



MySchoolBucks
Pay Express Advanced
Configuration Guide

Table of Contents

MSB Pay Express Advanced Configuration User Guide	3
Create a Button	3
Create a Custom Form Handler	4
JavaScript Header Code Example.....	4
JavaScript Form Handler Code Example.....	5
Web Server Form Post Example	8
Button Properties Reference	10
Support	11

MSB Pay Express Advanced Configuration User Guide

Pay Express buttons are the perfect way to collect payments through your own website. When a user clicks on a Pay Express button, they will be directed to MySchoolBucks to check it out. This new feature makes it easier for users to pay for any fee or school store item.

For web developers, MSB offers a way to modify payment button behaviors associated with Pay Express using custom-created code hosted on the school district (merchant) website.

Create a Button

Click here to see a basic configuration guide.

To enable button customization, first create a payment button for a store product using the **Add** button on the “Manage Payment Buttons” page.

Set the script type to “Expanded”.

Script: Expanded

Set the Action to “Add to cart”:

Action: Add to cart and go directly to the order checkout page

Expand the “Extended Options” panel by clicking the link.

Show Extended Options ?

Then click the checkbox at the bottom to allow custom pricing.

For web developers: ?

Allow custom pricing and properties to be specified on request

Create a Custom Form Handler

There are many possible methods available to dynamically modify button behaviors with custom code. Two basic examples are provided here – a JavaScript form handler and a web server form post. You may choose whichever solution best fits your needs.

JavaScript Header Code Example

This code is available to copy and paste from MySchoolBucks. No changes are needed to support custom form handling. Insert within the <head> [...] </head> section.

```
<script type="text/javascript"
src="https://www.myschoolbucks.com/scripts/stores/checkout/checkout.js"></script>

<link rel="stylesheet" type="text/css"
href="https://www.myschoolbucks.com/scripts/stores/checkout/checkout.css"/>
```

JavaScript Form Handler Code Example

An example script is provided below that uses a web form (First Name, Last Name, Program Type, Date) to pass dynamic values for before and after school program registration. Product Name, Price, Quantity, and Student ID values are also used with custom options. Insert within the `<head> [...] </head>` section.

```
<html>
<head>
<script type="text/javascript">
    function onFormSubmit(src)
    {
        var buttons = [];
        var btn = {};
        btn.buttonID = '5eeb3e2e-322d-48f5-894b-0d9436b23fde';
        btn.productName = 'Before and After School Care Fees';
        btn.quantity = '1';
        var programType = src.elements["programType"].value;
        if( programType=='After School' )
            btn.unitPrice = '25.00';
        else if( programType=='Before School' )
            btn.unitPrice = '35.00';
        else // both
            btn.unitPrice = '50.00';
        btn.studentid
        btn.properties = [];
        btn.properties.push(createProperty('FirstName',
src.elements["firstName"].value));
        btn.properties.push(createProperty('LastName',
src.elements["lastName"].value));
    }
</script>
</head>
</html>
```

Btn.buttonID is a unique key that is pulled from the expanded script code.

Btn.productName = Product Name
Btn.quantity = item qty

programType is the type of Before/After School Program. Price is calculated from 3 programType options specified in the body form code (below).

Btn.studentid is not used in this example

```
    btn.properties.push(createProperty('ProgramType',  
src.elements["programType"].value));
```

```
    btn.properties.push(createProperty('StartDate',  
src.elements["startDate"].value));
```

```
    buttons.push(btn);
```

```
    src.jsonButtons.value = JSON.stringify(buttons);
```

```
    msbCheckout.formCheckout(src);
```

```
    }
```

```
function createProperty(name, value)
```

```
{
```

```
    var property = {};
```

```
    property.name = name;
```

```
    property.value = value;
```

```
    return property;
```

```
}
```

```
</script>
```

```
</head>
```

```
<body>
```

```
<div class="msbcheckout">
```

```
<form target="msbcheckout"  
action="https://www.myschoolbucks.com/ver2/stores/checkout/addtocart"  
onsubmit="onFormSubmit(this);">
```

```
    <input type="hidden" name="clientKey" value="0G330XPY300T2CA"/>
```

```
    <input type="hidden" name="jsonButtons" value=""/>
```

clientkey is a unique key that is pulled from the expanded script code.

```
<h1>City Public Schools Before and After School Program</h1>
```


 Before and After School Program fees are calculated and paid for a 30 day period from the start

date selected below. You can choose to pay for either or both care options. Payment is taken via your

MySchoolBucks account.</br>

<h3>Before and After School Care Fees</h3>

First and Last Name Text Box

First Name: <input type="text" name="firstName" id="firstName"/>

Last Name: <input type="text" name="lastName" id="lastName"/>

Program: <select name="programType" id="programType">

<option value="Before School">Before School (\$35.00)</option>

<option value="After School">After School (\$25.00)</option>

<option value="Before and After School">Before and After School

(\$50.00)</option>

Program combo box

</select>

Date selector

Start Date: <input type="date" name="startDate" id="startDate"/>

<input type="image"

src="https://www.myschoolbucks.com/images/en/MSB_Add_To_Cart_Button_1.png"

