



**Child Items**

## Child Items

### 1 Adding Child Items to plugin control panels.

1.1	File Structure	4
1.2	Hacking childItems.html	7
1.3	Hacking childItem.html (without the s)	14

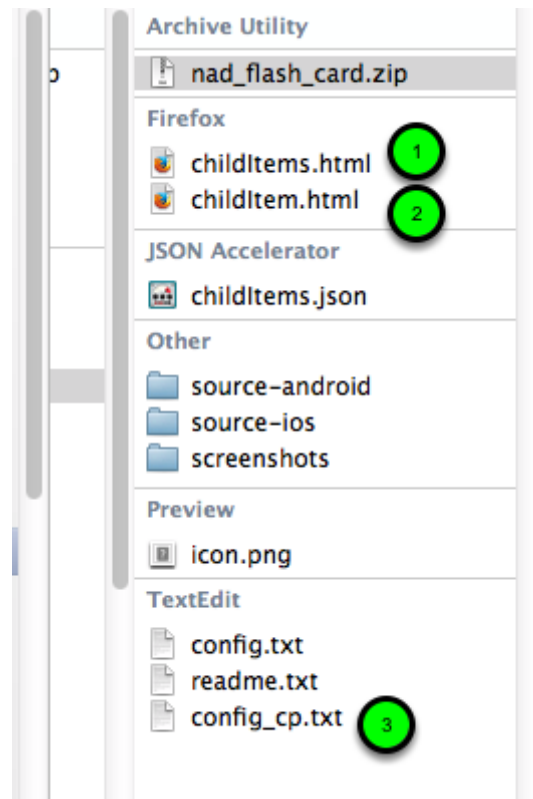
# **Adding Child Items to plugin control panels.**

# File Structure

---

Get the full sample download package here.

[https://dl.dropboxusercontent.com/u/2452503/nad\\_flash\\_card.zip](https://dl.dropboxusercontent.com/u/2452503/nad_flash_card.zip)



There are three files we will be working with

1. childItems.html
2. childItem.html (notice the difference)
3. config\_cp.txt

Screens / Actions > flash card

- Screen Nickname
- Top Navigation Bar
- Flash Card Deck 1
- Advertising Options
- Hide from Search Results


1. childItems.html (with the s) is used to create the child item menu section.

Screen Nickname

Top Navigation Bar

Flash Card Deck

You can add and remove items manually using this form or you can point to a dataURL that provides the items. Set the dataURL lower on this page in the Screen Data URL section.

Item ID	Row Number...	refresh
 027F5F13CFA4C7D8E1F8ABE	<input type="text" value="1"/>	delete

1 - 1  
of 1

Front Side Text

Back Side Text

add

1. When the user wants to edit a row item and they click on the item a pop up appears. This is the childItem.html (without the s).



This is the pop up that appears when a user wants to edit a child item row.  
This pupup uses childItem.html (without the s).



To include the child items menu section in the plugin add the following to the config\_cp.txt  
For fileType use 'customInclude' or 'custom'  
For fileName use childItems.html (with an s)  
The file childItem.html (without the s) is not referenced in the file config\_cp.txt because it gets called from inside childItems.html (with an s).

# Hacking childItems.html

---

This lesson focuses on editing the childItems.html file.

We are only going to edit some key areas in the text.

There is a lot of boiler plate html and javascript code that we are going to leave alone.

Download the example files to get the full code samples here:

[https://dl.dropboxusercontent.com/u/2452503/nad\\_flash\\_card.zip](https://dl.dropboxusercontent.com/u/2452503/nad_flash_card.zip)



Configuration Data | Publish ▼

Screens / Actions > flash card

- Screen Nickname
- Top Navigation Bar
- Flash Card Deck 1
- Advertising Options
- Hide from Search Results

P  
C

The screenshot shows a configuration interface for a 'flash card' screen. At the top, there is a breadcrumb trail 'Screens / Actions > flash card'. Below this, there are several expandable sections, each with a right-pointing chevron and a title: 'Screen Nickname', 'Top Navigation Bar', 'Flash Card Deck', 'Advertising Options', and 'Hide from Search Results'. The 'Flash Card Deck' section is currently expanded and has a green circle with the number '1' next to it, indicating it is the current focus. On the right side of the interface, there are two small icons: a 'P' icon and a 'C' icon.

Once again we use childItems.html (with an s) to create the child item menu section



```
childItems.json x childItem.html x childItems.html x config_cp.txt x
1
2
3 <!-- ##### --
>
4 <!-- ##### child items list ##### --
>
5 <div class='cpExpandoBox colorLightBg' id="section_childItems"
">
6 <a href='#' onClick="fnExpandCollapse('box_childItems')
;return false;"><img src='../..../images/arr_right.gif' alt
='arrow' />Flash Card Deck</a>
7 <div id="box_childItems" style="display:block;">
8
9 <div style='padding-top:10px;margin-bottom:10px;'>
10 You can add and remove items manually using this
form or you can point to a dataURL that provides
the items.
11 Set the dataURL lower on this page in the Screen
Data URL section.
12 </div>
```

1. For the first div we set the id to section\_childItems (with an s)
2. Inside our anchor link we put box\_childItems inside the fnExpandCollapse function.
3. In our second div we add box\_childItems for the id.





```
1 {
2   "childItems": [
3     {
4       "front_side": ""
5     },
6     {
7       "back_side": ""
8     }
9   ]
10 }
```

There are two child items we are going to add to our control panel.

front\_side

back\_side

There are four areas in the childItems.txt to edit to get these json variables into the control panel.

