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

Summary: Constants (#constants) | Inherited Constants (#inhconstants) | Ctors (#pubctors) | Methods (#pubmethods) | Inherited Methods (#inhmethods) | [Expand All] (#)

BackupAgent

```
public abstract class BackupAgent
extends ContextWrapper
(https://developer.android.com/reference/android/content/ContextWrapper.html)
```

java.lang.Object (https://developer.android.com/reference/java/lang/Object.html)
 ↳ android.content.Context (https://developer.android.com/reference/android/content/Context.html)
  ↳ android.content.ContextWrapper (https://developer.android.com/reference/android/content/ContextWrapper.html)
   ↳ android.app.backup.BackupAgent

(#)Known Direct Subclasses

BackupAgentHelper (https://developer.android.com/reference/android/app/backup/BackupAgentHelper.html)

Provides the central interface between an application and Android's data backup infrastructure. An application that wishes to participate in the backup and restore mechanism will declare a subclass of `BackupAgent` (https://developer.android.com/reference/android/app/backup/BackupAgent.html), implement the `onBackup()` (https://developer.android.com/reference/android/app/backup/BackupAgent.html#onBackup(android.os.ParcelFileDescriptor, android.app.backup.BackupDataOutput, android.os.ParcelFileDescriptor)) and `onRestore()` (https://developer.android.com/reference/android/app/backup/BackupAgent.html#onRestore(android.app.backup.BackupDataInput, int, android.os.ParcelFileDescriptor)) methods, and provide the name of its backup agent class in its `AndroidManifest.xml` file via the `<application>` (https://developer.android.com/guide/topics/manifest/application-element.html) tag's `android:backupAgent` attribute.

Developer Guides

For more information about using BackupAgent, read the Data Backup (https://developer.android.com/guide/topics/data/backup.html) developer guide.

Basic Operation

When the application makes changes to data that it wishes to keep backed up, it should call the `BackupManager.dataChanged()` (https://developer.android.com/reference/android/app/backup/BackupManager.html#dataChanged()) method. This notifies the Android Backup Manager that the application needs an opportunity to update its backup image. The Backup Manager, in turn, schedules a backup pass to be performed at an opportune time.

Restore operations are typically performed only when applications are first installed on a device. At that time, the operating system checks to see whether there is a previously-saved data set available for the application being installed, and if so, begins an immediate restore pass to deliver the backup data as part of the installation process.

When a backup or restore pass is run, the application's process is launched (if not already running), the manifest-declared backup agent class (in the `android:backupAgent` attribute) is instantiated within that process, and the agent's `onCreate()` (https://developer.android.com/reference/android/app/backup/BackupAgent.html#onCreate()) method is invoked. This prepares the agent instance to run the actual backup or restore logic. At this point the agent's `onBackup()` (https://developer.android.com/reference/android/app/backup/BackupAgent.html#onBackup(android.os.ParcelFileDescriptor, android.app.backup.BackupDataOutput, android.os.ParcelFileDescriptor)) or `onRestore()` (https://developer.android.com/reference/android/app/backup/BackupAgent.html#onRestore(android.app.backup.BackupDataInput, int, android.os.ParcelFileDescriptor)) method will be invoked as appropriate for the operation being performed.

A backup data set consists of one or more "entities," flattened binary data records that are each identified with a key string unique within the data set. Adding a record to the active data set or updating an existing record is done by simply writing new entity data under the desired key. Deleting an entity from the data set is done by writing an entity under that key with header specifying a negative data size, and no actual entity data.

Helper Classes

An extensible agent based on convenient helper classes is available in `BackupAgentHelper` (https://developer.android.com/reference/android/app/backup/BackupAgentHelper.html). That class is particularly suited to handling of simple file or `SharedPreferences` (https://developer.android.com/reference/android/content/SharedPreferences.html) backup and restore.

