Delphi Worlds

Delphi enters new worlds... Blog Resources › About Me search

Checking for an internet connection on mobile devices with Delphi XE5

UPDATE: The project has been updated to include retrieving the current SSID (that the device is connected to). For Android, this means adding the Access Wifi State permission. Refer to the updated text in the article.

It's a question that has been asked a few times in the Embarcadero forums: how to check if a mobile device (either Android or iOS) has an internet connection.

I already knew of a solution for iOS, and recently I came across one for Android, so I figured I would combine the two and create a platform-independent unit, called NetworkState. In that unit, I created an abstract class that could be derived from in other units that would provide the implementation for each platform, i.e. Android and iOS. The declaration for the abstract class TCustomNetworkState looks like this:

```
TCustomNetworkState = class(TObject)
  function GetSSID: String; virtual; abstract;
  function IsConnected: Boolean; virtual; abstract;
  function IsWifiConnected: Boolean; virtual; abstract;
  function IsMobileConnected: Boolean; virtual; abstract;
end;
```

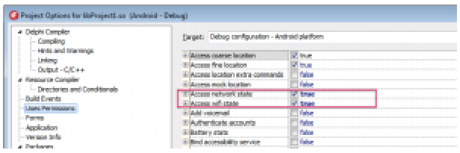
As can be seen, the intention is to simply provide functions for checking if either there is a connection at all, and specifically whether there is either a Wifi or Mobile (Cellular data) connection, and what the SSID is. It was also the intention that the implementation use the supported methods for checking for connections, rather than say, using the Indy components.

The units NetworkState.iOS and NetworkState.Android each declare TPlatformNetworkState that provide the actual implementation, and the implementation uses clause of NetworkState is appropriately IFDEF'd for each platform so that the actual instance that is created is the one for the specific platform being compiled for.

I've included a demo project below that includes all the necessary files. If you're going to use TNetworkState in your own project, make sure the files:

libReachability.a
NetworkState.pas
NetworkState.iOS.pas
NetworkState.Android.pas
SCNetworkReachability.pas
CaptiveNetwork.pas

are either in your project directory or in the compiler search path. Bear in mind that **for the Android platform** you'll need to include the Access Network State, and Access Wifi State permissions in the project options:



Target: Debug configuration - Android platform

Access coarse location	☑ true
Access fine location	☑ true
Access location extra commands	□ false
Access mock location	□ false
Access network state	☑ true
Access wifi state	☑ true
Add voicemail	□ false
Authenticate accounts	□ false
Battery stats	□ false
Bind accessibility service	□ false

For **iOS on the device**, you'll need to add the SystemConfiguration framework in the SDK manager. Please refer to this article, and instead of CoreBluetooth (as per the 2nd image), use SystemConfiguration.

Network State Demo

Delphi Worlds

Curtir 4

categories

- Code tips
- IDE tweaks
- Patches
- Product highlight
- Project tweaks
- Uncategorized
- Updates news

blog roll

- Brian Long
- Delphi Haven
- The Podcast At Delphi.Org
- Nick Hodges
- Jeroen Pluimers
- Chee Wee Chua

related sites

- Android Devices supported
- Embarcadero Android Devices

RSS - Posts

RSS - Comments

recent posts

- Delphi XE5 Update 2
- Product highlight: e-Restaurant Touch
- Checking for an internet connection on mobile devices with Delphi XE5
- Adding Javascript processing