org.jax.mgi.shr.dla.input Design Document

Author: Mike Walker

Created: March 20, 2003

Last Modified: December 1, 2004 10:29

1 Purpose of Document

This document describes the classes belonging to the org.jax.mgi.shr.dla.input package from the lib_java_dla product. There are three sub packages to this package which also will be discussed within this document. They are the embl, fasta, genbank, and tigr packages.

2 Introduction

This package is comprised of classes that are related to data input. Various data providers are represented within this package and each of them make up a sub package of the org.jax.mgi.shr.dla.input package. This package integrates with the lib_java_core product. You must be pointing to Java1.4 in your classpath when using this package. Most of the classes can be classified into a few general design patterns. Theses patterns are represented in table below.

Table 1: Design Patterns for Data Input

class category	category description
Input File Classes	Subclasses of the org.jax.mgi.shr.ioutils.InputDataFile class. These classes represent an input file. They define the input file record structure by setting start and end delimiters. They also provide iterators for iterating through the records and the ability to plug in interpreter classes (see org.jax.mgi.shr.ioutils.RecordDataInterpreter).
Data Interpreter Classes	Subclasses of the org.jax.mgi.shr.ioutils.RecordDataInterpreter class. These classes provide record parsing functionality. They create Java objects which represent an input record. They get plugged in to the iterators from the InputDataFile classes so that applications can iterate over Java input objects instead of simply text records.
Input Objects	These classes are the Java objects returned from the RecordDataInterpreters (see org.jax.mgi.shr.ioutils.RecordDataInterpreter). They typically are the input objects for load application processing.

Table 1: Design Patterns for Data Input

class category	category description
Input Filter Classes	These classes are responsible for filtering input records prior to being processed by the load applications by either changing the record attributes or declining a record.

3 Overview of Classes

This section is broken into sub sections based on the categories outlines in table 1.

3.1 Input File Classes

The **FASTAInputFile** extends **org.jax.mgi.shr.ioutils.InputDataFile** from the lib_java_core product. It represents a FASTA formatted file and provides functionality for iterating through the records in the file. This class is a member of the fasta sub package.

The **GBInputFile** extends **org.jax.mgi.shr.ioutils.InputDataFile** from the lib_java_core product. It represents a GenBank formatted file and provides functionality for iterating through the records in the file. This class is a member of the genbank sub package.

The **GBInputFileNoSeq** extends **org.jax.mgi.shr.ioutils.InputDataFile** from the lib_java_core product. It represents a GenBank formatted file without the GenBank nucleotide data. It provides functionality for iterating through the records in the file. This class is a member of the genbank sub package.

The **EMBLInputFile** extends **org.jax.mgi.shr.ioutils.InputDataFile** from the lib_java_core product. It represents a EMBL formatted file and provides functionality for iterating through the records in the file. This class is a member of the embl sub package.

The **EMBLInputFileNoSeq** extends **org.jax.mgi.shr.ioutils.InputDataFile** from the lib_java_core product. It represents a EMBL formatted file without the sequence data. It provides functionality for iterating through the records in the file. This class is a member of the embl sub package.

3.2 Input Objects

The **SequenceInput** represents one record of data which can be processed by the **SeqProcessor** class (see documentation for the org.jax.mgi.dbs.mgd.loads.Seq package). The **SeqProcessor** class is the core processor for all the sequence load applications (GenBank, Swiss-Prot, TIGR, Dots, etc).

The **FASTAData** class is an object which represents one FASTA data element. It has an identity attribute and a sequence attribute. It is used by the **FASTALoader** class (see org.jax.mgi.shr.dla.loader documentation). This class is a member of the fasta sub package.

3.3 Data Interpreter Classes

The **SequenceInterpreter** class implements **org.jax.mgi.shr.ioutils.RecordDataInterpreter** from the lib_java_core product. It is used to parse a single record from an input file and create a Java input object for subsequent processing by a particular load application.

The **GBFormatInterpreter** class extends **SequenceInterpreter**. It parses GenBank records and creates **SequenceInput** objects. This class is a member of the genbank sub package.

The **EMBLFormatInterpreter** class extends **SequenceInterpreter**. It parses EMBL records and creates **SequenceInput** objects. This class is a member of the embl sub package.