Threading

The constructor, as well as `onCreate()` (https://developer.android.com/reference/android/app/backup/BackupAgent.html#onCreate()) and `onDestroy()` (https://developer.android.com/reference/android/app/backup/BackupAgent.html#onDestroy()) lifecycle callbacks run on the main thread (UI thread) of the application that implements the BackupAgent. The data-handling callbacks: `onBackup()`

(https://developer.android.com/reference/android/app/backup/BackupAgent.html#onBackup(android.os.ParcelFileDescriptor, android.app.backup.BackupDataOutput, android.os.ParcelFileDescriptor)), onFullBackup(FullBackupDataOutput) (https://developer.android.com/reference/android/app/backup/BackupAgent.html#onFullBackup(android.app.backup.FullBackupDataOutput)), onRestore() (https://developer.android.com/reference/android/app/backup/BackupAgent.html#onRestore(android.app.backup.BackupDataInput, int, android.os.ParcelFileDescriptor)), onRestoreFile() (https://developer.android.com/reference/android/app/backup/BackupAgent.html#onRestoreFile(android.os.ParcelFileDescriptor, long, java.io.File, int, long, long)), onRestoreFinished() (https://developer.android.com/reference/android/app/backup/BackupAgent.html#onRestoreFinished()), and onQuotaExceeded() (https://developer.android.com/reference/android/app/backup/BackupAgent.html#onQuotaExceeded(long, long)) run on binder pool threads.

See also:

BackupManager (https://developer.android.com/reference/android/app/backup/BackupManager.html)

BackupAgentHelper (https://developer.android.com/reference/android/app/backup/BackupAgentHelper.html)

BackupDataInput (https://developer.android.com/reference/android/app/backup/BackupDataInput.html)

BackupDataOutput (https://developer.android.com/reference/android/app/backup/BackupDataOutput.html)

Summary

Constants	
int	TYPE_DIRECTORY (https://developer.android.com/reference/android/app/backup/BackupAgent.html#TYPE_DIRECTORY) During a full restore, indicates that the file system object being restored is a directory.
int	TYPE_FILE (https://developer.android.com/reference/android/app/backup/BackupAgent.html#TYPE_FILE) During a full restore, indicates that the file system object being restored is an ordinary file.

Inherited constants
(#)From class android.content.Context (https://developer.android.com/reference/android/content/Context.html)

Public constructors
BackupAgent (https://developer.android.com/reference/android/app/backup/BackupAgent.html#BackupAgent())()

Public methods	
final void	fullBackupFile (https://developer.android.com/reference/android/app/backup/BackupAgent.html#fullBackupFile(java.io.File, android.app.backup.FullBackupDataOutput))(File (https://developer.android.com/reference/java/io/File.html) file, FullBackupDataOutput (https://developer.android.com/reference/android/app/backup/FullBackupDataOutput.html) output) Write an entire file as part of a full-backup operation.
abstract void	onBackup (https://developer.android.com/reference/android/app/backup/BackupAgent.html#onBackup(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) The application is being asked to write any data changed since the last time it performed a backup operation.
void	onCreate (https://developer.android.com/reference/android/app/backup/BackupAgent.html#onCreate())() Provided as a convenience for agent implementations that need an opportunity to do one-time initialization before the actual backup or restore operation is begun.
void	onDestroy (https://developer.android.com/reference/android/app/backup/BackupAgent.html#onDestroy())() Provided as a convenience for agent implementations that need to do some sort of shutdown process after backup or restore is completed.
void	onFullBackup (https://developer.android.com/reference/android/app/backup/BackupAgent.html#onFullBackup(android.app.backup.FullBackupDataOutput))(FullBackupDataOutput (https://developer.android.com/reference/android/app/backup/FullBackupDataOutput.html) data) The application is having its entire file system contents backed up.
void	onQuotaExceeded (https://developer.android.com/reference/android/app/backup/BackupAgent.html#onQuotaExceeded(long, long))(long backupDataBytes,