Be sure to download the sample to follow along

```

<!-- the div for the front side text input box -->
<div style='padding-top:5px;'>
  <!-- just some html text for the the display -->
  <b>Front Side Text</b><br/>
  <!-- your text area input field -->
  <input name='front_side' id='front_side' type='textarea' value=""
  style='width:500px; ' />
</div>

<!-- the div for the back side text input box -->
<div style='padding-top:5px;'>
  <!-- just some html text for the the display -->
  <b>Back Side Text</b><br/>
  <!-- your text area input field -->
  <input name='back_side' id='back_side' type='textarea' value=""
  style='width:500px; ' />
</div>

```

\*\*\*\*\*  
 \*\*\*\*\*-->

We set up a div for the front\_side text and add an input box to it.

1&2. The name and id field get the front\_side variable.

3. type is text or textarea

4. value is a required but is left empty

Notice that NONE of the values is prefaced with json\_ eg. front\_side

5,6,7,8 are repeated for the back\_side json variable

```
175
176 //*****
*****
177 // Add variables for your json key values here.
178 // This turns your html input into javascript variables.
179 // Later on in the code the javascript variables will be
180 // turned into json childItems
181
182 var front_side = "";
183 if(obj["front_side"] != ""){
184     front_side = obj["front_side"];
185 }
186
187 var back_side = "";
188 if(obj["back_side"] != ""){
189     back_side = obj["back side"];
190 }
191 //*****
*****
192
```

The second section to modify is inside the <script> tags. We create the javascript variables here. 1,2,3,4 use the same variable for the this whole block of code. Repeat for the back\_side json variable.

```

257
258 //reference to the form...
259 var frm = document.forms[0];
260 //*****
261 //create the JSON string to pass to use in
    childItemAddJson form element...
262 var newItemJson = "{";
263 // The itemType is important for reference purposes.
    Best
264 // to leave it alone.
265 ① → newItemJson += "\"itemType\": \"BT_menuItem\", ";
266 // Enter your first json childItem here. Then your
    second, third, etc...
267 ② → newItemJson += ③ "front_side\": \"\" + frm.front_side.value ④ +
    + "\", ";
268 // don't forget to leave out the comma at the end of the
    last json item.
269 ⑤ → newItemJson += "\"back_side\": \"\" + frm.back_side.value +
    "\"";
270 newItemJson += "}";
271 //*****
272

```

The next section is used for forming the json config file.

1. The first row of the json is itemType. This is a required field and should be left as is.
2. Add a new row here.
3. The key goes here.
4. We save the value for frontside here.
5. The back\_side element starts here. This is the last element on the list.
6. Because it is the last item we remove the comma.

```
294
295         if(status.toUpperCase() == "SUCCESS"){
296
297 //*****
298 //clear form...
299 frm.front_side.value = "";
300 frm.back_side.value = "";
301 //*****
302
303
```

In the last section we clear the javascript values.

› Screen Nickname

---

› Top Navigation Bar

---

› Flash Card Deck

You can add and remove items manually using this form or you can point to a dataURL that provides the items. Set the dataURL lower on this page in the Screen Data URL section.

Item ID	Row Number...	refresh
027F5F13CFA4C7D8E1F8ABE	<input style="width: 50px;" type="text" value="1"/>	delete

1 - 1  
of 1

**Front Side Text**

**Back Side Text**

End result is a menu table with input boxes for our two json variables.

# Hacking childItem.html (without the s)

This section deals with modifying the childItem.html file.

We will only be modifying key portions of the text.

There is a lot of boiler plate html that we do not touch.

Get the full sample package here.

[https://dl.dropboxusercontent.com/u/2452503/nad\\_flash\\_card.zip](https://dl.dropboxusercontent.com/u/2452503/nad_flash_card.zip)

```
<!-- the div for the front side text input box -->
<div style='padding-top:5px;'>
  <!-- just some html text for the the display -->
  <b>Front Side Text</b><br/>
  <!-- your text area input field -->
  <input name='json_front_side' id='json_front_side' type='textarea'
  value="" style='width:500px; ' />
</div>

<!-- the div for the back side text input box -->
<div style='padding-top:5px;'>
  <!-- just some html text for the the display -->
  <b>Back Side Text</b><br/>
  <!-- your text area input field -->
  <input name='json_back_side' id='json_back_side' type='textarea'
  value="" style='width:500px; ' />
</div>
```

The childItem.html is a little easier to work with as there is no javascript required.

This section is very similar to the html section in the childItems.html.

The one difference is we prefix the variables with json\_ eg json\_front\_side and json\_back\_side.

Input box

1. use json\_front\_side in the name field
2. use json\_front\_side in the id field.
3. use text or textarea for the type
4. the value field is required. leave it blank.

Repeat for the back\_side variable.



Configuration Data | Publish ● close X

**Item Properties**

**Front Side Text**

**Back Side Text**

The result.