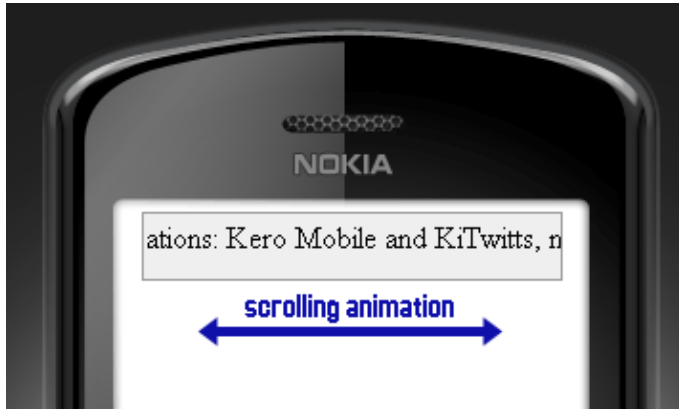




This article shows **how to build and use an animated News Ticker component** in **Web Runtime widgets**. The **News Ticker design patter** is described in detail in this Forum Nokia Wiki Article: [Mobile Design Pattern: News Ticker](#).



Contents

- [1 NewsTicker component: how to use it](#)
- [2 NewsTicker component implementation](#)
 - ◆ [2.1 NewsTicker constructor](#)
 - ◆ [2.2 Building DOM structure](#)
 - ◆ [2.3 Animating the news ticker](#)
 - ◆ [2.4 Showing the next available news](#)
 - ◆ [2.5 Managing the news ticker animation](#)
 - ◆ [2.6 Starting the component](#)
 - ◆ [2.7 Handling user interaction](#)
 - ◆ [2.8 Setting the ticker speed](#)
- [3 Downloads](#)

NewsTicker component: how to use it

The News Ticker component presented in this article can be used in any Web Runtime widget, by following these steps:

- **Include the NewsTicker.js JavaScript file** in your Widget (available for download here: [Media:Wrt NewsTicker component.zip](#))

News_Ticker_JavaScript_component_for_Web_Runtime

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

- Define a **HTML node**, in your Widget HTML code, to **contain the News Ticker component**:

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

- Define an **Array containing the news titles** to be displayed in the News Ticker:

```
var newsArray = new Array(
"Flash Lite Applications: Kero Mobile and KiTwitts, major development differences!",
"Web Runtime JavaScript battery and network signal components",
"Great News from Symbian Foundation:Opening the world of Infinite possibilities"
);
```

- **Create a NewsTicker instance** with these parameters:
 - ◆ the news **titles array**
 - ◆ the **parent HTML node**
 - ◆ the ticker **width**
 - ◆ the ticker **height**

In order to support display rotations, it is recommended to use widths expressed in percentage (e.g.: 90%).

```
var ticker = new NewsTicker(
    newsArray
    document.getElementById('ticker_container'),
    '90%',
    '30px'
);
```

- **Start the News Ticker**, by calling its **start()** method:

```
ticker.start();
```

NewsTicker component implementation

The News Ticker component will be implemented by following these steps:

- a constructor, that defines the component properties
- a method (**init()**) that initializes the component's DOM structure
- a method (**start()**) that actually starts the component
- a method (**animate()**) that performs the news title animation, letting the text scroll from right to left
- two methods (**startAnimation** and **stopAnimation()**) that will start and stop, respectively, the news ticker animation
- a method (**showNextNews()**) that shows the next available news inside the news ticker
- a method (**setNewsClickHandler()**) that allows to define a handler function to be called when the user clicks on the news ticker
- a method (**setAnimationStepPixels()**) to modify the speed of the scrolling animation

NewsTicker constructor

The NewsTicker constructor is implemented as follows, accepting 4 parameters:

- the **news titles array**
- the **parent HTML node**
- the **width** of the news ticker
- the **height** of the news ticker

It defines the properties shown in code, whose purpose is explained in the relative comment, and then calls the **init()** method, that will care of building up the component's DOM structure.

```
function NewsTicker(newsArray, parentElement, tickerWidth, tickerHeight)
{
  /* ticker animation step (in pixels) */
  this.animationStep = 2;

  /* DOM element containing the currently shown news title */
  this.newsElement = null;

  /* empty DOM element used to calculate the visible ticker width */
  this.emptyElement = null;

  /* array containing the news titles */
  this.newsArray = newsArray;

  /* index of the news currently shown */
  this.currentNewsIndex = 0;

  /* ID of the animation returned by setInterval() */
  this.animationInterval = null;

  /* news click handler */
  this.newsClickHandler = null;

  /* call to the DOM initialization method */
  this.init(parentElement, tickerWidth, tickerHeight);
}
```

Building DOM structure

The **init()** method builds up the DOM structure of the component, designed as follows:

- an **outer DIV element**, with the desired width and height. This element has the **overflow** CSS property equal to **hidden**, to hide the text overflowing the ticker width
- an **inner DIV element, that will actually contain the news title text**, with these characteristics:
 - ◆ **absolute positioning** that allows to perform the text scrolling animation
 - ◆ **white-space: nowrap** CSS property, that will allow the text to be displayed in a single line, even if longer than the available space
- an **empty DIV element, that will be used to calculate the available space** for the news' titles

Done this, the component DOM element is appended to the parent element passed as argument, and is **initialized with the title of the first available news**.

