

# Customizations Migration Guide

---

PowerSchool 8.x  
Student Information System

**Released June 2014**

Document Owner: Documentation Services

Content Owner: J. Cory

This edition applies to Release 8.0.0 of the PowerSchool software and to all subsequent releases and modifications until otherwise indicated in new editions or updates.

The data and names used to illustrate the reports and screen images may include names of individuals, companies, brands, and products. All of the data and names are fictitious; any similarities to actual names are entirely coincidental.

PowerSchool is a trademark, in the U.S. and/or other countries, of Pearson Education, Inc. or its affiliate(s).

Copyright © 2014 Pearson Education, Inc. or its affiliates. All rights reserved. All trademarks are either owned or licensed by Pearson Education, Inc. or its affiliates. Other brands and names are the property of their respective owners.

# Contents

<b>Overview</b> .....	<b>4</b>
Purpose .....	4
Audience .....	4
<b>Benefits of CSS</b> .....	<b>5</b>
<b>Effects on Customizations</b> .....	<b>6</b>
Case 1 .....	6
Case 2 .....	6
Case 3 .....	6
<b>Common UI Structures</b> .....	<b>7</b>
Wildcards .....	7
CSS Includes .....	7
Page Title Text.....	7
Rounded Boxes .....	8
Buttons.....	8
UI and Widget Reference.....	8
Legacy Migration Exceptions .....	9
<b>Model Code</b> .....	<b>10</b>
Old Code .....	10
Customized Old Code.....	12
New Code.....	15
New Customized Code .....	16
<b>UI Examples</b> .....	<b>20</b>
How to Access UI Examples via PowerSchool .....	20
How to Access UI Examples Via PowerSchool Administrator .....	20
<b>Custom Page Localization</b> .....	<b>22</b>
How to Convert Custom Pages for Localization .....	23
<b>Further Reading</b> .....	<b>24</b>
PowerSchool Reference Pages.....	24
Standards .....	24
Books .....	24
Resources .....	24

## Overview

With the release of PowerSchool 7, significant enhancements have been made to PowerSchool, including the use of a Cascading Style Sheet, or CSS. The new CSS-based structure provides a simplified structure that separates the content of the page (HTML) from the appearance of the content (CSS) into different files whereby providing you with more control over how different elements of PowerSchool web pages are displayed.

If updating from PowerSchool 6.2, the new CSS-based layout will affect your current PowerSchool custom web pages. In order for your customizations to take advantage of the new framework, see the *Effects on Customizations* section to determine how you will be affected and what migration steps, if any, need to be taken. If you do not, your customizations may no longer appear in the PowerSchool.

### Purpose

This document provides information about the benefits of using a CSS, the effects of using a CSS on existing customizations, examples of customized code, and links to additional resources.

### Audience

This document is intended for a district or school's PowerSchool system administrator or page customizer.

## Benefits of CSS

When the World Wide Web was initially established, web pages were primarily text based and very simple. As the Web gained popularity, the desire to improve the appearance and complexity of pages prompted developers to invent ways to layout and style content. Tables were a natural fit to conform content to rows and columns, and adding colors, fonts, borders and images within tables to style pages was born.

Years later Cascading Style Sheets (CSS) were introduced to provide control over the presentation of web pages, relieving the burden of layout from tables, and providing a standard way of controlling appearance. The advantage of CSS is the cleanliness, and control that it offers to web pages. Any page that utilizes a style sheet inherits the appearance declarations and your site gains consistency and global control.

In the old method you would need to open every page and edit the html tags and attributes. With CSS, changes made in the style sheet are cascaded to all the pages that refer to it. Color, font, position, borders, bullets, even visibility can all be controlled with CSS.

By separating the content from the appearance, the HTML is only providing an organizational structure for the content, making it more semantic, easier to read, and smaller in size. Another benefit is that the browser can cache the external style sheet so each subsequent page that loads only loads the content, not the styling. Other benefits include better accessibility for browsers, less reliance on javascript for effects, faster development, more flexibility, print-ready pages, and better hooks into content for JavaScript interactions and control.

