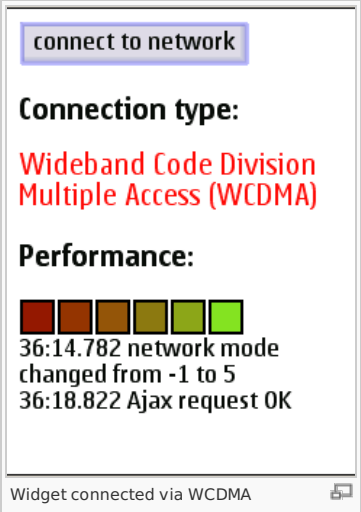
Network connection type detection in Symbian Web Runtime

Overview

Network connection* type detection is useful for widgets that make use of multimedia features such as big pictures. By detecting the network connection type, the widget can request content that is tailored for the available bandwidth from the server. For example, a smaller or more heavily compressed image could be returned if a 2G connection is used.

**Network connection refers to the connection that the widget is using. It is not always the same connection the phone uses. For example, a widget can use a WLAN connection in offline mode.*



Solution

Network connection type can be obtained via the SystemInfo Service API (WRT 1.1). However, there is no single API that would provide a satisfactory end result. Two APIs must be used together: `Network` and `Connectivity`.

Network monitoring must be done in three steps:

1) First start monitoring the connection type with `getNotification`, using the criteria *Connectivity ConnectionStatus*. If your widget, some other widget, or the browser initiates a network connection, then this callback is called. It is also called if the phone is set to offline mode.

```
var serviceObj = device.getServiceObject("Service.SysInfo", "ISysInfo");
var connectionStatusCriteria = {
   Entity : "Connectivity",
   Key : "ConnectionStatus"
}
serviceObj.ISysInfo.GetNotification(connectionStatusCriteria, connectionStatusChanged);
```

2) Next, check the initial connection state with `getInfo` criteria *Connectivity ConnectionStatus*. In case there are no ongoing active connections (the widget will prompt IAP selection when accessing the network), the returned `ConnectionList` has a length of zero. Use `ConnectionList.getNext();` to iterate over `NetworkInfo` objects.

```
serviceObj.ISysInfo.GetInfo(connectionStatusCriteria, connectionStatusChanged);

function connectionStatusChanged(transId, eventCode, result){
    var nwMode = document.getElementById("nwMode");
    if(result.ErrorCode == 0){
        var connInfo;

        //if wlan connection is made in offline mode, we get just conninfo object and not the list
        if(result.ReturnValue.ConnectionList){
     connInfo =  result.ReturnValue.ConnectionList.getNext();
 }else{
            connInfo = result.ReturnValue;
 }

        if(connInfo != undefined ){
     switch(connInfo.ConnectionType) {
  case -1: nwMode.innerHTML ="Unknown";  break;

  case 0: nwMode.innerHTML="Circuit Switched Data (CSD)";
   rate(0);
   break; // 9.6 kbit/s

  case 1: nwMode.innerHTML="Wideband Code Division Multiple Access (WCDMA)";
   rate(5);
   break; //384 Kbps to 2 Mbps

  case 2: nwMode.innerHTML="Local Area Network (LAN) [Emulator]";
   rate(5);
   break;

  case 3: nwMode.innerHTML="Code Division Multiple Access (CDMA2000)";
   rate(3);
   break; //153 Kbps

  case 4: nwMode.innerHTML="General Packet Radio Service (GPRS)";
   rate(2);
   break; //56-114 kbit/s.

  case 5: nwMode.innerHTML="High Speed Circuit Switched Data (HSCSD)";
   rate(1);
   break; //up to 38.4 kbit/s

  case 6: nwMode.innerHTML="Enhanced Data rates for Global Evolution (EDGE), Enhanced GPRS (EGPRS)";
   rate(4);
   break; //236.8 kbit/s

  case 7: nwMode.innerHTML="Wireless Local Area Network (WLAN)";
   rate(5);
   break; //fast
```


(C) Copyright Nokia 2013. All rights reserved.

```
  case 8: nwMode.innerHTML="Bluetooth";  break;

  case 9: nwMode.innerHTML="Virtual VPN";  break;

     }
            if(connInfo.ConnectionStatus == 0){
         nwMode.innerHTML += " (disconnected)";
               rate(-1);
            }
 }else{
     nwMode.innerHTML="Widget is not yet connected to net";
 }
    }else{
        writelog("Error connectionStatusChanged: "+ result.ErrorCode);
    }
}
```

3) This step is a workaround for `getNotification` *Connectivity ConnectionStatus*. It does not fire if the access point stays the same but `ConnectionType` changes, for example from 3G to 2G. Therefore, one must listen to `CurrentNetwork` as well. Once the callback `networkStatusChanged` is called, check with `getInfo` *Connectivity ConnectionStatus* if it affects the widget connection.

```
var networkCriteria = {
    Entity : "Network",
    Key : "CurrentNetwork"
}
serviceObj.ISysInfo.GetNotification(networkCriteria, networkStatusChanged);

var oldNetworkMode = -1;
function networkStatusChanged(transId, eventCode, result){
    if(result.ErrorCode == 0){
        //only if network mode has changed
 if(result.ReturnValue.NetworkMode != oldNetworkMode){
     writelog("network mode changed from "+ oldNetworkMode+ " to " + result.ReturnValue.NetworkMode);
     oldNetworkMode = result.ReturnValue.NetworkMode;
     getInfo(connectionStatusCriteria, connectionStatusChanged);
 }
    }
}
```

For a complete example, see: ConType.zip

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.