The **MGSAssemblyFormatInterpreter** class extends **SequenceInterpreter.** It parses MGS assembly records and creates **SequenceInput** objects. This class is a member of the mgs subpackage.

The **MGSCoordinateFormatInterpreter** class extends **SequenceInterpreter.** It parses MGS assembly records and creates **SequenceInput** objects. This class is a member of the mgs subpackage.

3.4 Input Filter Classes

The **FASTAFilter** interface is used by the **GeneIndexLoader** class to encapsulate specific logic for filtering input data before it is processed by the load application. The **GeneIndexLoader** obtains the implementaion at runtime through a configuration parameter.

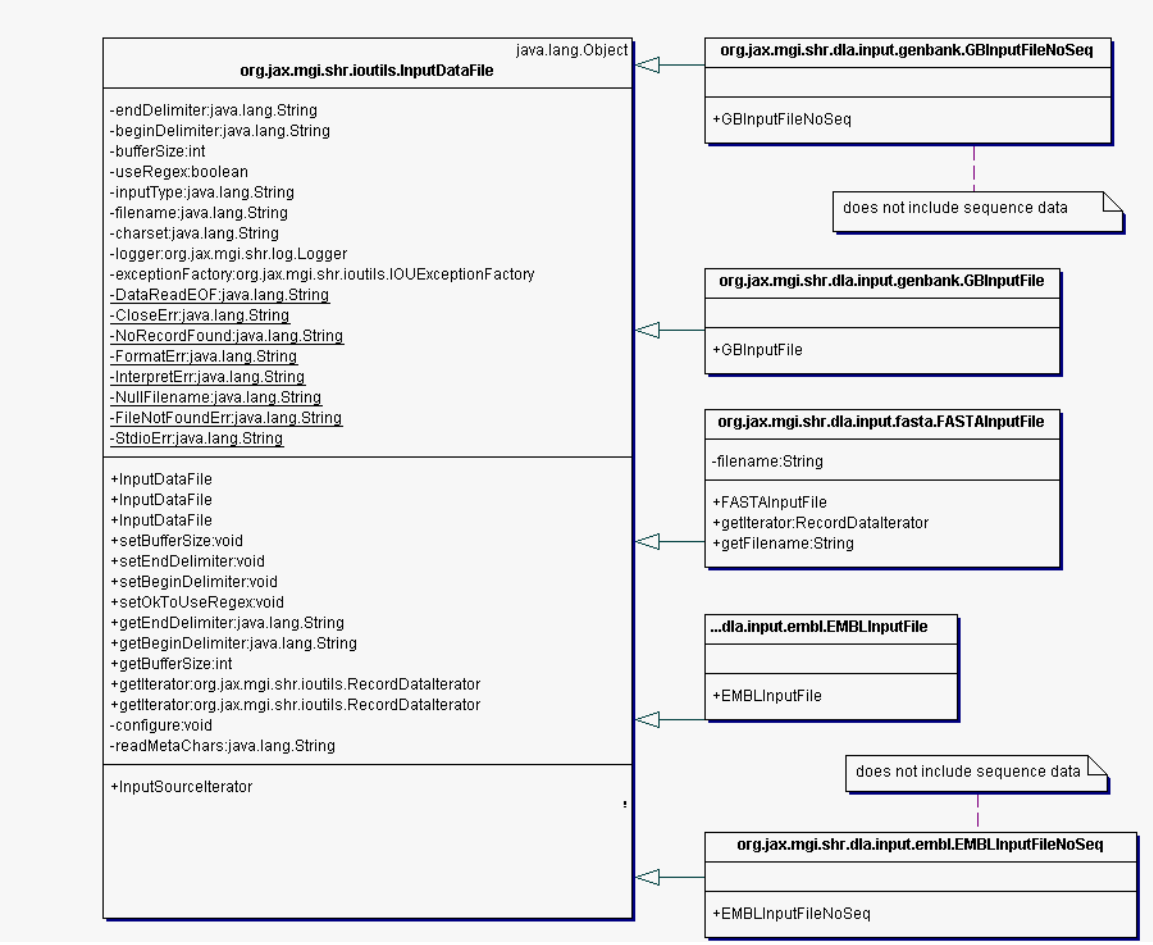
The **TIGRFilter** class is an implementation of the **FASTAFilter**. It only accepts TC sequences from the input file.

