FilterFs

这个类包含了一些其他的文件系统，作为基本文件系统，可以在其中重写文件系统的某些方法以满足需求，而FilterFs 的子类也同样可以继承它宠妾一些其他的方法

void checkPath(Path)： 检查路径是否符合该文件系统

createInternal(Path f, java.util.EnumSet<CreateFlag> flag, FsPermission absolutePermission, int bufferSize, short replication, long blockSize, Progressable progress, Options.ChecksumOpt checksumOpt, boolean createParent)：同AbstractFileSystem.create()方法

createSymlink(Path target, Path link, boolean createParent)：同FileContext.createSymlink()方法

delete(Path f, boolean recursive)：同FileContext.delete()方法

getCanonicalServiceName()：得到文件系统的规范名

getDelegationTokens(java.lang.String renewer)：得到文件系统的代理记号

getFileBlockLocations(Path f, long start, long len)：FileContext.getFileBlockLocations()

getFileChecksum(Path f)：FileContext.getFileChecksum()

getFileLinkStatus(Path f)：FileContext.getFileLinkStatus(Path)，并且抛出异常

getFileStatus(Path f)：FileContext.getFileStatus(Path)，抛出异常

getFsStatus()：FileContext.getFsStatus(Path)

getFsStatus(Path f)：FileContext.getFsStatus(Path)

getHomeDirectory()：返回用户在该文件系统中的根目录

getInitialWorkingDirectory()：得到本地文件系统的初始工作目录

getLinkTarget(Path f)：FileContext.getLinkTarget(Path);

getServerDefaults()：得到默认的服务器参数

getUri()：得到一个唯一标识该文件系统的URI

getUriDefaultPort()：得到默认接口

getUriPath(Path p)：得到部分路径名

listStatus(Path f)：FileContext.Util.listStatus(Path)

makeQualified(Path path)：是路径名符合文件系统的命名标准

mkdir(Path dir, FsPermission permission, boolean createParent)：FileContext.mkdir(Path, FsPermission, boolean)

open(Path f)：FileContext.open(Path)

open(Path f, int bufferSize)：FileContext.open(Path, int)

renameInternal(Path src, Path dst)：FileContext.rename(Path, Path, Options.Rename...)

renameInternal(Path src, Path dst, boolean overwrite)：FileContext.rename(Path, Path, Options.Rename...)

resolvePath(Path p)：返回一个标准路径名，去除其中的内部链接和挂载点

setOwner(Path f, java.lang.String username, java.lang.String groupname)：FileContext.setOwner(Path, String, String)

setPermission(Path f, FsPermission permission)：setPermission(Path f, FsPermission permission)

setReplication(Path f, short replication)：FileContext.setReplication(Path, short)

setTimes(Path f, long mtime, long atime)：FileContext.setTimes(Path, long, long)

setVerifyChecksum(boolean verifyChecksum)：FileContext.setVerifyChecksum(boolean, Path)

supportsSymlinks()：返回是否支持内部链接

ChecksumFs.java

继承自FilterFs，提供了一个最基本的文件系统校验的实现，为每个原始文件生成一个校验文件，并且在客户端进行校验

createInternal(Path f, java.util.EnumSet<CreateFlag> createFlag, FsPermission absolutePermission, int bufferSize, short replication, long blockSize, Progressable progress, Options.ChecksumOpt checksumOpt, boolean createParent)：AbstractFileSystem.create(Path, EnumSet, Options.CreateOpts...)

delete(Path f, boolean recursive)：ChecksumFileSystem.delete()

getBytesPerSum()：单个校验的字节数

getChecksumFile(Path file)：该校验文件对应的原文件名

getChecksumLength(long size, int bytesPerSum)：校验文件的长度

getRawFs()：得到原文件系统

isChecksumFile(Path file)：返回该文件是否是一个校验文件

listStatus(Path f)：FileContext.Util.listStatus(Path)

open(Path f, int bufferSize)：通过给定的路径打开一个数据输入流

renameInternal(Path src, Path dst)：重命名

reportChecksumFailure(Path f, FSDataInputStream in, long inPos, FSDataInputStream sums, long sumsPos)：返回校验错误

setReplication(Path src, short replication)：复制文件

setVerifyChecksum(boolean inVerifyChecksum)：设定是否进行校验

LocalFs.java

实现了ChecksumFS的更底层的子类

DelegateToFileSystem.java

作为一个代理，以已经实现了的FileSystem为基础，实现AbstractFileSystem

createInternal(Path f, java.util.EnumSet<CreateFlag> flag, FsPermission absolutePermission, int bufferSize, short replication, long blockSize, Progressable progress, Options.ChecksumOpt checksumOpt, boolean createParent)：AbstractFileSystem.create(Path, EnumSet, Options.CreateOpts...)

