

# How to detect disconnected network?

Published 2015-01-15 | (Compatible with SDK 2.5,3.5,4.5,5.0,5.1 and 2011,2012,2013,2014 models)

Lead

Implementation tips for network disconnected event

The Smart TV policy requires that all network errors which disable application functions should be populated with proper message. This includes errors occurred while downloading some content or during Audio/Video playback. It is strongly recommended that not to rely on the OnNetworkDisconnected player event only, but rather to check periodically all available network interfaces.

Below, there is a sample code snippet on how to implement that.

Note

This example just guarantee till gateway connection, do not till internet network. If app should guarantee connection of internet network, app should communicate with app server whether TV connected on internet network or not.

1. Add the plugin network object in index.html:

```
<object id="pluginObjectNetwork" border="0" classid="clsid:SAMSUNG-INFOLINK-NETWORK"></object>
```

2. Set the interval for network connection check.

Too frequent repeating may result in slowing down the application, too rare may not discover the network error before the playback is hanged. The recommended values are between 500 and 2000 milliseconds.

```
var networkPlugin = document.getElementById('pluginObjectNetwork');  
var internetConnectionInterval = 500; //this is just an example value that repeats the connection check every 500 milli  
seconds
```

3. Add a handler function for showing and hiding the network disconnection popup.

Note that the player should be stopped as soon as the network error is discovered to prevent potential errors.

```
function cyclicInternetConnectionCheck() {  
  
    if(!checkConnection()){  
  
        // no internet connection  
  
        if(!internetConnectionPopup.isShown){  
  
            internetConnectionPopup.show();  
  
            Player.stop();  
  
        }  
  
    } else {  
  
        // if error message was shown, it should be returned back to normal  
  
        if(internetConnectionPopup.isShown){  
  
            internetConnectionPopup.hide();  
  
            // return to normal screen  
  
        }  
  
    }  
  
    }  
  
    setInterval("cyclicInternetConnectionCheck",internetConnectionInterval);  
}
```

4. Check all available network interfaces:

```
function checkConnection() {  
  
    var gatewayStatus = 0,  
  
    // Get active connection type - wired or wireless.  
  
    currentInterface =networkPlugin.GetActiveType();  
  
    // If no active connection.  
  
    if (currentInterface === -1) {  
  
        return false;  
  
    }  
  
    // Check Gateway connection of current interface.  
  
    gatewayStatus = networkPlugin.CheckGateway(currentInterface);  
  
    // If not connected or error.  
  
    if (gatewayStatus !== 1) {  
  
        return false;  
  
    }  
  
    // Everything went OK.  
  
    return true;  
  
}
```

#### Note

The checkConnection function needs to verify gateway of wired or wireless connection on the basis of current active type.