	long quotaBytes) Notification that the application's current backup operation causes it to exceed the maximum size permitted by the transport.
abstract void	onRestore (https://developer.android.com/reference/android/app/backup/BackupAgent.html#onRestore(android.app.backup.BackupDataInput, int, android.os.ParcelFileDescriptor))(BackupDataInput (https://developer.android.com/reference/android/app/backup/BackupDataInput.html) data, int appVersionCode, ParcelFileDescriptor (https://developer.android.com/reference/android/os/ParcelFileDescriptor.html) newState) The application is being restored from backup and should replace any existing data with the contents of the backup.
void	onRestoreFile (https://developer.android.com/reference/android/app/backup/BackupAgent.html#onRestoreFile(android.os.ParcelFileDescriptor, long, java.io.File, int, long, long))(ParcelFileDescriptor (https://developer.android.com/reference/android/os/ParcelFileDescriptor.html) data, long size, File (https://developer.android.com/reference/java/io/File.html) destination, int type, long mode, long mtime) Handle the data delivered via the given file descriptor during a full restore operation.
void	onRestoreFinished (https://developer.android.com/reference/android/app/backup/BackupAgent.html#onRestoreFinished())() The application's restore operation has completed.

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

Constants

TYPE_DIRECTORY

added in API level 14 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

```
int TYPE_DIRECTORY
```

During a full restore, indicates that the file system object being restored is a directory.

Constant Value: 2 (0x00000002)

TYPE_FILE

added in API level 14 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

```
int TYPE_FILE
```

During a full restore, indicates that the file system object being restored is an ordinary file.

Constant Value: 1 (0x00000001)

Public constructors

BackupAgent

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

```
BackupAgent ()
```

Public methods

fullBackupFile

added in API level 14 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

```
void fullBackupFile (File (https://developer.android.com/reference/java/io/File.html) file,
                FullBackupDataOutput (https://developer.android.com/reference/android/app/backup/FullBackupDataOutput.html) output)
```

Write an entire file as part of a full-backup operation. The file's contents will be delivered to the backup destination along with the metadata necessary to place it with the proper location and permissions on the device where the data is restored.

> Attempting to back up files in directories that are ignored by the backup system will have no effect. For example, if the app calls this method with a file inside the `getNoBackupFilesDir()` (https://developer.android.com/reference/android/content/ContextWrapper.html#getNoBackupFilesDir()) directory, it will be ignored. See `onFullBackup(FullBackupDataOutput)` (https://developer.android.com/reference/android/app/backup/BackupAgent.html#onFullBackup(android.app.backup.FullBackupDataOutput)) for details on what directories are excluded from backups.

Parameters	
`file`	`File`: The file to be backed up. The file must exist and be readable by the caller.
`output`	`FullBackupDataOutput`: The destination to which the backed-up file data will be sent.

onBackup

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

```
void onBackup (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)
```

The application is being asked to write any data changed since the last time it performed a backup operation. The state data recorded during the last backup pass is provided in the `oldState` file descriptor. If `oldState` is `null`, no old state is available and the application should perform a full backup. In both cases, a representation of the final backup state after this pass should be written to the file pointed to by the file descriptor wrapped in `newState`.

Each entity written to the `BackupDataOutput` (https://developer.android.com/reference/android/app/backup/BackupDataOutput.html) `data` stream will be transmitted over the current backup transport and stored in the remote data set under the key supplied as part of the entity. Writing an entity with a negative data size instructs the transport to delete whatever entity currently exists under that key from the remote data set.

Parameters	
`oldState`	`ParcelFileDescriptor`: An open, read-only ParcelFileDescriptor 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`: A structured wrapper around an open, read/write file descriptor 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 pointing to an empty file. The application should record the final backup state here after writing the requested data to the `data` output stream.

Throws	
`IOException` (https://developer.android.com/reference/java/io/IOException.html)	

onCreate

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

```
void onCreate ()
```

Provided as a convenience for agent implementations that need an opportunity to do one-time initialization before the actual backup or restore operation is begun.

onDestroy

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

```
void onDestroy ()
```

Provided as a convenience for agent implementations that need to do some sort of shutdown process after backup or restore is completed.

