

roRegistry

The Registry is an area of non-volatile storage where a small number of persistent settings can be stored.

Supported Interfaces

- [ifRegistry](#)

Description

The Registry provides a means for an application to write and read small amounts of data such as settings, scores, etc. The data persists even if the user exits the application and even if the player is rebooted. Registry data is removed only when the application explicitly removes it, the user uninstalls the application, which remove the registry for the application, or the user performs a factory reset, which removes the registry for all applications.

There is a separate registry for each [developer ID](#). This allows multiple applications to use the registry without being able to read or modify the registry from other applications. If desired, a single registry can be shared across multiple applications by using the same developer ID to package the applications. This is the conventional way that an "application suite" with shared preferences and other shared information should work. Each registry is divided into sections which are specified by the developer for organization and grouping of attributes. Methods in [ifRegistry](#) are provided to list the sections in the registry and to provide access to the data in each section. The maximum size of each application registry is 16K bytes. Care should be taken to minimize the amount of data stored and the frequency it is updated.

The Registry also supports the use of a special *transient* registry section. A registry section named "Transient" can be used to store attributes that have the lifetime of a single boot. Within a specific boot session, these values will be persistent to the application and stored as any other registry value. Whenever the user reboots the Roku Streaming Player, all "Transient" registry sections are removed and the values no longer persist. This technique is useful for caching data to minimize network access, yet still ensuring that this data is always fresh after a system reboot.

The registry is encrypted, and updates are relatively performance intensive and should be used sparingly. Note that all writes to the registry are delayed, and not committed to non-volatile storage until [ifRegistry.Flush\(\)](#) or [ifRegistrySection.Flush\(\)](#) is explicitly called. The platform may choose opportune times to flush data on its own, but no application is technically correct unless it explicitly calls [Flush\(\)](#) at appropriate times. Flushing the registry is a relatively time-consuming operation, so it should be done as infrequently as possible. The Registry data is stored in a fault tolerant manner by preserving a backup for each write which is automatically rolled-back in the event of a failure.

Access to the registry is available through the [roRegistry](#) object. This object is created with no parameters.

```
CreateObject("roRegistry")
```