# Effects on Customizations

If updating from PowerSchool 6.2, the new CSS-based page structure will affect your current PowerSchool custom web pages. How PowerSchool custom web pages will be affected is based on the type of customizations you have made to PowerSchool pages.

Each case below outlines the different types of customizations and how those customizations are impacted when updating from PowerSchool 6.2. Review each case to determine if it applies to you, and if so, what corrective steps are advised:

## Case 1

Migrate added content or removed content on an existing page

1. Open the PowerSchool Administrator application.
2. Navigate to the page in the site tree.
3. Click the Customize page button.
4. Add/remove the content in the 7.0 version of the page.
5. The new content should inherit the attributes of the page.

## Case 2

Migrate customer created page to new framework

1. Open the PowerSchool Administrator application.
2. Add a new page.
3. Use the Rich Text Editor (RTE) to inject a page template.
4. Use the RTE to inject user interface (UI) templates.
5. Copy the content of your old customized page into templates.

*Optional step*

6. Migrate your text into translatable keys.

## Case 3

Migrate customer created page, but don't update it to new framework

1. Open the PowerSchool Administrator application.
2. Add a new page.
3. Copy your old page into new page.

## Common UI Structures

This section provides examples of legacy PowerSchool HTML structures, and the new structure that replaced it. Only replaced structures, not updated, will be documented.

### Wildcards

To provide some backwards compatibility for customers using the case 3 migration plan, legacy PowerSchool wildcards were kept intact. New CSS based wildcards were developed and used for the shipping version of PowerSchool pages. These new wildcards mirror the legacy files but add a suffix to differentiate them: "\_css". These wildcards provide the main structural UI components for PowerSchool pages.

### CSS Includes

There are two new CSS includes that replace the old powerschool.css include.

```
<link href="/images/css/screen.css" rel="stylesheet"
media="screen">
```

```
<link href="/images/css/print.css" rel="stylesheet" media="print">
```

Screen.css governs the browser display page rendering, and styles everything you see on PowerSchool pages. Print.css governs printed PowerSchool pages. Every page in PowerSchool provides a print button, allowing users to print pages without the navigation, link colors, and other print irrelevant content.

### Page Title Text

At the top of every PowerSchool page is a title. The legacy structure was a table with the title text inside the inner <td> tag.

```
<!-- start of title -->
<table border="0" cellspacing="0" cellpadding="4">
<tr>
<td class="header">District Setup</td>
</tr>
</table>
<br><br>
<!-- end of title -->
```

This complicated structure has been replaced with an <h1> tag.

```
<h1>District Setup</h1>
```

## Rounded Boxes

A signature UI element of PowerSchool is the rounded box. The legacy structure was another complex table, with sliced corner images, and sides, along with spacer images to hold the structure in place. Here is an example of a portion of this structure.

```
<table border="0" cellspacing="0" cellpadding="0" align="center">
<tr>
<td colspan="2" background="/images/bond_3000_top.gif"
align="right"></td>
<td width="25"></td>
</tr>
<tr>
<td width="23" background="/images/bond_box_left_edge.gif"></td>
<td>
```

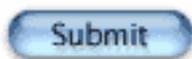
This legacy structure has been replaced with a single tag.

```
<div class="box-round"> content goes here </div>
```

The new structure's rounded corners are rendered via the CSS3 attribute `border-radius`. This provides standards compliant browsers a means to render the rounded corners, and it gracefully degrades to square corners for browsers that are not standards compliant. Either way the content and function of the page is maintained for all users.

## Buttons

The legacy structure for buttons was an `<input>` tag of `type="image"`. While this did provide a means to render the "aqua" style buttons it did not allow for customization, or translation of the text on the button.



The new structure uses either a `<button>` tag, or an `<a class="button">` tag. `Screen.css` then renders the button with the system styles.



## UI and Widget Reference

In addition to this document PowerSchool provides a set of pages with examples of the new structures as well as the widgets found throughout the system. For more information, see *UI Examples* or *PowerSchool Reference Pages*.