3.5 Miscellaneous Classes

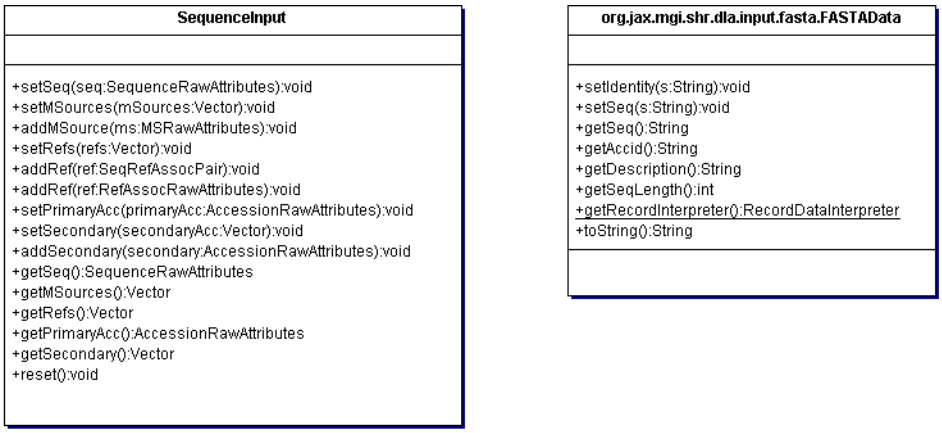
The **DateConverter** is used to parse date text from genbank and EMBL formats.

4 Class Diagrams

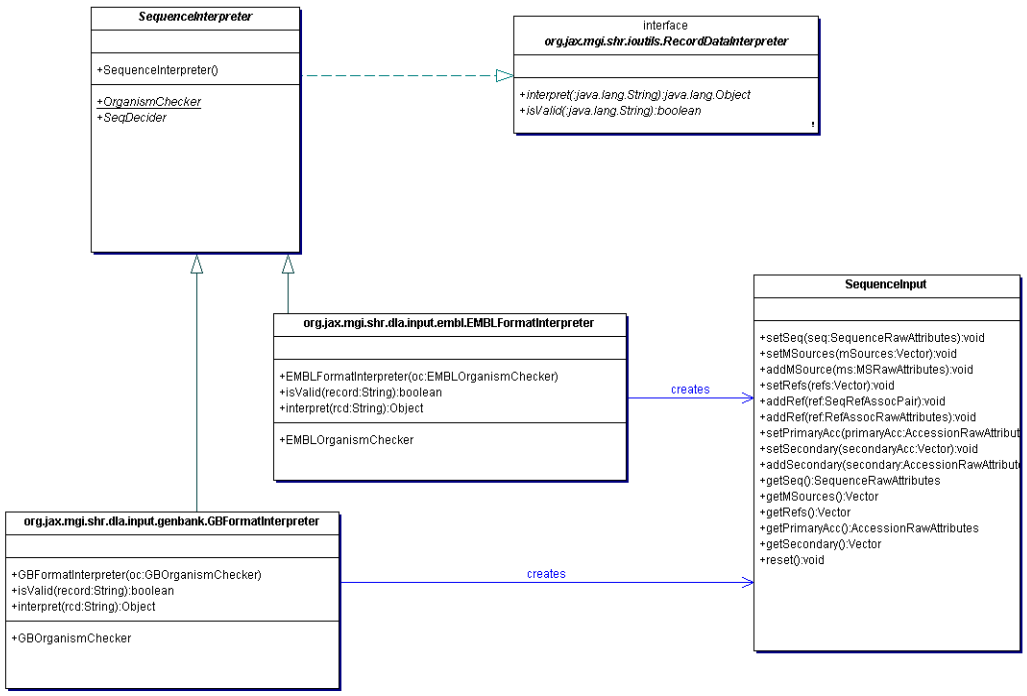
4.1 Input File Classes



4.2 Input Objects



4.3 Data Interpreter Classes



4.4 Input Filter Classes