exist”)
```