## Legacy Migration Exceptions

One of the legacy structures that will not render in the new framework is the `<font>` tag. This tag has been deprecated in favor of CSS, and modern browsers will ignore any use of this tag when a CSS style is in place. The simple replacement for this is using the `style=""` attribute on tags and adding any CSS based styling you require for your customizations.

## Model Code

The following examples demonstrate several sections of code. The original code that shipped with PowerSchool, a customized version, and how the customization looks in the new code. The matching code in the customized versions is grayed out to improve readability of the differences.

### Old Code

```
<table border="0" cellpadding="0" cellspacing="0" width="100%">
  <tr align="center">
    <td>
      <a href="home.html?~[randomnumber;20]"
        ONFOCUS="this.blur();">
      </a>
    </td>
    <td>
      <a href="termgrades.html?~[randomnumber;20]"
        ONFOCUS="this.blur();">
      </a>
    </td>
    <td>
      <a href="attendance.html?~[randomnumber;20]"
        ONFOCUS="this.blur();">
      </a>
    </td>
  </tr>
  ~[if.isguardian]
  <td>
    <a href="autoemailsetup.html?~[randomnumber;20]"
      ONFOCUS="this.blur();">
    
    </a>
  </td>
[/if]
  <td>
```

```

        <a href="teachercomments.html?~[randomnumber;20]"
        ONFOCUS="this.blur();" >
        
        </a>
    </td>
    <td>
        <a href="bulletin.html?~[randomnumber;20]"
        ONFOCUS="this.blur();" >
        
        </a>
    </td>
    <td>
        <a href="requestform.html?schedulerequestyearid=
        ~(schedule.yearidfuture)&subtitle=Future&~[randomnum
        ber;20]" ONFOCUS="this.blur();" >
        
        </a>
    </td>
~[if.pref.nolunchdisplay=1][else]
    <td>
        <a href="lunchbal.html?~[randomnumber;20]"
        ONFOCUS="this.blur();" >
        </a>
    </td>[/if.pref.nolunchdisplay=1]
    <td>
        <a href="mycalendars.html?~[randomnumber;20]"
        ONFOCUS="this.blur();" >
        </a>
    </td>
</tr>
</table>

```

## Customized Old Code

With this customization the customer has added additional pages to the site, and hidden others depending on the school context.

```
<table border="0" cellpadding="0" cellspacing="0" width="100%">
<tr align="center">
~[if.~(schoolid)=8407]
    <td>
        <a
            href="http://psreportviewer.ecsd.net/report.aspx?P1=
            ~(Schoolid)&P2=~(studentnumber)&P3=~(id)"
            ONFOCUS="this.blur();"></a>
        </td>
[else]
    <td>
        <a href="home.html?~[randomnumber;20]"
            ONFOCUS="this.blur();"></a>
        </td>
    <td>
        <a href="schedule.html?~[randomnumber;20]"
            ONFOCUS="this.blur();"></a>
        </td>
    <td>
        <a href="termgrades.html?~[randomnumber;20]"
            ONFOCUS="this.blur();"></a>
        </td>
    <td>
        <a href="attendance.html?~[randomnumber;20]"
            ONFOCUS="this.blur();"></a>
        </td>
~[if.isguardian]
    <td>
```

```

        <a href="autoemailsetup.html?~[randomnumber;20]"
        ONFOCUS="this.blur();"></a>
    </td>
[/if.isguardian]
    <td>
        <a href="teachercomments.html?~[randomnumber;20]"
        ONFOCUS="this.blur();"></a>
    </td>
[/if]
~[if.~(House)="KE" ]
    <td>
        <a href="home.html?~[randomnumber;20]"
        ONFOCUS="this.blur();"></a>
    </td>
    <td>
        <a href="schedule.html?~[randomnumber;20]"
        ONFOCUS="this.blur();"></a>
    </td>
    <td>
        <a href="termgrades.html?~[randomnumber;20]"
        ONFOCUS="this.blur();"></a>
    </td>
    <td>
        <a href="attendance.html?~[randomnumber;20]"
        ONFOCUS="this.blur();"></a>
    </td>
~[if.isguardian]
    <td>
        <a href="autoemailsetup.html?~[randomnumber;20]"
        ONFOCUS="this.blur();"></a>

