

ellucian™

# Banner Flexible Registration Style Customization Guide

Release 8.6  
June 2013



Banner®, Colleague®, PowerCAMPUS®, Luminis® and Datatel® are trademarks of Ellucian or its affiliates and are registered in the U.S. and other countries. Ellucian, Advance, DegreeWorks, fsaATLAS, Course Signals, SmartCall, Recruiter, MOX, ILP, and WCMS are trademarks of Ellucian or its affiliates. Other names may be trademarks of their respective owners.

© 2011-2013 Ellucian. All rights reserved. The unauthorized possession, use, reproduction, distribution, display or disclosure of this material or the information contained herein is prohibited.

Contains confidential and proprietary information of Ellucian and its subsidiaries. Use of these materials is limited to Ellucian licensees, and is subject to the terms and conditions of one or more written license agreements between Ellucian and the licensee in question.

In preparing and providing this publication, Ellucian is not rendering legal, accounting, or other similar professional services. Ellucian makes no claims that an institution's use of this publication or the software for which it is provided will guarantee compliance with applicable federal or state laws, rules, or regulations. Each organization should seek legal, accounting and other similar professional services from competent providers of the organization's own choosing.

Prepared by: Ellucian  
4375 Fair Lakes Court  
Fairfax, Virginia 22033  
United States of America

## Revision History

<b>Publication Date</b>	<b>Summary</b>
June 2013	New version that supports Banner Flexible Registration 8.6 software.

# Banner Flexible Registration 8.6 Style Customization Guide

## Contents

---

- Customizing Styles . . . . . 5**
  
- Cascading Style Sheets and Banner Flexible Registration . . . . . 5**
  
- Difference between CSS in Adobe and CSS in HTML . . . . . 6**
- CSS attribute definitions . . . . . 6
- CSS selectors . . . . . 6
  - Global selector . . . . . 6
  - Application selector . . . . . 7
  - Type selectors . . . . . 7
  - Class selectors . . . . . 8
  
- Customization process. . . . . 8**
- Customization files . . . . . 9
- Customizing styles . . . . . 9
  
- Examples . . . . . 9**
  
- CSS inspector . . . . . 10**
  
- Structure of the Flexible Registration CSS file . . . . . 10**
  
- Known Limitations . . . . . 11**
- Styling final invoices and past invoices. . . . . 11
  
- Other references. . . . . 11**
  
- Updated styles . . . . . 11**



# Customizing Styles

---

This guide discusses how to customize styles in the Banner Flexible Registration module of the Banner Student System.

The ability to customize the look of Banner Flexible Registration enables an institution to provide its end users a consistent and familiar user experience.

This guide will discuss how the user interface can be customized to adhere to institution specific style standards and help incorporate a common brand and image.

It will provide a brief overview on the following:

- How the application uses cascading style sheets?
- How they can be tailored to existing institution standards?
- How the files are organized?

After reviewing this document, the reader will be familiar with how to alter the installed style sheets and understand how to tailor the user interface look such that it can adhere to the readers institution standards.

## Cascading Style Sheets and Banner Flexible Registration

---

Cascading Style Sheets (most referenced by the term CSS) is a language and format used to describe various user interface look and format, most commonly found on web sites and or Web applications. The use of CSS helps separate such things as typography, color, and other layout concepts from the actual web page or screen control. This approach enables a high degree of customization and is how institutions can easily tailor the look of Banner Flexible Registration.

One of the major benefits of using CSS is the ability to reuse and simplify customization. Most specifically, it is the ability to define once but reference and reuse many times.

For example, the definition of fonts. If used properly, a CSS structured approach would enable the definition of a common font used throughout the entire interface in just one location. Therefore, making it quite simple to change over time.

Banner Flexible Registration uses CSS to format the layout of screens and pages, including elements such as the layout, colors, and fonts.

# Difference between CSS in Adobe and CSS in HTML

---

Banner Flexible Registration utilizes Adobe Flex and the Flash Player to deliver the bulk of the user interface. Although CSS is a common web standard, some content providers further customize the CSS standard itself and create proprietary extensions. In the case of Banner Flexible Registration there are two views and support models for CSS. Adobe, the provider of the Flash platform, and HTML which is delivered directly through the end users Internet browser.

The following sections describe the differences between Adobe Flex CSS and CSS in HTML.

## CSS attribute definitions

CSS attribute definitions are different in Adobe CSS compared to standard HTML CSS. Adobe CSS does not have “-” in it’s attribute names and are camel cased.

