# Network Signal JavaScript component for WRT

This article explains **how to create and use a JavaScript Network signal component** in a **Web Runtime widget**. This example uses WRT 1.0 SystemInfo Service API, and is so **compatible with both WRT 1.0 and 1.1 versions**.

## NetworkSignal component: how to use it

The following steps are necessary in order to use the Network Signal component in a WRT Widget:

- **Import the networksignal_component.js** JavaScript library (source code available here: Media:Networksignal component.txt).

```
<script language="javascript" type="text/javascript"
src="networksignal_component.js"></script>
```

- **Define a parent element** within your WRT HTML code, that will contain the component itself:

```
<body>
[...]
 <div id="signal_holder"></div>
[...]
</body>
```

- **Embed the WRT 1.0 SysInfo Service API plugin HTML code** snippet in your widget HTML code:

```
<embed type="application/x-systeminfo-widget" hidden="yes"></embed>
```

- Add to your WRT project 2 images of the same size, one representing zero network signal, the other one representing the full network signal. Since the network signal values can go from 0 to 7, it's good practice to create signal images with these 7 values clearly visible. Sample images used in this article are the following:

- **Initialize a NetworkSignal instance**, passing these arguments:
  - the zero signal image path
  - the full signal image path
  - the images width


(C) Copyright Nokia 2014. All rights reserved.

- the images height

Once instantiated, append the NetworkSignal instance to the parent element defined above:

```
var signal = new NetworkSignal('images/signal_empty.png', 'images/signal_full.png', 72,
168);

signal.appendTo(document.getElementById('signal_holder'));
```

# Source code: the NetworkSignal component implementation

## NetworkSignal constructor and DOM structure

Let's start implementing the NetworkSignal constructor, and a first method that will build up the component DOM structure.

```
/**
 *
 * @param {String} emptySignalSrc
 * @param {String} fullSignalSrc
 * @param {Number} signalWidth
 * @param {Number} signalHeight
 */
function NetworkSignal(emptySignalSrc, fullSignalSrc, signalWidth, signalHeight)
{
 this.signalHeight = signalHeight;
 this.signalWidth = signalWidth;

 this.emptySignalElement = null;
 this.domElement = null;

 this.interval = null;

 this.init(emptySignalSrc, fullSignalSrc);
}

NetworkSignal.prototype.init = function(emptySignalSrc, fullSignalSrc)
{
 var el = document.createElement('div');
 el.style.position = 'relative';

 var fullImage = document.createElement('img');
 fullImage.style.position = 'absolute';
 fullImage.src = fullSignalSrc;

 this.emptySignalElement = document.createElement('div');
 this.emptySignalElement.style.position = 'absolute';
 this.emptySignalElement.style.overflow = 'hidden';
 this.emptySignalElement.style.width = this.signalWidth + 'px';
 this.emptySignalElement.style.height = '0px';

 var emptyImage = document.createElement('img');
 emptyImage.src = emptySignalSrc;

 this.emptySignalElement.appendChild(emptyImage);

 el.appendChild(fullImage);
```


(C) Copyright Nokia 2014. All rights reserved.

```
 el.appendChild(this.emptySignalElement);

 this.domElement = el;
}
```

Apart from calling the init() method, that creates the component DOM structure, the constructor stores the images width and height for later usage, and defines an **interval** variable that will be used by methods defined below.

## WRT 1.0 SystemInfo Service initialization

To access SystemInfo Service API objects and properties, the following HTML code snippet must be embedded in the Widget's HTML code:

```
<embed type="application/x-systeminfo-widget" hidden="yes"></embed>
```

Done this, it is possible to use SystemInfo Service API.

- The **appendTo()** method appends the component DOM element to the parent element passed as argument, and then call the **startService()** method.
- The **startService()** method uses JavaScript **setInterval()** to periodically check the network signal, and to accordingly update the component appearance.
- The **stopService()** method can be used to stop the network signal updating, when it is no more needed.

```
/**
 *
 * @param {Node} parentElement
 */
NetworkSignal.prototype.appendTo = function(parentElement)
{
 parentElement.appendChild(this.domElement);

 this.startService();
}

/* WRT 1.0 service */
NetworkSignal.prototype.startService = function()
{
 var sysinfo = document.embeds[0];

 if(sysinfo)
 {
  var self = this;

  var signalHandler = function()
  {
   var bars = sysinfo.signalbars;

   if(bars == undefined || bars < 0)
    bars = 0;

   var value = bars * 100 / 7;

   self.updateSignalValue(value);
  };

  signalHandler();
```


(C) Copyright Nokia 2014. All rights reserved.

```
  this.interval = setInterval(signalHandler, 2000);
 }
}
NetworkSignal.prototype.stopService = function()
{
 if(this.interval != null)
 {
  clearInterval(this.interval);

  this.interval = null;
 }
}
```

In the **startService()** method, an anonymous function is defined and referenced by the **signalHandler** variable. This function is **periodically called to check the network signal**, and then it calls the **updateSignalValue()** method to update the component UI with the latest retrieved value:

```
NetworkSignal.prototype.updateSignalValue = function(value)
{
 this.emptySignalElement.style.height = Math.round(((100 - value) * this.signalHeight) /
100) + 'px';
}
```

## Downloads

Here's a list of related downloads:

- Sample WRT Widget using the NetworkSignal component: Media:NetworkSignalWidget.zip
- Zero Network Signal image: Media:Wrt signal component empty.png
- Full Network Signal image: Media:Wrt signal component full.png
- WRT NetworkSignal component source code: Media:Networksignal component.txt (just change file extension to ".js")


(C) Copyright Nokia 2014. All rights reserved.