```

```

        </td>
[/if.isguardian]
        <td>
            <a href="teachercomments.html?~[randomnumber;20]"
                ONFOCUS="this.blur();"></a>
        </td>
[/if]
        <td>
            <a href="bulletin.html?~[randomnumber;20]"
                ONFOCUS="this.blur();"></a>
        </td>
<!--ECSD Customization-remove schedule requests + lunch balances
        <td>
            <a href="requestform.html?schedulerequestyearid=
            ~(schedule.yearidfuture)&subtitle=Future&~[randomnum
            ber;20]" ONFOCUS="this.blur();"></a>
        </td>
~[if.pref.nolunchdisplay=1][else]
        <td>
            <a href="lunchbal.html?~[randomnumber;20]"
                ONFOCUS="this.blur();"></a>
        </td>
[/if.pref.nolunchdisplay=1]
        <td>
            <a href="mycalendars.html?~[randomnumber;20]"
                ONFOCUS="this.blur();"></a>
        </td>end of hidden icons)-->
</tr>
</table>

```

## New Code

This example illustrates the main navigation of Parent Access. The navigation is now an unordered list of links, the text is actual html text, and there is an ID on each list item. This code is much simpler to read and edit, and the ID's provide an additional means to customize the site.

**Note:** The list is in reverse order to enable the hiding/deletion of items and maintaining the remaining items position on the right. With the ID on each list item items can be hidden through JavaScript, or CSS.

```
<ul id="nav-main">
~[if.isguardianAndSSOEnabled]
    <li id="btn-accountPreferences">
        <a href="accountmanagement_profile.html">Account
        Preferences</a>
    </li>
[/if]
~[if.isguardian]
    <li id="btn-emailNotification">
        <a href="autoemailsetup.html">Email Notification</a>
    </li>
[/if]
    <li id="btn-myCalendars">
        <a href="mycalendars.html">My Calendars</a>
    </li>
~[if.pref.nolunchdisplay=1][else]
    <li id="btn-balance">
        <a href="lunchbal.html">Balance</a>
    </li>
[/if.pref.nolunchdisplay=1]
    <li id="btn-classRegistration">
        <a href="requestform.html?
        schedulerequestyearid=~(schedule.yearidfuture)&subti
        tle=Future">Class Registration</a>
    </li>
    <li id="btn-schoolBulletin">
        <a href="bulletin.html">School Bulletin</a>
    </li>
    <li id="btn-teacherComments">
        <a href="teachercomments.html">Teacher Comments</a>
```

```

</li>
<li id="btn-attendanceHistory">
    <a href="attendance.html">Attendance History</a>
</li>
<li id="btn-gradesHistory">
    <a href="termgrades.html">Grades History</a>
</li>
<li id="btn-gradesAttendance">
    <a href="home.html">Grades and Attendance</a>
</li>
</ul>

```

## New Customized Code

This example uses the previous customization code, but migrates it into the new framework. New links are added to "nav-main" so new styles should be added to the CSS, and new images need to be created. As buttons are added the width for the <li> width might need to be adjusted to fit additional icons. This example hides the buttons with CSS, but you could also use jQuery to hide the <li> elements too.

## Image

32bit Png files so that the background is transparent.



## CSS Style

```

#btn-myCalendars, #btn-balance, #btn-classRegistration {
    /*this style hides these buttons*/
    display:none;
}
#btn-reportViewer a {
    background: url("/images/btn_PSDLReportCard.png"
) no-repeat; }
#btn-schedule a {
    background: url("/images/btn_schedule.png"