Agents do not need to override this method.

onFullBackup

added in API level 14 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

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

The application is having its entire file system contents backed up. `data` points to the backup destination, and the app has the opportunity to choose which files are to be stored. To commit a file as part of the backup, call the `fullBackupFile(File, FullBackupDataOutput)` `(https://developer.android.com/reference/android/app/backup/BackupAgent.html#fullBackupFile(java.io.File, android.app.backup.FullBackupDataOutput))` helper method. After all file data is written to the output, the agent returns from this method and the backup operation concludes.

Certain parts of the app's data are never backed up even if the app explicitly sends them to the output:

- The contents of the `getCacheDir()` `(https://developer.android.com/reference/android/content/ContextWrapper.html#getCacheDir())` directory
- The contents of the `getCodeCacheDir()` `(https://developer.android.com/reference/android/content/ContextWrapper.html#getCodeCacheDir())` directory
- The contents of the `getNoBackupFilesDir()` `(https://developer.android.com/reference/android/content/ContextWrapper.html#getNoBackupFilesDir())` directory
- The contents of the app's shared library directory

The default implementation of this method backs up the entirety of the application's "owned" file system trees to the output other than the few exceptions listed above. Apps only need to override this method if they need to impose special limitations on which files are being stored beyond the control that `getNoBackupFilesDir()` `(https://developer.android.com/reference/android/content/ContextWrapper.html#getNoBackupFilesDir())` offers. Alternatively they can provide an xml resource to specify what data to include or exclude.

Parameters	
`data`	`FullBackupDataOutput`: A structured wrapper pointing to the backup destination.

Throws	
	IOException
`IOException` `(https://developer.android.com/reference/java/io/IOException.html)`	

See also:

`getNoBackupFilesDir()` `(https://developer.android.com/reference/android/content/Context.html#getNoBackupFilesDir())`

`ERROR(/ApplicationInfo#fullBackupContent)` `(https://developer.android.com/)`

`fullBackupFile(File, FullBackupDataOutput)` `(https://developer.android.com/reference/android/app/backup/BackupAgent.html#fullBackupFile(java.io.File, android.app.backup.FullBackupDataOutput))`

`onRestoreFile(ParcelFileDescriptor, long, File, int, long, long)` `(https://developer.android.com/reference/android/app/backup/BackupAgent.html#onRestoreFile(android.os.ParcelFileDescriptor, long, java.io.File, int, long, long))`

onQuotaExceeded

added in API level 24 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

```
void onQuotaExceeded (long backupDataBytes, 
                long quotaBytes)
```

Notification that the application's current backup operation causes it to exceed the maximum size permitted by the transport. The ongoing backup operation is halted and rolled back: any data that had been stored by a previous backup operation is still intact. Typically the quota-exceeded state will be detected before any data is actually transmitted over the network.

The `quotaBytes` value is the total data size currently permitted for this application. If desired, the application can use this as a hint for determining how much data to store. For example, a messaging application might choose to store only the newest messages, dropping enough older content to stay under the quota.

> Note that the maximum quota for the application can change over time. In particular, in the future the quota may grow. Applications that adapt to the quota when deciding what data to store should be aware of this and implement their data storage mechanisms in a way that can take advantage of additional quota.

Parameters	
`backupDataBytes`	`long`: The amount of data measured while initializing the backup operation, if the total exceeds the app's alloted quota. If initial measurement suggested that the data would fit but then too much data was actually submitted as part of the operation, then this value is the amount of data that had been streamed into the transport at the time the quota was reached.
`quotaBytes`	`long`: The maximum data size that the transport currently permits this application to store as a backup.

onRestore

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

```
void onRestore (BackupDataInput (https://developer.android.com/reference/android/app/backup/BackupDataInput.html) data,
                int appVersionCode,
                ParcelFileDescriptor (https://developer.android.com/reference/android/os/ParcelFileDescriptor.html) newState)
```

The application is being restored from backup and should replace any existing data with the contents of the backup. The backup data is provided through the `data` parameter. Once the restore is finished, the application should write a representation of the final state to the `newState` file descriptor.