For example,

- `fontSize` - Adobe CSS format
- `font-size` - standard CSS format
- `backgroundColor` - Adobe CSS format
- `background-color` - standard CSS format
- Adobe Flex 3.1 only supports a subset of the CSS 1.0 specification. This means standard properties like height or width cannot be used. Refer to [http://www.loscavio.com/downloads/blog/flex3\\_css\\_list/flex3\\_css\\_list.htm](http://www.loscavio.com/downloads/blog/flex3_css_list/flex3_css_list.htm) for a full list of supported properties.

## CSS selectors

The following sections discuss the various types of selectors supported in Banner Flexible Registration.

### Global selector

Adobe Flex includes a global selector that you can use to apply styles to all controls. Properties defined by a global selector apply to every control unless that control explicitly overrides it. Because the global selector is like a type selector, it’s definition is not prefaced with a period in CSS.

The following example defines *fontSize* and *textDecoration* to the global selector:

```
global
{
    fontSize:22;

    textDecoration: underline;
}
```

## Application selector

The Application container is the top-most container in a Adobe Flex application. Styles defined on the Application type selector that are inheritable are inherited by all of the container's children as well as the container's subclasses. Styles that are not inheritable are only applied to the Application container itself and not its children. Styles applied with the Application type selector are not inherited by the Application object's children if those styles are non-inheritable.

When the styles are defined for the Application type selector, it is not required to declare a style for each component, because the components are children of these classes and inherit the Application type selector styles.

Use the following syntax to define styles for the Application type selector:

```
Application
{
    style_definition
}
```

## Type selectors

Type selectors assign styles to all components of a particular type. When you define a type selector, you are not required to explicitly apply that style. Instead, Adobe Flex applies the style to all classes of that type. Adobe Flex also applies the style properties defined by a type selector to all subclasses of that type.

The following example shows a type selector for Button controls:

```
Button
{
    fontSize: 15;
    color: #9933FF;
}
```

## Class selectors

Class selectors define a set of styles (or a class) which can be applied to any component. First define the style class, and then point to the style class using the `styleName` property of the component's MXML tag. All Adobe Flex components that are a subclass of the `UIComponent` class support the `styleName` property.

The following example defines a new style *myFontStyle*

```
.myFontStyle
{
    fontSize: 15;
    color: #9933FF;
}
```

### Note

The names of class selectors cannot include hyphens in Adobe Flex. If you use a hyphenated class selector name, such as `my-class-selector`, Adobe Flex ignores the style. ■

## Customization process

---

The customization process to change a desired style is designed to be simple and efficient. The actual changing of the styles must be done by an application administrator that has access to the application server where Banner Flexible Registration is installed.

The administrator must also have the ability to edit text files and save them in their respective locations.

Before making any changes to any installed files, it is highly recommended to make backup copies for easy rollback if needed.

## Customization files

There are three CSS files used by the application. The styles that are used by the Adobe Flash component of the user interface can be found in a *styles\_ascii.css* file located in the sub directory called *styles*. The styles used by the header bar are located in a *navigation.css* file located in the *\navigation\css* sub directory. The files that control the look and feel of the file attachment window during the registration process are located in an *attachment.css* file in the *\attachments\css* sub directory.

## Customizing styles

The following instructions describe how to customize a style:

1. If you want to change the styles in the header, you can use your Internet browser to inspect the element in the header bar or view its source to find the CSS element. If you want to change a style in the Adobe Flash component, locate the style selector using the CSS Inspector.
2. Locate the appropriate CSS file to change.
3. Edit the CSS file using a non-word processing editor, such as Notepad, Notepad++, or TextPad, and so on. Once the desired selector is located, modify any existing CSS attributes within the selector, or add new ones. Ensure that any newly added attributes are supported by Adobe FLEX by referring to [“CSS attribute definitions” on page 6](#).
4. Save the CSS file.
5. Refresh the page to see the changes with in the application.

### Note

The application server should not need to be restarted for changes to take effect. ■

## Examples

---

1. How do I change the font family from Arial to Verdana across the application?
  - Locate the *global* selector in the CSS file.
  - Change the *fontFamily* property value from *arial* to *verdana*.
  - Save the CSS and refresh the page to see the changes.
2. How do I change the Advanced Search link color from blue to red?
  - Navigate to the landing page which contains the Advanced Search link.

- Press the SHIFT key and left-click on that link.
- A popup window appears with the style name applied on that link. In this case, the style name is underlineLinkButton.
- Edit the CSS file.
- Change the *color* property value from #3880ad to #ff0000.
- Save the CSS and refresh the page to see the changes.

## CSS inspector

---

The CSS Inspector alert box is a popup dialog that displays the CSS style information for a given component. It shows the style name of the component that the mouse is hovering over. This information can then be used to locate the appropriate selector in the CSS style sheet in order to make changes to the given style. The CSS Inspector can be displayed by pressing the shift key over the desired visual component and clicking the left mouse button.

