Overview Package Class Tree Index Help

PREV CLASS NEXT CLASS **FRAMES NO FRAMES All Classes**

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

edu.mit.csail.aeolus.api

Class AeolusFile

```
java.lang.Object
  └ edu.mit.csail.aeolus.api.AeolusFile
```

`public class` **`AeolusFile`**`extends java.lang.Object`

This class is used to create and manipulate AeolusFiles.

Constructor Summary
`AeolusFile``(java.lang.String hostname, java.lang.String path)`

Method Summary	
`boolean`	**`createNewFile`**`(AeolusLabel sLabel, AeolusLabel iLabel)` Creates a new file if the parent directory exists and there is no file with the given name.
`boolean`	**`delete`**`()` Deletes the file.
`AeolusLabel`	**`getIntegrity`**`()` Returns the integrity label of the file
`AeolusLabel`	**`getSecrecy`**`()` Returns the secrecy label of the file
`java.lang.String[]`	**`list`**`()` Lists the contents of the directory as an array of file names.
`boolean`	**`mkdir`**`(AeolusLabel sLabel, AeolusLabel iLabel)` Creates a new directory if the parent directory exists and there is no file with the given name.

Methods inherited from class java.lang.Object
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructor Detail

AeolusFile

```
public AeolusFile(java.lang.String hostname,
                  java.lang.String path)
```

Method Detail

createNewFile

```
public boolean createNewFile(AeolusLabel sLabel,
                             AeolusLabel iLabel)
                      throws java.io.FileNotFoundException,
                             InfoFlowControlException,
                             java.io.IOException,
                             AeolusException
```

Creates a new file if the parent directory exists and there is no file with the given name. See reference manual for constraints on labels. Returns true if file was successfully created and false if file with same name existed. Please see the reference manual for rules concerning the root directory.

Parameters:

`sLabel` -- file’s secrecy label

`iLabel` -- file’s integrity label

Returns:

true if the file was successfully created

Throws:

`InfoFlowControlException` -- if caller is not allowed to read and write the file’s parent directory or if it is not allowed to read a directory in the path. Also the file’s labels cannot be less constraining than the thread’s labels. Please see the reference manual for rules concerning the root directory.

`java.io.FileNotFoundException` -- if the parent directory (or some part of the path) does not exist

`java.io.IOException` -- if an I/O error occurs while creating the file

`AeolusException` -- if there was an error communicating with hostname

delete

```
public boolean delete()
               throws java.io.FileNotFoundException,
                      InfoFlowControlException,
                      java.io.IOException,
                      AeolusException
```

Deletes the file. Caller labels must allow reading and writing of the parent directory.

Returns:

true if the file was successfully deleted

Throws:

`java.io.FileNotFoundException` -- if the file does not exist and the caller labels allow reading its parent directory

`java.io.IOException` -- if an I/O error occurred while reading directory labels

`InfoFlowControlException` -- if caller labels don't allow reading and writing of the parent directory or reading a higher parent directory in the case the file is missing

`AeolusException` -- if a failure occurs while communicating with hostname

---

getIntegrity

```
public AeolusLabel getIntegrity()
                         throws java.io.FileNotFoundException,
                                InfoFlowControlException,
                                java.io.IOException,
                                AeolusException
```

Returns the integrity label of the file

Returns:

- a Label object representing the integrity label of the file

Throws:

`java.io.FileNotFoundException` -- if the file does not exist on hostname

`java.io.IOException` -- if an I/O error occurs while retrieving the label

`InfoFlowControlException` -- if caller labels don't allow reading the directories in the path

`AeolusException` -- if communication with hostname fails

---

getSecrecy

```
public AeolusLabel getSecrecy()
                       throws java.io.FileNotFoundException,
                              InfoFlowControlException,
                              java.io.IOException,
                              AeolusException
```

Returns the secrecy label of the file

Returns:

- a Label object representing the secrecy label of the file

Throws:

`java.io.FileNotFoundException` -- if the file does not exist on hostname
`java.io.IOException` -- if an I/O error occurs while retrieving the label
`InfoFlowControlException` -- if caller labels don't allow reading the directories in the path
`AeolusException` -- if communication with hostname fails

---

list

```
public java.lang.String[] list()
                        throws java.io.FileNotFoundException,
                               InfoFlowControlException,
                               java.io.IOException,
                               AeolusException
```

Lists the contents of the directory as an array of file names. Caller labels must allow reading the directory.

Returns:

contents of the directory as an array of file names

Throws:

`java.io.FileNotFoundException` -- if the directory does not exist and the caller labels allow reading its parent directory
`java.io.IOException` -- if an I/O error occurred while retrieving directory labels
`InfoFlowControlException` -- if caller labels don't allow reading the directory or in the case the directory is missing, reading a higher parent directory of the given directory
`AeolusException` -- if a failure occurs while communicating with hostname

---

mkdir

```
public boolean mkdir(AeolusLabel sLabel,
                     AeolusLabel iLabel)
              throws java.io.FileNotFoundException,
                     InfoFlowControlException,
                     java.io.IOException,
                     AeolusException
```

Creates a new directory if the parent directory exists and there is no file with the given name. See reference manual for label constraints. Returns true if directory was successfully created and false if file with same name existed.

Parameters:

`sLabel` -- directory's secrecy label
`iLabel` -- directory's integrity label

Returns:

true if the directory was successfully created

Throws:

`InfoFlowControlException` -- if caller is not allowed to read and write directory's parent directory or if it is not allowed to read a directory in the path. Also the directory's labels cannot be less constraining than the tread labels. Please see the reference manual for rules concerning the root directory.
`java.io.FileNotFoundException` -- if parent directory (or some part of the path)

does not exist
`java.io.IOException` - - if an I/O error occurs while creating the directory
`AeolusException` - - if there was an error communicating with hostname