added in API level 8
(https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

Summary: Ctors (#pubctors) | Methods (#pubmethods) | Protected Methods (#promethods) | Inherited Methods (#inhmethods) | [Expand All] (#)

FileBackupHelper

`public class FileBackupHelper`
`extends Object (https://developer.android.com/reference/java/lang/Object.html) implements BackupHelper (https://developer.android.com/reference/android/app/backup/BackupHelper.html)`

java.lang.Object (https://developer.android.com/reference/java/lang/Object.html)
↳ android.app.backup.FileBackupHelper

A helper class that can be used in conjunction with `BackupAgentHelper` `(https://developer.android.com/reference/android/app/backup/BackupAgentHelper.html)` to manage the backup of a set of files. Whenever backup is performed, all files changed since the last backup will be saved in their entirety. When backup first occurs, every file in the list provided to `FileBackupHelper(Context, String...)` `(https://developer.android.com/reference/android/app/backup/FileBackupHelper.html#FileBackupHelper(android.content.Context, java.lang.String...))` will be backed up.

During restore, if the helper encounters data for a file that was not specified when the FileBackupHelper object was constructed, that data will be ignored.

> **Note:** This should be used only with small configuration files, not large binary files.

Summary

Public constructors
`FileBackupHelper` `(https://developer.android.com/reference/android/app/backup/FileBackupHelper.html#FileBackupHelper(android.content.Context, java.lang.String...))(Context (https://developer.android.com/reference/android/content/Context.html) context, String... (https://developer.android.com/reference/java/lang/String.html) files)` Construct a helper to manage backup/restore of entire files within the application's data directory hierarchy.

Public methods	
`void`	`performBackup` `(https://developer.android.com/reference/android/app/backup/FileBackupHelper.html#performBackup(android.os.ParcelFileDescriptor, android.app.backup.BackupDataOutput, android.os.ParcelFileDescriptor))(ParcelFileDescriptor (https://developer.android.com/reference/android/os/ParcelFileDescriptor.html) oldState, BackupDataOutput (https://developer.android.com/reference/android/app/backup/BackupDataOutput.html) data, ParcelFileDescriptor (https://developer.android.com/reference/android/os/ParcelFileDescriptor.html) newState)` Based on `oldState`, determine which of the files from the application's data directory need to be backed up, write them to the data stream, and fill in `newState` with the state as it exists now.
`void`	`restoreEntity` `(https://developer.android.com/reference/android/app/backup/FileBackupHelper.html#restoreEntity(android.app.backup.BackupDataInputStream))(BackupDataInputStream (https://developer.android.com/reference/android/app/backup/BackupDataInputStream.html) data)` Restore one record [representing a single file] from the restore dataset.
`void`	`writeNewStateDescription` `(https://developer.android.com/reference/android/app/backup/FileBackupHelper.html#writeNewStateDescription(android.os.ParcelFileDescriptor))(ParcelFileDescriptor (https://developer.android.com/reference/android/os/ParcelFileDescriptor.html) fd)` Called by `BackupAgentHelper` `(https://developer.android.com/reference/android/app/backup/BackupAgentHelper.html)` after a restore operation to write the backup state file corresponding to the data as processed by the helper.

Protected methods	
`void`	`finalize` `(https://developer.android.com/reference/android/app/backup/FileBackupHelper.html#finalize())()` Called by the garbage collector on an object when garbage collection determines that there are no more references to the object.

Inherited methods
(#)From class `java.lang.Object` `(https://developer.android.com/reference/java/lang/Object.html)`

(#)From interface android.app.backup.BackupHelper (https://developer.android.com/reference/android/app/backup/BackupHelper.html)

Public constructors

FileBackupHelper

added in API level 8 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

```
FileBackupHelper (Context (https://developer.android.com/reference/android/content/Context.html) context,
                String... (https://developer.android.com/reference/java/lang/String.html) files)
```

Construct a helper to manage backup/restore of entire files within the application's data directory hierarchy.

Parameters	
context	Context: The backup agent's Context object
files	String: A list of the files to be backed up or restored.

Public methods

performBackup

added in API level 8 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

```
void performBackup (ParcelFileDescriptor (https://developer.android.com/reference/android/os/ParcelFileDescriptor.html) oldState,
                BackupDataOutput (https://developer.android.com/reference/android/app/backup/BackupDataOutput.html) data,
                ParcelFileDescriptor (https://developer.android.com/reference/android/os/ParcelFileDescriptor.html) newState)
```

Based on oldState, determine which of the files from the application's data directory need to be backed up, write them to the data stream, and fill in newState with the state as it exists now. When oldState is null, all the files will be backed up.