## Structure of the Flexible Registration CSS file

---

The CSS file that the Banner Flexible Registration utilizes is comprised of the following 11 sections:

- Global navigation styles - Contains styles for the global navigation
- Landing page styles - Contains styles for the landing page
- Search styles - Defines all search styles (advanced, compact and expanded).
- Landing page styles
- Schedule styles - Styles for both the weekly view and list view
- Registration styles - Contains styles for course options, registration results, payment options, payment, and final invoice
- Section details styles - Styles for the section details component
- UI Themes styles - Contains styles for the components in the uithemes lib
- Common styles - Styles that are shared across multiple components
- Default FLEX components - Defines styles for all the default Adobe Flex components

## Known Limitations

---

There are known limitations when customizing styles for Banner Flexible Registration. This section outlines these limitations.

### Styling final invoices and past invoices

The user's schedule is displayed on the invoice on the final invoice and past invoice screens. The following styles are used to display the data grid on these invoices:

- `invoiceScheduleRendererHeader`
- `invoiceScheduleRendererDatagrid`

The data grid header has a fixed height. The style that governs the look of the data grid header is `invoiceScheduleRendererHeader`. If the font is changed for this style, please be aware that larger sized fonts will truncate the text of the column headers due to their fixed height.

## Other references

---

There are many reference materials and books published that will provide more education on how to use and tailor with CSS.

The W3 Schools publishes many online reference guides and tutorials for learning CSS.

## Updated styles

---

The following styles were added to the `styles_ascii.css` file in the 8.6 release:

- `groupRegistrationChoicePromptComboBox`
- `groupRegistrationChoicePromptLabel`
- `groupRegistrationChoiceBar`
- `learnerClassRendererPadding`
- `learnerClassItemRenderer`
- `learnerClassItemRendererCanvas`
- `learnerClassList`
- `learnerClassListHeaderText`
- `wrappedContainerTopFooter`

- learnerClassQuantity
- learnerClassConnectorEmptyClassMsgOverlay
- learnerClassConnectorEmptyClassMessage
- learnerClassConnectionListArea
- learnerClassConnectionRenderer
- learnerClassListSelClass
- learnerClassListSelectAll
- learnerClassListSelArea
- learnerClassPageFooter
- learnerClassPageTopFooter
- learnerClassRendererActions
- learnerClassConnectionCheckbox
- relationshipManagementTab
- userRelationshipsDatagrid
- classOptionRendererOptionTitle
- classOptionRendererSectionTitle
- classOptionRenderer
- classOptionRendererOptions
- classOptionLearnerList
- classOptionRendererOptionsSectionTitleRow
- classOptionsRendererOptionsSectionContainer
- classOptionsRendererOptionsSectionContainerItem
- classOptionsListCanvas
- classOptionRendererLearnerTitle
- registrationUserList
- registrationRendererBottomStroke
- openLearningOptionsList
- openLearningRendererOptionsSectionTitleRow
- openLearningOptionRendererSectionTitle
- openLearningOptionLearnerList
- openLearningOptionsListCanvas
- openLearningOptionsPageActionBar
- openLearningRendererOptionsDescDates

- openLearningOptionRenderer
- openLearningSetAllOptions
- openLearningOptionsFieldLearnerNameHeader
- openLearningOptionsFieldStartDateHeader
- openLearningOptionsFieldEndDateHeader
- openLearningOptionsFieldLearnerName
- openLearningOptionsFieldStartDateError
- openLearningOptionsFieldStartDate
- openLearningOptionsFieldEndDate
- openLearningRendererOptionsSectionContainer
- openLearningRendererOptionsSectionContainerItem
- openLearningSectionByLearnerOptionRenderer
- invoicePaymentAmount
- angledCollisionConflictBgrdGroup
- Button.registrationRecordListRenderCflctGrp
- progressBarBackground
- activeProgressNode
- progressBarActiveText
- progressBarDoneText
- progressBarUpcomingText

The following styles were updated in the 8.6 release:

- classOptionsList
- classOptionsPageActionBar
- registrationList
- RegistrationSectionRenderer
- registrationSectionRendererChargeColumn
- registrationSectionRendererDetailsColumn
- invoiceRequiredMaterials
- invoiceClassDescription
- classOptionRendererOptionTitle
- classOptionRendererSectionTitle

The following style was deleted in the 8.6 release:

- `courseOptionsRenderer`

The following styles were added to the *attachment.css* file in the 8.6 release:

- `sectionAttachmentSectionTitle`
- `sectionAttachmentContainer`
- `sectionLearnerList`
- `sectionAttachmentLearnerName`
- `sectionLearnerListItem`
- `attachmentTemplateText`
- `attachmentHelpText`

The following style was updated in the *attachment.css* file in the 8.6 release:

- `sectionAttachment`