

```

/*
Working with CSS Styles Programmatically in Flex
http://userflex.wordpress.com/2008/03/06/css-in-as3/
*/
package com.userflex.sample.util
{
    // import statements
    import mx.styles.CSSStyleDeclaration;
    import mx.styles.StyleManager;

    /**
     * The StyleUtil class is an all-static class with methods for working
     * with CSS styles in Flex.
     */
    public class StyleUtil
    {
        /**
         * Defines a style property for a class or type selector.
         *
         * @param selector the class or type selector
         * @param property the style property
         * @param value the value of the style property
         */
        public static function setCSSStyle (selector : String, property :
            String, value : *) : void
        {
            var styleDeclaration : CSSStyleDeclaration =
                StyleManager.getStyleDeclaration (selector);

            if (styleDeclaration == null)
            {
                styleDeclaration = new CSSStyleDeclaration ();

                styleDeclaration.setStyle (property, value);

                StyleManager.setStyleDeclaration
                    (selector, styleDeclaration, true);
            }
            else
            {
                styleDeclaration.setStyle (property, value);
            }
        }

        /**
         * Returns the style property value for a class or type selector.
         *
         * @param selector the class or type selector

```

```

* @param property the style property
*
* @return the value of the style property
*/
public static function getCSSStyle (selector : String, property :
    String) : *
{
    var styleDeclaration : CSSStyleDeclaration =
        StyleManager.getStyleDeclaration (selector);

    if (styleDeclaration == null)
    {
        styleDeclaration = new CSSStyleDeclaration ();

        StyleManager.setStyleDeclaration
            (selector, styleDeclaration, true);
    }

    return styleDeclaration.getStyle (property);
}

/**
* Converts an object into a hexadecimal value.
*
* @param value the object to convert
*
* @return the hexadecimal value of the object
*/
public static function hex (value : *) : uint
{
    var hexValue : uint;

    if (value is uint)
    {
        // nothing to change!
        hexValue = value;
    }
    else if (value is String)
    {
        var stringValue : String = value as String;

        // removes any pound (#) signs
        stringValue = stringValue.replace (/#/g, "");

        // checks if the string already has a hexadecimal prefix
        if (stringValue.indexOf ("0x") == -1)
        {
            // if not, we add it!
            hexValue = new uint ("0x" + stringValue);
        }
        else
        {

```

```

        // otherwise we just convert the string!
        hexValue = new uint (value);
    }
}
else
{
    // not guaranteed to work, but worth trying!
    hexValue = new uint (value);
}

return hexValue;
}

/**
 * Converts a hexadecimal value into a string.
 *
 * @param value the hexadecimal value to convert
 *
 * @return the string representation of the hexadecimal value
 */
public static function hexToString (value : uint) : String
{
    return Number (value).toString (16);
}

}
}
}

```