News_Ticker_JavaScript_component_for_Web_Runtime

```
NewsTicker.prototype.init = function(parentElement, tickerWidth, tickerHeight)
{
    var mainElement = document.createElement('div');
        mainElementName = NewsTicker.cssClass;
        mainElementWidth = tickerWidth;
        mainElementHeight = tickerHeight;
        mainElementOverflow = 'hidden';
        mainElementPosition = 'relative';

    var fakeElement = document.createElement('div');
        mainElement.appendChild(fakeElement);
    this.emptyElement = fakeElement;

    var textElement = document.createElement('div');
        textElementPosition = 'absolute';
        textElementWhiteSpace = 'nowrap';

    this.newsElement = textElement;

        mainElement.appendChild(textElement);

        parentElement.appendChild(mainElement);

        textElement.innerHTML = this.newsArray[0];
}
}
```

Animating the news ticker

In order to display the whole news title text, **a right-to-left scrolling animation will be implemented**. The following **animate()** method takes care of performing a single animation step and, if the right bound of the news text is already visible, it calls the **showNextNews()** method (implemented later) to display the following news. A **small delay (2 seconds)** is used before showing the following news, **to allow the user to easily read the whole news title**.

```
NewsTicker.prototype.animate = function()
{
    var textLeft = parseInt(this.newsElement.offsetLeft);
    var textWidth = parseInt(this.newsElement.offsetWidth);

    if(textLeft + textWidth > this.emptyElement.offsetWidth)
    {
        var newLeft = textLeft - this.animationStep;

        this.newsElement.style.left = newLeft + 'px';
    }
    else
    {
        this.stopAnimation();
    }

    var self = this;

    (function()
    {
        setTimeout(
            function()
            {
                showNextNews();
            },
            self,
            2000
        );
    });
}
```

```
}
```

Showing the next available news

When a single news title reaches the end of its animation, then it has to be replaced with the following one. This will be done by the following `showNextNews()` method, that takes care of:

- **resetting the news DOM element position** (because of the animation movement)
- **replacing the old news title with the new one**
- **restarting the news scrolling animation** after a small delay (2 seconds) to allow the user to easily read the first part of the news title

```
NewsTicker.prototype.showNextNews = function()
{
  this.currentNewsIndex = (this.currentNewsIndex + 1) % this.newsArray.length;

  this.newsElement.style.left = '0px';

  this.newsElement.innerHTML = this.newsArray[this.currentNewsIndex];

  var self = this;

  setTimeout
  function()
  {
    startAnimation(); self.
  },
    2000
);
}
```

Managing the news ticker animation

In order to manage and perform the news ticker animation, the `setInterval()` method is used. **The following methods will respectively start and stop the animation:**

```
/**
 * Starts the news ticker animation
 */
NewsTicker.prototype.startAnimation = function()
{
  if(this.animationInterval == null)
  {
    var self = this;

    this.animationInterval = setInterval(
    function()
    {
      animate(); self.
    },
      100
    );
  }
}
```

News_Ticker_JavaScript_component_for_Web_Runtime

```
/**
 * Stops the news ticker animation
 */
NewsTicker.prototype.stopAnimation = function()
{
  if(this.animationInterval != null)
  {
    (this.$animationInterval);

    this.animationInterval = null;
  }
}
```

Starting the component

As seen from the first part of this article, after having instantiated a NewsTicker, it is necessary to start it via the **start()** method. The **start()** method, implemented as follows, uniquely calls the **startAnimation()** method, with a delay of 2 seconds, to start the news ticker animation.

```
NewsTicker.prototype.start = function()
{
  var self = this;

  setTimeout
  function()
  {
    startAnimation(); self.
  },
    2000
);
}
```

Handling user interaction

In order to **allow user interaction** on the News Ticker component, a handler function must be called. Specifically, we want the component to call a handler function **when the user clicks on the component** itself, passing the **currently shown news' index as argument**. This is done by the following **setNewsClickHandler()** function:

```
NewsTicker.prototype.setNewsClickHandler = function(clickHandler)
{
  this.newsClickHandler = clickHandler;

  if(clickHandler != null)
  {
    var self = this;

    this.newsElement.onclick = function()
    {
      newsClickHandler(self.currentNewsIndex);
    }
  }
  else
  {
    this.newsElement.onclick = undefined;
  }
}
```

```
}
```

Setting the ticker speed

In order to have a **faster or slower News Ticker instances**, it is possible to modify the **animationStep** property, that defines the number of scrolled pixels for each animation step. The following **setAnimationStepPixels()** will allow this:

```
NewsTicker.prototype.setAnimationStepPixels = function(stepPixels)
{
  if(stepPixels < 0)
    = 1; stepPixels

  this.animationStep = stepPixels;
}
```

Downloads

The following files, related to this article, are available for download:

- A sample Web Runtime widget using the News Ticker component: [Media:NewsTickerWidget.zip](#)
- The News Ticker component source code: [Media:Wrt_NewsTicker_component.zip](#)