SharedPreferencesBackupHelper

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] (#)

`public class SharedPreferencesBackupHelper`
`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.SharedPreferencesBackupHelper

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 `SharedPreferences` `(https://developer.android.com/reference/android/content/SharedPreferences.html)`. Whenever a backup is performed, it will back up all named shared preferences that have changed since the last backup operation.

To use this class, the application's backup agent class should extend `BackupAgentHelper` `(https://developer.android.com/reference/android/app/backup/BackupAgentHelper.html)`. Then, in the agent's `onCreate()` `(https://developer.android.com/reference/android/app/backup/BackupAgent.html#onCreate())` method, an instance of this class should be allocated and installed as a backup/restore handler within the BackupAgentHelper framework. For example, an agent supporting backup and restore for an application with two groups of `SharedPreferences` `(https://developer.android.com/reference/android/content/SharedPreferences.html)` data might look something like this:

```
import android.app.backup.BackupAgentHelper;
import android.app.backup.SharedPreferencesBackupHelper;

public class MyBackupAgent extends BackupAgentHelper {
    // The names of the SharedPreferences groups that the application maintains.  These
    // are the same strings that are passed to getSharedPreferences(String, int) (https://developer.android.com/reference/andro
    static final String PREFS_DISPLAY = "displayprefs";
    static final String PREFS_SCORES = "highscores";

    // An arbitrary string used within the BackupAgentHelper implementation to
    // identify the SharedPreferenceBackupHelper's data.
    static final String MY_PREFS_BACKUP_KEY = "myprefs";

    // Simply allocate a helper and install it
    void onCreate() {
        SharedPreferencesBackupHelper helper =
                new SharedPreferencesBackupHelper(this, PREFS_DISPLAY, PREFS_SCORES);
        addHelper(MY_PREFS_BACKUP_KEY, helper);
    }
}
```

No further implementation is needed; the `BackupAgentHelper` `(https://developer.android.com/reference/android/app/backup/BackupAgentHelper.html)` mechanism automatically dispatches the `BackupAgent.onBackup()` `(https://developer.android.com/reference/android/app/backup/BackupAgent.html#onBackup(android.os.ParcelFileDescriptor, android.app.backup.BackupDataOutput, android.os.ParcelFileDescriptor))` and `BackupAgent.onRestore()` `(https://developer.android.com/reference/android/app/backup/BackupAgent.html#onRestore(android.app.backup.BackupDataInput, int, android.os.ParcelFileDescriptor))` callbacks to the SharedPreferencesBackupHelper as appropriate.

Summary

Public constructors
`SharedPreferencesBackupHelper` `(https://developer.android.com/reference/android/app/backup/SharedPreferencesBackupHelper.html#SharedPreferencesBackupHelper(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)` `prefGroups)` Construct a helper for backing up and restoring the `SharedPreferences` `(https://developer.android.com/reference/android/content/SharedPreferences.html)` under the given names.

Public methods	
void	performBackup (https://developer.android.com/reference/android/app/backup/SharedPreferencesBackupHelper.html#performBackup(android.os.ParcelFileDescriptor, android.app.backup.BackupDataOutput, android.os.ParcelFileDescriptor))(ParcelFileDescriptor (https://developer.android.com/reference/android/os/ParcelFileDescript oldState, BackupDataOutput (https://developer.android.com/reference/android/app/backup/BackupDataOutput.html) data, ParcelFileDescriptor (https://developer.android.com/reference/android/os/ParcelFileDescriptor.html) newState) Backs up the configured SharedPreferences (https://developer.android.com/reference/android/content/SharedPreferences.html) groups.
void	restoreEntity (https://developer.android.com/reference/android/app/backup/SharedPreferencesBackupHelper.html#restoreEntity(android.app.backup.BackupDataInputStream) (BackupDataInputStream (https://developer.android.com/reference/android/app/backup/BackupDataInputStream.html) data) Restores one entity from the restore data stream to its proper shared preferences file store.
void	writeNewStateDescription (https://developer.android.com/reference/android/app/backup/SharedPreferencesBackupHelper.html#writeNewStateDescription(android.os.ParcelFileDescriptor))(ParcelFile (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 fil corresponding to the data as processed by the helper.

Protected methods	
void	finalize (https://developer.android.com/reference/android/app/backup/SharedPreferencesBackupHelper.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

SharedPreferencesBackupHelper

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

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

Construct a helper for backing up and restoring the SharedPreferences (https://developer.android.com/reference/android/content/SharedPreferences.html) under the given names.

Parameters	
context	Context: The application Context (https://developer.android.com/reference/android/content/Context.html)
prefGroups	String: The names of each SharedPreferences (https://developer.android.com/reference/android/content/SharedPreferences.html) file to back up

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)
```

Backs up the configured SharedPreferences (https://developer.android.com/reference/android/content/SharedPreferences.html) groups.

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)`

Restores one entity from the restore data stream to its proper shared preferences file store.

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)`	