How should I interpret the various values of NLM_CONNECTIVITY?

devblogs.microsoft.com/oldnewthing/20230112-00

January 12, 2023

Raymond Chen

The `NLM_CONNECTIVITY` flags enumeration describes what types of network connectivity are available, as far as the networking infrastructure can tell.

Mind you, network connectivity is a hazy concept, because whether a particular endpoint can be reached is dependent not only on the configuration of the local system, but also on the configuration of every machine between you and the endpoint, and those configurations can change at any time. A hunter accidentally damages a power line and suddenly you lose connection to a server. The system won't know about this until you try to contact that server.

Even for the state of the local system, it takes time for the system to re-evaluate the network connectivity after a change in configuration (such as an unplugged network cable), so you have to accept that the values you receive are based on the most recent information available, but that information may be in flux. And of course there are the shenanigans noted above.

The networking folks gave me this breakdown of what the flags mean and how apps should deal with them.

Flag	Meaning	Recommendation
`DISCONNECTED`	No network interface detects any network.	Treat as offline.
`NOTRAFFIC`	An interface is connected, but it cannot send or receive network traffic.	
`SUBNET`	An interface has been configured to send traffic, but the system cannot confirm Internet connectivity.	Make one attempt to contact service.
`LOCALNETWORK`		
`INTERNET`	The system has confirmed access to Microsoft Internet sites.	Treat as fully online.

In the case of `SUBNET` or `LOCALNETWORK`, you can make one attempt to contact your Internet service even though Windows doesn't think it's going to work. This deals with the case where people employ shenanigans to prevent Windows from detecting Internet connectivity, such as blocking access to the `msftconnecttest.com` site, or to all Microsoft-owned IP addresses.

The `IsConnected` property considers your network to be connected if it is connected to a `LOCALNETWORK` or `INTERNET`. The `IsConnectedToInternet` property requires `INTERNET`.

If the system is trapped behind a captive portal, it will report itself as `LOCALNETWORK`. To identify the captive portal case specifically, call `INetworkListManager::GetNetworks` and use the `IEnumNetworks` to enumerate all the `INetwork` objects. Query each `INetwork` for `IPropertyBag` and check the `NA_InternetConnectivityV4` and `NA_Internet-ConnectivityV6` properties. If either one has the `NLM_INTERNET_CONNECTIVITY_WEB-HIJACK` flag set, then you are trapped in a captive portal.

Another way to check whether you're stuck in a captive portal is to call `Connection-Profile.GetNetworkConnectivityLevel` and check for `ConstrainedInternetAccess`, which is the name `NetworkConnectivityLevel` gives to being stuck in a captive portal.

(I'm sorry it's so complicated, but networking is complicated.)

Raymond Chen

Follow