createSymlink(Path target, Path link, boolean createParent)：FileContext.createSymlink(Path, Path, boolean);

delete(Path f, boolean recursive)：FileContext.delete(Path, boolean)

getCanonicalServiceName()：得到文件系统的规范名

getDelegationTokens(java.lang.String renewer)：得到文件系统的代理符号

getFileBlockLocations(Path f, long start, long len)：FileContext.getFileBlockLocations(Path, long, long)

getFileChecksum(Path f)：FileContext.getFileChecksum(Path)

getFileLinkStatus(Path f)：FileContext.getFileLinkStatus(Path)

getFileStatus(Path f)：FileContext.getFileStatus(Path)

getFsStatus()：FileContext.getFsStatus(Path).

getInitialWorkingDirectory()：得到本地文件系统的初始工作目录

getLinkTarget(Path f)：FileContext.getLinkTarget(Path);

getServerDefaults()：得到服务器的默认参数

getUriDefaultPort()：默认端口

listStatus(Path f)：FileContext.Util.listStatus(Path)

mkdir(Path dir, FsPermission permission, boolean createParent)：FileContext.mkdir(Path, FsPermission, boolean)

open(Path f, int bufferSize)：FileContext.open(Path, int)

renameInternal(Path src, Path dst)：FileContext.rename(Path, Path, Options.Rename...)

setOwner(Path f, java.lang.String username, java.lang.String groupname)：FileContext.setOwner(Path, String, String)

setPermission(Path f, FsPermission permission)：FileContext.setPermission(Path, FsPermission)

setReplication(Path f, short replication)：FileContext.setPermission(Path, FsPermission)

FileContext.java

这个类提供了一系列的接口，供应用程序的开发者使用

clearStatistics()：清空文件系统中所有AbstractFileSystem中的Statistics数据

create(Path f, java.util.EnumSet<CreateFlag> createFlag, Options.CreateOpts... opts)：在给定的路径名下创建一个文件的输入流，新建或者覆盖一个文件

createSymlink(Path target, Path link, boolean createParent)：为存在的文件创建符号链接

delete(Path f, boolean recursive)：删除文件

deleteOnExit(Path f)：在JVM结束时删除文件

getDelegationTokens(Path p, java.lang.String renewer)：在给定的文件目录下，得到文件系统允许查看的所有代理记号

getFileBlockLocations(Path f, long start, long len)：找到文件存储的块的位置和偏移量

getFileChecksum(Path f)：得到文件校验

getFileContext()：从$HADOOP\_CONFIG/core.xml中得到默认参数，以此来创建一个FileContext类

getFileContext(AbstractFileSystem defaultFS)：用默认参数为特定的文件系统创建一个FileContext

getFileContext(AbstractFileSystem defFS, Configuration aConf)：创建一个filecontext与指定的FS作为默认使用指定的配置

getFileContext(Configuration aConf)：通过传递进来的参数创建FileContext

getFileContext(java.net.URI defaultFsUri)：为特定的URI和默认参数创建FileContext

getFileContext(java.net.URI defaultFsUri, Configuration aConf)：为特定的URI和给定的参数创建FileContext

getFileLinkStatus(Path f)：返回一个代表路径的文件状态

getFsStatus(Path f)：

getHomeDirectory()：返回文件系统的根目录

getLinkTarget(Path f)：根据给定的符号链接返回目标路径

getUgi()：得到ugi

getWorkingDirectory()：得到工作目录

listLocatedStatus(Path f)：当路径是一个目录时，列出所有的文件或者目录状态

listStatus(Path f)：当路径是一个目录时，列出所有的文件或者目录状态

makeQualified(Path path)：使路径完全符合文件系统的要求

mkdir(Path dir, FsPermission permission, boolean createParent)：创造一个目录

open(Path f)：通过给定的路径创建一个输入流

open(Path f, int bufferSize)

printStatistics()：通过标准化输出打印所有数据

rename(Path src, Path dst, Options.Rename... options)：重命名路径

resolve(Path f)：分解所有的符号链接

resolveIntermediate(Path f)

setOwner(Path f, java.lang.String username, java.lang.String groupname)：为一个路径设置一个拥有着

setPermission(Path f, FsPermission permission)：为路径设置一个权限

setReplication(Path f, short replication)：复制文件

setTimes(Path f, long mtime, long atime)：设置可修改时间

setUMask(FsPermission newUmask)：根据提供的参数设置umask

setVerifyChecksum(boolean verifyChecksum, Path f)：设置校验位

setWorkingDirectory(Path newWDir)：创建工作目录

util()