

# PersistentObject

---

This article explains how to create and use simple persistent objects in QML

## Introduction

---

Apps often need to store (persist) low-volume data between sessions. This can be account names, user settings, recently viewed items, etc. QML in Qt Quick has no direct support for this. However, combining Qt Quick's Offline Storage API and QML, this feature can be added very elegantly.

## Usage

---

Include `PersistentObject.qml` and `PersistentObjectStore.js` to your QML directory (see "Code Example" on the right for these files). Now you can declare a `PersistentObject` object everywhere you want. The properties you declare in this object are automatically loaded and saved each time the application is started or exited.

Example:

```
import QtQuick 1.0

Rectangle {
    width: 300
    height: 300

    PersistentObject {
        id: simpleObject
        property string someText: "initial text"
    }

    TextInput {
        text: simpleObject.someText
        onTextChanged: simpleObject.someText = text
    }
}
```

When you start this application for the first time, the text "initial text" is shown in an editable input field. When you alter this text, quit the app and restart it again, the altered text is shown instead. Data is stored on a per-application basis. For the Nokia N9, the storage directory is `/home/user/.local/share/data/QML/OfflineStorage/Databases`.

More demos and sources can be downloaded from the [Persistentobject.zip](#). For a real live example take a look at "main.qml" of the [Nokia N9 buienradar app](#).

## Limitations and caveats

---

- As this method was designed and tested to store nothing more than some basic application settings, it's probably best to limit the number and complexity of the properties you want to store.
- Data is stored unencrypted and can be read by other apps.
- If you want to use more than one `PersistentObject` in your application, don't forget to provide an unique `objectName` for each `PersistentObject` (see `demo4.qml`). The default `objectName` is "default".

## Alternatives

---

Some alternative solutions for storing data in Qt Quick apps worth investigating:

- [Using QSettings in QML with also json and XML support](#)
- [How-to\\_create\\_a\\_persistent\\_settings\\_database\\_in\\_Qt\\_Quick\\_\(QML\)](#)

## Summary

---

PersistentObjects are simple pure QML objects just like QObject. However, the property values of a PersistentObject are automatically saved between sessions. Useful for storing user settings in QML applications. Uses the QML Offline Storage API.