```

```
) no-repeat; }
```

## HTML

```
<ul id="nav-main">
    <li id="btn-schoolBulletin">
        <a href="bulletin.html">School Bulletin</a>
    </li>
~[if.~(schoolid)=8407]
    <li id="btn-reportViewer">
        <a href="http://psreportviewer.ecsd.net/report.aspx?
P1= ~(Schoolid)&P2=~(studentnumber)&P3=~(id)">Report
Viewer</a>
    </li>
[else]
~[if.isguardianAndSSOEnabled]
    <li id="btn-accountPreferences">
        <a href="accountmanagement_profile.html">Account
Preferences</a>
    </li>
[/if.isguardianAndSSOEnabled]
~[if.isguardian]
    <li id="emailNotification">
        <a href="autoemailsetup.html">Email Notification</a>
    </li>□
[/if.isguardian]□
    <li id="teacherComments">□
        <a href="teachercomments.html">Teacher Comments</a>□
    </li>□
    <li id="btn-attendanceHistory">□
        <a href="attendance.html">Attendance History</a>□
    </li>□
    <li id="btn-gradesHistory">□
        <a href="termgrades.html">Grades History</a>□
    </li>□<li id="btn-schedule">□
        <a href="schedule.html">Schedule</a>□
    </li>□
    <li id="btn-gradesAttendance">□
        <a href="home.html">Grades and Attendance</a>□
```

```

        </li>
[/if]
~[if.~(House)="KE" ]
~[if.isguardianAndSSOEnabled]
    <li id="btn-accountPreferences">
        <a href="accountmanagement_profile.html">Account
        Preferences</a>
    </li>
[/if.isguardianAndSSOEnabled]
~[if.isguardian]
    <li id="emailNotification">
        <a href="autoemailsetup.html">Email
        Notification</a>
    </li>
[/if.isguardian]
    <li id="btn-myCalendars">
        <a href="mycalendars.html">My Calendars</a>
    </li>
~[if.pref.nolunchdisplay=1][else]
    <li id="btn-balance">
        <a href="lunchbal.html">Balance</a>
    </li>
[/if.pref.nolunchdisplay=1]
    <li id="btn-classRegistration">
        <a href="requestform.html?
        schedulerequestyearid=~(schedule.yearidfuture)&subti
        tle=Future">Class Registration</a>
    </li>
    <li id="teacherComments">
        <a href="teachercomments.html">Teacher Comments</a>
    </li>
    <li id="btn-attendanceHistory">
        <a href="attendance.html">Attendance History</a>
    </li>
    <li id="btn-gradesHistory">
        <a href="termgrades.html">Grades History</a>
    </li>
    <li id="btn-schedule">

```

```
        <a href="schedule.html">Schedule</a>□  
</li>□  
<li id="btn-gradesAttendance">□  
    <a href="home.html">Grades and Attendance</a>□  
</li>□  
[/if]□□  
</ul>
```

## UI Examples

The pages in PowerSchool are constructed using a standardized set of HTML tags and styles using CSS. These pages provide a reference guide so that you can more easily construct and modify your custom pages. Reusing these design patterns not only makes custom page development easier, it helps you to learn to use new pages by providing a familiar look and feel.

**Note:** For more information about custom web pages, see the PowerSchool Administrator online help or the *Custom Web Pages User Guide* available on [PowerSource](#).

### How to Access UI Examples via PowerSchool

1. On the start page, enter `/admin/ui_examples/home.html` in the Address field.
2. Press **ENTER** or **RETURN**. The UI Examples page appears.
3. Use the field description table to below to assist you in constructing and modifying your custom pages.

### How to Access UI Examples Via PowerSchool Administrator

1. Sign in to PowerSchool. The PowerSchool start page appears.
2. Click **PS Administrator**. The PowerSchool Administrator Login page appears.
3. Log in to PowerSchool Administrator. The PowerSchool Administrator start page appears.
4. Click **Custom Pages**. The Custom Pages page appears.
5. Click **UI Examples**. The UI Examples page appears.
6. Use the following tables to assist you in constructing and modifying your custom pages:

#### HTML

Link	Description
Admin Page	Click to view an overview of the Admin page structure.
Teacher Page	Click to view an overview of the Teacher page structure.
Parent Page	Click to view an overview of the Parent page structure.
The Content Section	Click to view information about how to code the content area.
Tables	Click view information about how to code tables in PowerSchool.
Static Tabs	Click to view information about how to code tabs in PowerSchool.
Buttons	Click to view information about how to code buttons in PowerSchool.

