|  |  |
| --- | --- |
| **FileTypeDetector** | Liên quan |
| A file type detector for probing a file to guess its file type.  A file type detector is a concrete implementation of this class, has a zero-argument constructor, and implements the abstract methods specified below.  The means by which a file type detector determines the file type is highly implementation specific. A simple implementation might examine the file extension (a convention used in some platforms) and map it to a file type. In other cases, the file type may be stored as a file [attribute](file:///E:\Data\Software\eclipse\Local\h2\H2%20gae%20Source\src\com\newatlanta\repackaged\java\nio\file\attribute\package-summary.html) or the bytes in a file may be examined to guess its file type. | ➊**SecurityManager**  ➊**RuntimePermission**  ➊**FileRef[2]** |

|  |  |
| --- | --- |
| **FileSystemProvider** | Liên quan |
| Service-provider class for file systems.  A file system provider is a concrete implementation of this class that implements the abstract methods defined by this class. A provider is identified by a URI [scheme](eclipse-javadoc:%E2%98%82=gaevfs/src%3Ccom.newatlanta.repackaged.java.nio.file.spi%7BFileSystemProvider.java%E2%98%83FileSystemProvider%E2%98%82%E2%98%82getScheme%E2%98%82). The default provider is identified by the URI scheme "file". It creates the [FileSystem](eclipse-javadoc:%E2%98%82=gaevfs/src%3Ccom.newatlanta.repackaged.java.nio.file.spi%7BFileSystemProvider.java%E2%98%83FileSystemProvider%E2%98%82FileSystem) that provides access to the file systems accessible to the Java virtual machine. The [FileSystems](eclipse-javadoc:%E2%98%82=gaevfs/src%3Ccom.newatlanta.repackaged.java.nio.file.spi%7BFileSystemProvider.java%E2%98%83FileSystemProvider%E2%98%82FileSystems) class defines how file system providers are located and loaded. The default provider is typically a system-default provider but may be overridden if the system property java.nio.file.spi.DefaultFileSystemProvider is set. In that case, the provider has a one argument constructor whose formal parameter type is FileSystemProvider. All other providers have a zero argument constructor that initializes the provider.  A provider is a factory for one or more [FileSystem](eclipse-javadoc:%E2%98%82=gaevfs/src%3Ccom.newatlanta.repackaged.java.nio.file.spi%7BFileSystemProvider.java%E2%98%83FileSystemProvider%E2%98%82FileSystem) instances. Each file system is identified by a URI where the URI's scheme matches the provider's [scheme](eclipse-javadoc:%E2%98%82=gaevfs/src%3Ccom.newatlanta.repackaged.java.nio.file.spi%7BFileSystemProvider.java%E2%98%83FileSystemProvider%E2%98%82%E2%98%82getScheme). The default file system, for example, is identified by the URI "file:///". A memory-based file system, for example, may be identified by a URI such as "memory:///?name=logfs". The [newFileSystem](eclipse-javadoc:%E2%98%82=gaevfs/src%3Ccom.newatlanta.repackaged.java.nio.file.spi%7BFileSystemProvider.java%E2%98%83FileSystemProvider%E2%98%82%E2%98%82newFileSystem) method may be used to create a file system, and the [getFileSystem](eclipse-javadoc:%E2%98%82=gaevfs/src%3Ccom.newatlanta.repackaged.java.nio.file.spi%7BFileSystemProvider.java%E2%98%83FileSystemProvider%E2%98%82%E2%98%82getFileSystem) method may be used to obtain a reference to an existing file system created by the provider. Where a provider is the factory for a single file system then it is provider dependent if the file system is created when the provider is initialized, or later when the newFileSystem method is invoked. In the case of the default provider, the FileSystem is created when the provider is initialized.  In addition to file systems, a provider is also a factory for [FileChannel](eclipse-javadoc:%E2%98%82=gaevfs/src%3Ccom.newatlanta.repackaged.java.nio.file.spi%7BFileSystemProvider.java%E2%98%83FileSystemProvider%E2%98%82FileChannel) and [AsynchronousFileChannel](eclipse-javadoc:%E2%98%82=gaevfs/src%3Ccom.newatlanta.repackaged.java.nio.file.spi%7BFileSystemProvider.java%E2%98%83FileSystemProvider%E2%98%82AsynchronousFileChannel) channels. The [newFileChannel](eclipse-javadoc:%E2%98%82=gaevfs/src%3Ccom.newatlanta.repackaged.java.nio.file.spi%7BFileSystemProvider.java%E2%98%83FileSystemProvider%E2%98%82%E2%98%82newFileChannel) and [AsynchronousFileChannel](eclipse-javadoc:%E2%98%82=gaevfs/src%3Ccom.newatlanta.repackaged.java.nio.file.spi%7BFileSystemProvider.java%E2%98%83FileSystemProvider%E2%98%82%E2%98%82newAsynchronousFileChannel) methods are defined to open or create files, returning a channel to access the file. These methods are invoked by static factory methods defined in the [com.newatlanta.repackaged.java.nio.channels](eclipse-javadoc:%E2%98%82=gaevfs/src%3Ccom.newatlanta.repackaged.java.nio.file.spi%7BFileSystemProvider.java%E2%98%83FileSystemProvider%E2%98%82com.newatlanta.repackaged.java.nio.channels) package.  All of the methods in this class are safe for use by multiple concurrent threads. | ➊**List, ArrayList,Set**  ➊**SercurityManager**  ➊**RuntimePermission**  ➊**ServiceLoader**  ➊**ClassLoader**  ➊**AccessController**  ➊**PrivilegedAction**  ➊**Map**  ➊**URI**  ➊**ExecutorService**  ➊**FileSystem[2]**  ➊**Path[2]**  ➊**FileRef[2]**  ➊**FileChannel[4]**  ➊**OpenOption[2]**  ➊**FileAttribute[2]**  ➊**AsynchronousFileChannel[3]** |