

FileConnection & PIM API

The FileConnection (FC) API - an optional package of JSR-75 - gives Java ME devices access to file systems residing on mobile devices, primarily access to removable storage media such as external memory cards. The PIM Optional Package (PIM) APIs give J2ME devices access to personal information management data native to mobile devices, such as address books, calendars, and to-do lists. It's important to note that the FC and PIM packages are independent of each other. This article will focus on the FileConnection APIs.

The FileConnection Optional Package To gain access to file systems located in a device's internal memory, or on removable memory media such as SmartMedia cards and CompactFlash cards, the FC APIs use the Generic Connection Framework (GCF) for file-system connectivity.

The FileConnection APIs The FC APIs are defined in the package `javax.microedition.io.file`, which includes two interfaces and three classes:

Interface Description FileConnection Interface for access to files or directories. FileSystemListener Listener interface for receiving status notification when adding or removing a file-system root.

Class Description FileSystemRegistry Central registry for listeners that need to add or remove a file system. ConnectionClosedException Exception thrown when a method of a file connection is invoked but cannot be completed because the connection is closed. IllegalModeException Exception thrown when a method is invoked that requires a particular security mode, such as READ or WRITE, but the connection opened is not in that mode.

Establishing Connections An application opens a connection using `Connector.open()`. The input string must comprise a fully qualified, absolute pathname of the form `file://<host>/<root>/<directory>/.../<name>`. The host element may be empty - and often will be, when the string refers to a file on the local host. The root directory corresponds to a logical mount point for a particular storage unit. Root names are device-specific. Once you've established a connection to a file system, you can perform several kinds of queries, using the FileConnection object's methods, including among others:

Get a filtered list of files and directories using the method `list(String filter, boolean includeHidden)`. In the filter parameter you can use `*` as a wild card to specify zero or more occurrences of any character. The `includeHidden` parameter specifies whether you want to list only visible files, or hidden files as well. Discover whether a file or directory exists using `exists()`. Discover whether a file or directory is hidden using `isHidden()`. Create or delete a file or directory using `create()`, `mkdir()`, or `delete()`.

For a list of all the valid root values in a device, call the `listRoots()` method of `FileSystemRegistry`.

```
public void createFile() {
    try {
        FileConnection filecon = (FileConnection)
            Connector.open("file:///SDCard/mynewfile.txt");
        // Always check whether the file or directory exists.
        // Create the file if it doesn't exist.
        if(!filecon.exists()) {
            filecon.create();
        }
        filecon.close();
    } catch(IOException ioe) {
    }
}
```