The application is responsible for properly erasing its old data and replacing it with the data supplied to this method. No "clear user data" operation will be performed automatically by the operating system. The exception to this is in the case of a failed restore attempt: if onRestore() throws an exception, the OS will assume that the application's data may now be in an incoherent state, and will clear it before proceeding.

Parameters	
`data`	`BackupDataInput`: A structured wrapper around an open, read-only file descriptor pointing to a full snapshot of the application's data. The application should consume every entity represented in this data stream.
`appVersionCode`	`int`: The value of the `android:versionCode` (https://developer.android.com/guide/topics/manifest/manifest-element.html#vcode) manifest attribute, from the application that backed up this particular data set. This makes it possible for an application's agent to distinguish among any possible older data versions when asked to perform the restore operation.
`newState`	`ParcelFileDescriptor`: An open, read/write ParcelFileDescriptor pointing to an empty file. The application should record the final backup state here after restoring its data from the `data` stream. When a full-backup dataset is being restored, this will be `null`.

Throws	

`IOException`
`(https://developer.android.com/reference/java/io/IOException.html)`	

onRestoreFile

added in API level 14 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

```
void onRestoreFile (ParcelFileDescriptor (https://developer.android.com/reference/android/os/ParcelFileDescriptor.html) data,
                long size,
                File (https://developer.android.com/reference/java/io/File.html) destination,
                int type,
                long mode,
                long mtime)
```

Handle the data delivered via the given file descriptor during a full restore operation. The agent is given the path to the file's original location as well as its size and metadata.

The file descriptor can only be read for `size` bytes; attempting to read more data has undefined behavior.

The default implementation creates the destination file/directory and populates it with the data from the file descriptor, then sets the file's access mode and modification time to match the restore arguments.

Parameters	
`data`	`ParcelFileDescriptor`: A read-only file descriptor from which the agent can read `size` bytes of file data.

size	long: The number of bytes of file content to be restored to the given destination. If the file system object being restored is a directory, size will be zero.
destination	File: The File on disk to be restored with the given data.
type	int: The kind of file system object being restored. This will be either TYPE_FILE (https://developer.android.com/reference/android/app/backup/BackupAgent.html#TYPE_FILE) or TYPE_DIRECTORY (https://developer.android.com/reference/android/app/backup/BackupAgent.html#TYPE_DIRECTORY).
mode	long: The access mode to be assigned to the destination after its data is written. This is in the standard format used by chmod().
mtime	long: The modification time of the file when it was backed up, suitable to be assigned to the file after its data is written.

Throws	
IOException (https://developer.android.com/reference/java/io/IOException.html)	

onRestoreFinished

added in API level 21 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

```
void onRestoreFinished ()
```

The application's restore operation has completed. This method is called after all available data has been delivered to the application for restore (via either the onRestore() (https://developer.android.com/reference/android/app/backup/BackupAgent.html#onRestore(android.app.backup.BackupDataInput, int, android.os.ParcelFileDescriptor)) or onRestoreFile() (https://developer.android.com/reference/android/app/backup/BackupAgent.html#onRestoreFile(android.os.ParcelFileDescriptor, long, java.io.File, int, long, long)) callbacks). This provides the app with a stable end-of-restore opportunity to perform any appropriate post-processing on the data that was just delivered.

See also:

onRestore(BackupDataInput, int, ParcelFileDescriptor) (https://developer.android.com/reference/android/app/backup/BackupAgent.html#onRestore(android.app.backup.BackupDataInput, int, android.os.ParcelFileDescriptor))

onRestoreFile(ParcelFileDescriptor, long, File, int, long, long) (https://developer.android.com/reference/android/app/backup/BackupAgent.html#onRestoreFile(android.os.ParcelFileDescriptor, long, java.io.File, int, long, long))