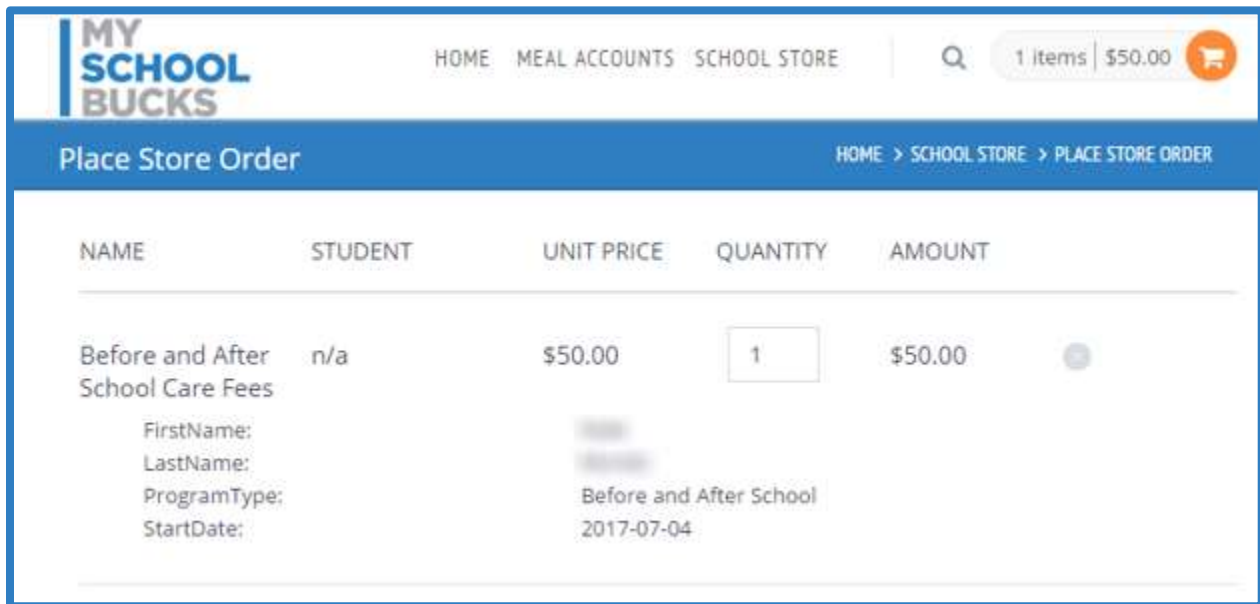
border="0"/>

Payment Button

</form>

</div>

When the form submit button is clicked, the form will post to MySchoolBucks, and the result will appear in a pop-up window with the item(s) added to the user's cart.



Web Server Form Post Example

For some scenarios, it may be useful to first post the form back to the merchant's web server to perform more complex calculations before forwarding the request to MySchoolBucks. To do that, modify the HTML form target action to post the form submit request to your web server instead of to MySchoolBucks.

```
<form target="msbcheckout" action="/postmycustomform"
onsubmit="msbCheckout.formCheckout(this);">
...
</form>
```

Create an action handler on your website to process the request. You may perform any operations required to calculate the product options and then respond to the request with an HTTP 302 redirect to the MySchoolBucks website URL

(<https://www.myschoolbucks.com/ver2/stores/checkout/addtocart>) with two string parameters.

ClientKey=<your client key goes here>

jsonButtons=<a JSON formatted string with button properties>

For example, a JSON string parameter looks like this:

```
[{'buttonID': '5eeb3e2e-322d-48f5-894b-0d9436b23fde', 'productName': 'Before and After School Care Fees', 'unitPrice': '35.00'}]
```

If needed, you may include more than one button in the request to add multiple items to the cart at once.

```
[{'buttonID': 'ef35e362-551a-4d4e-838e-67f13f576214', 'productName': 'T-Shirt', 'unitPrice': '30.00'}, {'buttonID': '7159eab2-3687-4d2d-9338-bee899b62054', 'productName': 'Shorts', 'unitPrice': '25.00'}]
```

Here's another JSON example with custom properties:

```
[{'buttonID': '5eeb3e2e-322d-48f5-894b-0d9436b23fde', 'productName': 'Before and After School Care Fees', 'unitPrice': '35.00', 'quantity': '1', 'properties': [{'name': 'First Name', 'value': 'Nate'}]}]
```

Be sure to URL encode the value for the request. We also recommend using a JSON library to format the button values. JSON libraries are freely available online for most programming languages.

The redirected HTTP request will appear in the MSB checkout pop-up window and the item(s) will automatically be added to the user's cart.

Button Properties Reference

Available button properties include:

Property	Description	Type/Length	Required/Optional
buttonID	The unique ID representing the button	String/60	Required
productName	The custom name of the product	String/80	Optional
stockOptionID	The unique ID representing the stock option	String/60	Optional
unitPrice	Payment amount (per item)	Numeric	Optional
quantity	Number of items	Numeric	Optional
studentID	The student ID representing the student	String/60	Optional
reference	A merchant-supplied value to identify the item	String/80	Optional
properties	A set of name/value pairs with custom properties	String/80	Optional

If not otherwise provided, optional values will default to those button values as provided by the store administrator on MySchoolBucks.

Support

For additional support, please contact Technical Support between the hours of 8 AM EST and 5 PM EST at 1-800-803-6755.

You may also contact customer technical support by using our [support request form](#).

This confidential document is issued pursuant to non-disclosure agreements in place and subject to the terms and conditions contained therein. Reproduction or use, without the express permission from Heartland School Solutions of any portion of this manual is prohibited. Copyright 2017, Heartland School Solutions.