WebProgressBar White Paper

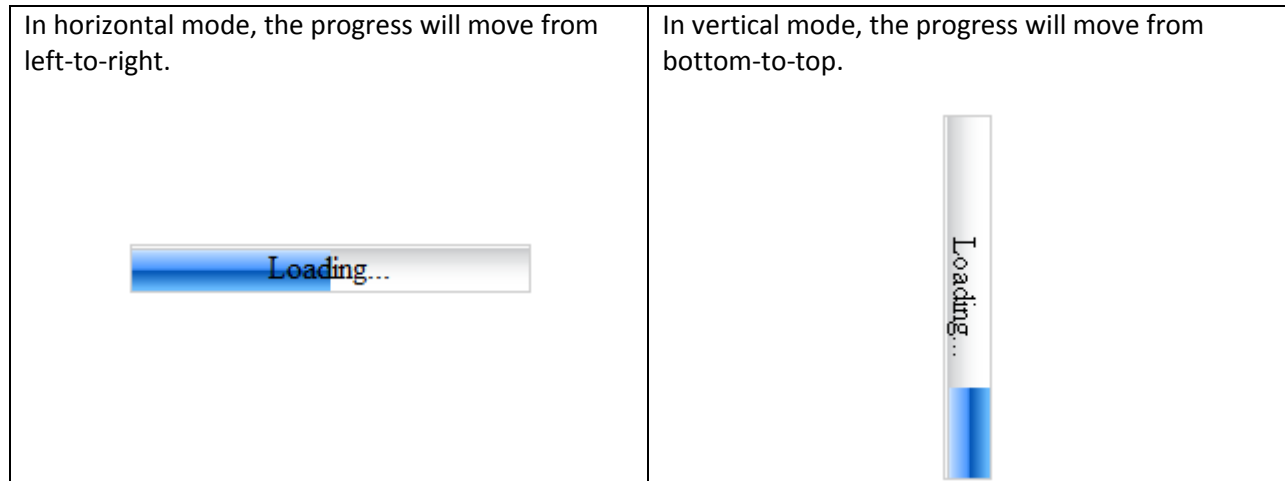This whitepaper describes the concept and features introduced in WebProgressBar.

Table of Contents

View Customization

WebProgressBar supports horizontal and vertical orientations.

In horizontal mode, the progress will move from left-to-right.	In vertical mode, the progress will move from bottom-to-top.
Loading...	Loading...

Behavior Customization

WebProgressBar offers two progress modes, *determinate* and *indeterminate*. In *Determinate* option when the progress reaches 100%, the entire progress bar will be filled. The *Indeterminate* mode, the progress indicator runs continuously in an infinite cycle.

Loading...   Loading...
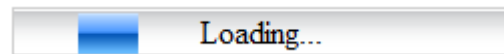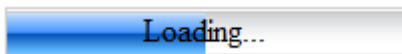
Integration to FullPostBack and FlyPostBack (aka AJAX)

Postback in WebProgressBar is triggered by several occasions, such as: on after certain interval passed, finished progress bar, or when the progress bar's value has changed.

Like the other WebEssential controls, WebProgressBar offers two postback options, FullPostBack and *FlyPostBack*; by simply set the **PostBackMode** property.
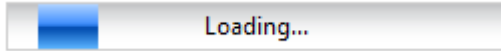
XHTML Compliance

WebProgressBar is designed to fully comply with W3C's XHTML standard – providing identical rendering when running in different modern XHTML-ready browsers.
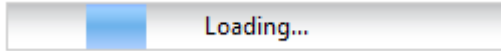
Appearance and Styles

Default Styles

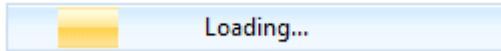WebProgressBar includes four predefined styles that you can select from properties.
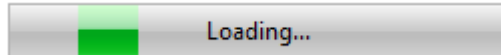
- Default

- Safari

- Office 2007

- Win7

Round Corner

WebProgressBar provides true round corner rendering using CSS3's round corner property. However, this is limited to non-IE browser only, as IE, even in V8, does not support CSS3. A workaround in http://code.google.com/p/curved-corner/ is used to achieve the equally elegant and consistent result.

If you want to make the corner rounder, simply use the border-radius CSS3's property in **ProgressContainerBarStyle** and **ProgressBarStyle** properties. For example:

-moz-border-radius: 3px; -webkit-border-radius: 3px; -khtml-border-radius: 3px; border-radius: 3px;

Layout overview
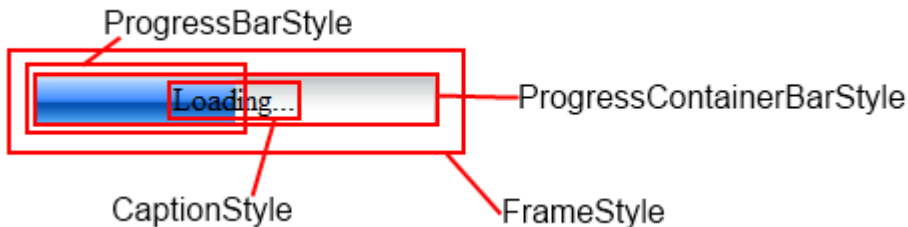
Indeterminate Animation

During Inderterminate mode, the animation could be controlled in 2 ways, using *Step* property and *RefreshAnimation* property.

The *Step* property will determine how far the ProgressBar will move after each tick, by default it is set to 0.01 (1% of width in horizontal WebProgressBar).

The *RefreshAnimation* will determine how fast is the next tick, by default it is set to 500 ms.

Comprehensive Client-side Events

The following is the list of client side events in WebProgressBar.

OnValueChanged(*controlId, value*)

This client-side event will be invoked when the value is changed.

OnProgressInterval(*controlId, value, caption*)

This client-side event will be invoked after a certain interval has passed.

OnFinished(*controlId, endTime, value, caption*)

This client-side event will be invoked after the progress has finished.

OnInitialize(*controlId*)

This client-side event will be invoked when WebProgressBar has finished the initializing process.

OnResponse(*controlId, action, xmlResponse*)

This client-side event will be invoked when response is returned after a FlyPostBack action.

OnResponseError(*controlId, errorMessage*)

This client-side event will be invoked if error occurs during a FlyPostBack action.

Rich Client-side APIs

The following is the list of client side APIs in WebProgressBar.

Functions	Description
GetElement ()	Gets WebProgressBar's frame element.
GetBarElement ()	Gets WebProgressBar's bar element.
GetCaptionElement ()	Gets WebProgressBar's caption element
DoStep ()	Animates the progress according the specified setting.
GoToStep (*n*)	Moves the progress bar to a specific point determined by the *n* value.
SetValue (value)	Sets WebProgressBar's value and move it to the point determined by the value.
SetCaption (caption)	Sets WebProgressBar's caption text.
HideCaption()	Hides the caption element of the WebProgressBar.
ShowCaption()	Shows the caption element of the WebProgressBar.
Start()	Indicates the WebProgressBar instance as started.
Finish()	Indicates the WebProgressBar instance as finished.