* Seekable

Seekable是一个接口，作用是能指定当前处理文件中读取位置的偏移量，获取偏移量以及从某个位置开始查找一个新的数据拷贝。声明了三个方法：

（1）void seek(long pos) throws IOException

用于设定相当于文件头的偏移量。

（2）long getPos() throws IOException

方法用于返回当前的偏移量。

（3）boolean seekToNewSource(long targetPos) throws IOException

返回从指定位置开始是否存在一个新的数据拷贝。

* PositionedReadable

PositionedReadable是一个接口，作用是能够从指定位置开始读取数据。声明了以下方法：

（1）int read(long position, byte[] buffer, int offset, int length)

这是个保障线程安全的方法，功能是从指定位置开始读取字节并返回读取的字节数，且不改变文件的偏移量。

（2）void readFully(long position, byte[] buffer, int offset, int length)

throws IOException

这是个保障线程安全的方法，功能是从指定位置开始读取指定字节数的数据，且不改变文件的偏移量

（3）void readFully(long position, byte[] buffer) throws IOException

这是个保障线程安全的方法，从指定位置开始读取文件长度个字节数的数据，且不改变文件的偏移量

* Syncable

Syncable是一个接口，作用是能使实现它的类方便的进行缓冲区的清空，将数据都存入磁盘。

* BufferedFSInputStream

BufferedFSInputStream是一个继承了BufferedInputStream，实现了PositionedReadable和Seekable接口的抽象类。实现了以下若干方法：

（1）int read(long position, byte[] buffer, int offset, int length) throws IOException

从指定位置开始读取字节并返回读取的字节数，且不改变文件的偏移量

（2）void readFully(long position, byte[] buffer, int offset, int length) throws IOException

从指定位置开始读取指定字节数的数据，且不改变文件的偏移量

（3）public long skip(long n) throws IOException

将偏移量向后移动一段距离，也就是跳过一定字节数。

* FSDataInputStream

FSDataInputStream是一个继承了DataInputStream，实现了PositionedReadable和Seekable接口的类。实现了以下若干方法：

（1）synchronized void seek(long desired) throws IOException:

用于设定相当于文件头的偏移量。

（2）long getPos() throws IOException：

用于返回当前的偏移量。

（3）int read(long position, byte[] buffer, int offset, int length)

throws IOException

从指定位置开始读取字节并返回读取的字节数，且不改变文件的偏移量。

（4）void readFully(long position, byte[] buffer, int offset, int length)

throws IOException

从指定位置开始读取指定字节数的数据，且不改变文件的偏移量

（5）boolean seekToNewSource(long targetPos) throws IOException

返回从指定位置开始是否存在一个新的数据拷贝。

* FSDataOutputStream

FSDataOutputStream是一个继承了DataOutputStream，实现了Syncable接口的类。内部包裹了OutputStream对象，通过一个BufferedOutputStream缓冲输出并生成一个统计文件，实现了以下若干方法：

（1）void write(int b) throws IOException

用于向out中写入一个字节，并更新统计文件。

（2）void sync() throws IOException

wrappedStream是必须实现Syncable接口的流类，强制同步全部缓冲区。

* FSInputStream

FSInputStream是一个继承了InputStream，实现了PositionedReadable和Seekable接口的抽象类。实现了以下若干方法：

（1）int read(long position, byte[] buffer, int offset, int length) throws IOException

在从指定位置读取指定字节数的数据后，将偏移量复原，返回读取的字节个数

（2）void readFully(long position, byte[] buffer, int offset, int length)

throws IOException

从指定位置读取指定字节数的数据。

（3）void readFully(long position, byte[] buffer) throws IOException

读取整个文件

* FSOutputSummer

FSOutputSummer是一个用于在数据写入输出流之前统计数据数据量的抽象类

* FTPInputStream

FTPInputStream类继承了FSInputStream。其中包括一个InputStream对象，FTPClient客户端对象，FileSystem.Statistics文件系统统计量对象，不支持seek()方法。

read()读取功能由包裹的InputStream对象完成。

* KFSInputStream

KFSInputStream是一个继承了FSInputStream的类，功能是能够读取存储在KFS文件系统中的数据。KFS是一种高性能的，为网络应用存储日志文件的分布式文件系统。定义了如下几个独特方法：

（1）int available() throws IOException

用于返回当前偏移量到文件结尾的距离。

* KFSOutputStream

KFSOutputStream是一个继承了FSOutputStream的类，功能是能够将数据存储到KFS文件系统中。包含了progressReporter用于在数据存入KFS文件系统之前记录进度。

* S3InputStream

S3InputStream是用于读取以Amazon S3方式存储的文件的一个类。定义了若干独有的方法：

（1）void blockSeekTo(long target) throws IOException

用于找到目标所在块并读取其中数据

* S3OutputStream

S3InputStream是用于读取以Amazon S3方式存储的文件的一个类。定义了若干独有的方法

（1）File newBackupFile() throws IOException

用于生成一个新的备份文件。

（2）void flushData(int maxPos) throws IOException

用于在本地块数据备份，

（3）void endBlock() throws IOException

标志着结束本地备份并发送给S3，成功后便删除临时备份文件生成一个全新的备份文件。

（4）void nextBlockOutputStream() throws IOException

用于新建一个S3 Block输出流。