Link	Description
Tooltips	Click to view information about how to code tooltips in PowerSchool.

### JavaScript

Link	Description
Double Click Prevention	Click for information about double click prevention.
Calendar Popup	Click for information about calendar popups.
Dialog Popups	Click for information about dialog popups.
Dynamic Tabs	Click for information about dynamic tabs.
Collapsing boxes	Click for information about collapsible boxes.

### Example Page

Link	Description
<b>UI Live Example</b>	Click to view a page that has a near complete collection of all the UI elements provided in PowerSchool.

## Custom Page Localization

Customized PowerSchool pages can be converted to display on the Localize PowerSchool page for batch translation. Use the PowerSchool Administration application to convert the data for translation.

For more information on custom pages, see *the Custom Web Page User Guide* available on PowerSource, or consult the *PowerSchool Administrator online help*.

Custom Pages now displays an HTML editing toolbar to assist with editing custom pages in PowerSchool. This toolbar appears at the top of the Custom Page tab when an existing custom page is opened, or when the Customize Page button is selected.

This HTML editing toolbar contains common HTML editing functions. On the far right side of the toolbar are the following specialized functions for PowerSchool:

Toolbar Button	Description
 [Template]	Click to choose one of the following formatted PowerSchool elements to insert in the page: <ul style="list-style-type: none"> <li>• Box</li> <li>• Tabs</li> <li>• Dynamic Tabs</li> <li>• Toggle</li> <li>• Dialog</li> <li>• Calendar</li> <li>• Table</li> </ul>
 [Page Template]	Click to choose one of the following formatted PowerSchool page templates: <ul style="list-style-type: none"> <li>• Admin</li> <li>• Teacher</li> <li>• Parent</li> </ul>
 [Extract to Key]	Click to extract the selected text to a key for translation.
 [Clean]	Click to remove all HTML tags and display only the page text.
 [Preview Keys]	Click to preview the extracted keys for the page. A dialog displays the extracted key and the current text display.

Toolbar Button	Description
 [Preview Page]	Click to preview the page in the browser view.

## How to Convert Custom Pages for Localization

1. On the PowerSchool Administrator start page, click **Custom Pages**. The Custom Pages page appears.
2. From the Site Structure, select the customized page file you want to edit. The content of the selected file appears in the **Customized Page** tab view.

**Note:** You must select a page from the Site Structure in order to activate all functions on the page. If you create a new page, be sure to select it on the Site Structure to activate all functions, including creating message keys.

**Note:** To increase the view area, click the blue double arrow on the Customized Page tab to hide the Site Structure. Click the blue double arrow again to display the Site Structure.

4. Highlight the text you want to convert for translation and click the **Extract to Key** button on the HTML editing toolbar. Repeat for each text string to be translated.
5. Click the **Preview Keys** button to display all of the extracted keys and the current text value.
5. Click **Save**. A confirmation message appears.
6. Click **OK**. The edited file, indicated by blue, appears in the Site Structure, and the extracted keys are now available for batch translation on the Localize PowerSchool page in PowerSchool, and, if enabled, on individual page localization. For more information, see **Error! Reference source not found.**

**Important Note:** You cannot edit an existing text key. You must delete the key and perform the extract to key function again in order to change a key.

## Further Reading

The following is a list of resources you can reference for additional information:

### PowerSchool Reference Pages

PowerSchool HTML/CSS/JS: [/admin/ui\\_examples/](#)

1. Admin Page
2. Teacher Page
3. Parent Page
4. The Content Section
5. Tables
6. Static Tabs
7. Buttons
8. Tooltips
9. Double Click Prevention
10. Calendar Popups
11. Dynamic Tabs
12. Collapsing Boxes
13. UI Live Examples

### Standards

Web Standards Project: <http://www.webstandards.org/>

CSS and Accessibility: <http://www.w3.org/TR/CSS-access>

### Books

Three great books for developers: <http://simplebits.com/work/>

### Resources

Browser support for CSS: <http://www.quirksmode.org/compatibility.html>

Examples of CSS flexibility: <http://www.csszengarden.com/>

A JavaScript library for interactivity: <http://jquery.com/>