

```
package br.com.optimedia.assets
{
    import flash.events.Event;
    import flash.events.IOErrorEvent;
    import flash.events.ProgressEvent;
    import flash.events.SecurityErrorEvent;

    import mx.controls.Image;
    import mx.controls.ProgressBar;
    import mx.controls.ProgressBarMode;
    import mx.events.FlexEvent;
    import mx.events.ResizeEvent;

    public class ImageWithProgressBar extends Image
    {
        private var _progressBar:ProgressBar;

        private var _progressPercentWidth:Number = 80;
        private var _progressHeight:Number = 20;

        public function set progressPercentWidth( value:Number ):void
        {
            _progressPercentWidth = value;
            evaluateProgressSize();
        }
        public function get progressPercentWidth():Number
        {
            return _progressPercentWidth;
        }

        public function set progressHeight( value:Number ):void
        {
            _progressHeight = value;
            evaluateProgressSize();
        }
        public function get progressHeight():Number
        {
            return _progressHeight;
        }

        public function set progressLabel( value:String ):void
        {
            _progressBar.label = value;
        }
        public function get progressLabel():String
        {
            return _progressBar.label;
        }

        public function ImageWithProgressBar()
        {
            this.addEventListener( ResizeEvent.RESIZE, resizeHandler );
            this.addEventListener( Event.OPEN, openHandler );
        }
    }
}
```

```
this.addEventListener( Event.COMPLETE, completeHandler );
this.addEventListener( IOErrorEvent.IO_ERROR, ioErrorHandler );
this.addEventListener( SecurityErrorEvent.SECURITY_ERROR, securityErrorHandler );
}

override protected function createChildren():void
{
    _progressBar = new ProgressBar();
    _progressBar.addEventListener( FlexEvent.CREATION_COMPLETE, progressCreated );
    _progressBar.visible = false;
    _progressBar.includeInLayout = false;

    _progressBar.mode = ProgressBarMode.EVENT;
    _progressBar.source = this;

    _progressBar.label = "%3%";
    _progressBar.labelPlacement = "center";

    this.addChild( _progressBar );
}

protected function evaluateProgressSize():void
{
    if( _progressBar == null )
        return;

    _progressBar.x = -135;
    _progressBar.width = 270;
    _progressBar.height = progressHeight
}

protected function showProgress():void
{
    this.dispatchEvent( new ProgressEvent(ProgressEvent.PROGRESS, false, false, 0, 100 )
);

    _progressBar.visible = true;
}

protected function hideProgress():void
{
    _progressBar.visible = false;
}

private function progressCreated( event:FlexEvent ):void
{
    evaluateProgressSize();
}

private function resizeHandler( event:ResizeEvent ):void
{
    evaluateProgressSize();
}
```

```
private function openHandler( event:Event ):void
{
    showProgress();
}

private function completeHandler( event:Event ):void
{
    hideProgress();
}

private function ioErrorHandler( event:IOErrorEvent ):void
{
    hideProgress();
}

private function securityErrorHandler( event:SecurityError ):void
{
    hideProgress()
}
}
```