This should only be called directly from within the BackupAgentHelper (https://developer.android.com/reference/android/app/backup/BackupAgentHelper.html) implementation. See onBackup(ParcelFileDescriptor, BackupDataOutput, ParcelFileDescriptor) (https://developer.android.com/reference/android/app/backup/BackupAgent.html#onBackup(android.os.ParcelFileDescriptor, android.app.backup.BackupDataOutput, android.os.ParcelFileDescriptor)) for a description of parameter meanings.

Parameters	
oldState	ParcelFileDescriptor: An open, read-only ParcelFileDescriptor (https://developer.android.com/reference/android/os/ParcelFileDescriptor.html) pointing to the last backup state provided by the application. May be null, in which case no prior state is being provided and the application should perform a full backup.
data	BackupDataOutput: An open, read/write BackupDataOutput (https://developer.android.com/reference/android/app/backup/BackupDataOutput.html) pointing to the backup data destination. Typically the application will use backup helper classes to write to this file.
newState	ParcelFileDescriptor: An open, read/write ParcelFileDescriptor (https://developer.android.com/reference/android/os/ParcelFileDescriptor.html) pointing to an empty file. The application should record the final backup state here after writing the requested data to the data output stream.

restoreEntity

added in API level 8 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

```
void restoreEntity (BackupDataInputStream (https://developer.android.com/reference/android/app/backup/BackupDataInputStream.html) data)
```

Restore one record [representing a single file] from the restore dataset.

This should only be called directly from within the BackupAgentHelper (https://developer.android.com/reference/android/app/backup/BackupAgentHelper.html) implementation.

Parameters	
data	BackupDataInputStream: An open BackupDataInputStream (https://developer.android.com/reference/android/app/backup/BackupDataInputStream.html) from which the backup data can be read.

writeNewStateDescription

added in API level 8 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

```
void writeNewStateDescription (ParcelFileDescriptor (https://developer.android.com/reference/android/os/ParcelFileDescriptor.html) fd)
```

Called by BackupAgentHelper (https://developer.android.com/reference/android/app/backup/BackupAgentHelper.html) after a restore operation to write the backup state file corresponding to the data as processed by the helper. The data written here will be available to the helper during the next call to its performBackup() (https://developer.android.com/reference/android/app/backup/BackupHelper.html#performBackup(android.os.ParcelFileDescriptor, android.app.backup.BackupDataOutput, android.os.ParcelFileDescriptor)) method.

This method will be called even if the handler's restoreEntity() (https://developer.android.com/reference/android/app/backup/BackupHelper.html#restoreEntity(android.app.backup.BackupDataInputStream)) method was never invoked during the restore operation.

> **Note:** The helper should not close or seek the newState file descriptor.

Parameters	
fd	ParcelFileDescriptor: A ParcelFileDescriptor (https://developer.android.com/reference/android/os/ParcelFileDescriptor.html) to which the new state will be written.

Protected methods

finalize

added in API level 8 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

```
void finalize ()
```

Called by the garbage collector on an object when garbage collection determines that there are no more references to the object. A subclass overrides the finalize method to dispose of system resources or to perform other cleanup.

The general contract of finalize is that it is invoked if and when the Java™ virtual machine has determined that there is no longer any means by which this object can be accessed by any thread that has not yet died, except as a result of an action taken by the finalization of some other object or class which is ready to be finalized. The finalize method may take any action, including making this object available again to other threads; the usual purpose of finalize, however, is to perform cleanup actions before the object is irrevocably discarded. For example, the finalize method for an object that represents an input/output connection might perform explicit I/O transactions to break the connection before the object is permanently discarded.

The finalize method of class Object performs no special action; it simply returns normally. Subclasses of Object may override this definition.

The Java programming language does not guarantee which thread will invoke the finalize method for any given object. It is guaranteed, however, that the thread that invokes finalize will not be holding any user-visible synchronization locks when finalize is invoked. If an uncaught exception is thrown by the finalize method, the exception is ignored and finalization of that object terminates.

After the finalize method has been invoked for an object, no further action is taken until the Java virtual machine has again determined that there is no longer any means by which this object can be accessed by any thread that has not yet died, including possible actions by other objects or classes which are ready to be finalized, at which point the object may be discarded.

The finalize method is never invoked more than once by a Java virtual machine for any given object.

Any exception thrown by the finalize method causes the finalization of this object to be halted, but is otherwise ignored.

Throws	
Throwable (https://developer.android.com/reference/java/lang/Throwable.html)	