

PSUG National Information Exchange

Users Helping Users



Building a Plug-in: Start to Finish

ROGER SPRIK

VALLEY CHRISTIAN SCHOOLS

TECHNOLOGY DIRECTOR



Agenda and Expectations

- Advanced.
- You're interested in a customization example and/or want to learn how to share your customization
- There have been classes this week on:
 - Database Extensions, Page Fragments, and customization
- Will NOT cover any one topic in depth
- All code is available in the online documentation
- Take what's been covered and put them all together in an example for managing student vehicles
- Install that plugin on other servers



About the trainer. . .

- Roger Sprik, Technology Director
rsprik@vcschools.org
- Valley Christian Schools, Cerritos CA
- 1300+ students Preschool - 12
- PowerSchool since 2003 K-8, 2005 9-12



Planning



Group, Table and Field names

- Database Extensions have 3 “levels” of names
 - *Group name (similar to a folder or project name)
 - *Table names
 - Field names

*These must be unique



Group, Table and Field names

- Unique group and table names important
- All group and table names start with U_
- We have tried to come up with brief but unique group names
 - U_FSH (Family Service Hours)
 - U_ETX (Equipment Tracking Extension)
 - U_LMS (LMS integration)
 - U_JAMF (JAMF MDM integration)



Group, Table and Field names

- Best Practice: Make table names unique by prepending them with the group name:
 - U_FSH_FAMILY
 - U_ETX_EQUIPMENT
- Group names are short for this reason
- Keep possible future sharing in mind when naming.



Normalization – Plan your schema

- We always “whiteboard” our schema plan.
- Always developed on test server
- Currently cannot delete or modify extended schema, so proceed with caution



U_VEX – Vehicle Extension

- Let's plan!



Vehicle Extension Plugin
Extension Name: U_VEX

Students

U_VEX_STUDENTS
Type: 1:1

Fields:
PERMIT
DRIVER_LICENSE

U_VEX_VEHICLE
Type: 1:Many

Fields:
VEHICLE_TYPE
MAKE
MODEL
COLOR*
LICENSE
NOTE

Schema Creation



Create the Extension

- System – Page and Data Management – Manage Database Extensions
- We'll create the U_VEX_STUDENTS table, which is a 1 to 1 extension of the students table.*

* Each group table can only have a single 1:1 extension, but multiple 1:Many extensions



Manage Database Extensions

▼ Step 1: Choose Functional Area

Choose the database table you want to extend, or choose "Other" to see a longer list of PowerSchool tables or to add an independent table.

Choose the Database Table to Extend

Students ▼

Choose Workflow Type

☐ Basic Extension - Add fields to the default extension

☒ Advanced Extension - Create and manage database extensions, tables and fields



Manage Database Extensions

► Step 1: Choose Functional Area

▼ Step 2: Choose or Add New Database Extension Group for Students

Choose an existing database extension group for the selected functional area, or click "Add" to create a new database extension group. Note that a new extension group is automatically prefixed with U_ in the title.

- ☒ View only database extension groups for the current functional area (Recommended)
- ☐ View all user created database extension groups

	Extension Name	Date Created	Last Modified	Status	Edit	Delete
☐	U_FSH	11/07/2014	11/19/2014	Existing		
☒	U_STUDENTS_EXTENSION	11/07/2014	11/07/2014	Existing		

Add

Add Extension

U_VEX

Apply

Cancel

☒	U_VEX	1/9/2015	1/9/2015	Pending		
----------------------------------	-------	----------	----------	---------	---	---

You may want to delete the suggested default table name.

Manage Database Extensions

- ▶ Step 1: Choose Functional Area
- ▶ Step 2: Choose or Add New Database Extension Group for Students
- ▼ Step 3: Choose or Add New Database Extension Table for U_VEX

Choose a table to work with, or click "Add" to add a new table.
Then click "Next".

								Add
	Extension Table Name	Table Type	Description	Date Created	Last Modified	Status	Edit	Delete
☒	U_DEF_EXT_STUDENTS	One-to-one		1/9/2015	1/9/2015	Pending		



Choose a table to work with, or click "Add" to add a new table.
Then click "Next".

Add Table

Table
Name

U_ VEX_STUDENTS

Description



☐ Can have multiple records (For example
a Student might have multiple records
related to activities or programs in which
they are involved)

Apply

Cancel

*Note that you can only add one exten

▼ Step 3: Choose or Add New Database Extension Table for U_VEX

Choose a table to work with, or click "Add" to add a new table.
Then click "Next".

[Add](#)

	Extension Table Name	Table Type	Description	Date Created	Last Modified	Status	Edit	Delete
☒	U_VEX_STUDENTS	One-to-one		1/13/2015	1/13/2015	Pending		

*Note that you can only add one extended table, but you can add multiple one-to-many extensions.

[Previous](#)

3 / 4

[Next](#)

▼ Step 4: Create New Fields for Database Extension Table: U_VEX_STUDENTS

Click "Add" to add new Fields.

The database extension is not saved.

Add Field

Name

DRIVER_LICENSE

Type

String ?

Length

40

Default Value

Migrate Data From

Description

Table Details

Apply

Cancel

▼ Step 4: Create New Fields for Database Extension Table: U_VEX_STUDENTS

Click "Add" to add new Fields.

The database extension is not saved until you click "Submit".

Add

Field Name	Data Type	Default Value	Description	Status	Edit	Delete
DRIVER_LICENSE	String(40)			Pending		

Previous

4 / 4

Submit



▼ Step 4: Create New Fields for Database Extension Table: U_VEX_STUDENTS

Click "Add" to add new Fields.

The database extension is not saved until you click "Submit".

Add

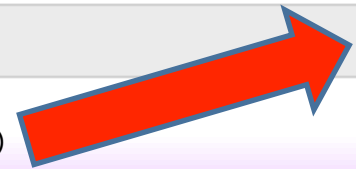
Field Name	Data Type	Default Value	Description	Status	Edit	Delete
DRIVER_LICENSE	String(40)			Pending		
PERMIT	String(40)			Pending		

Previous

4 / 4

Submit

Point of no return ☺



Create the Extension – 1:Many

- System – Page and Data Management – Manage Database Extensions
- We'll create the second table – U_VEX_VEHICLE, which is a 1 to many child of the students table.



Manage Database Extensions

▼ Step 1: Choose Functional Area

Choose the database table you want to extend, or choose "Other" to see a longer list of PowerSchool tables or to add an independent table.

Choose the Database Table to Extend

Students ▼

Choose Workflow Type

- ☐ Basic Extension - Add fields to the default extension
- ☒ Advanced Extension - Create and manage database extensions, tables and fields



▼ Step 2: Choose or Add New Database Extension Group for Students

Choose an existing database extension group for the selected functional area, or click "Add" to create a new database extension group. Note that a new extension group is automatically prefixed with U_ in the title.

- ☒ View only database extension groups for the current functional area (Recommended)
- ☐ View all user created database extension groups

[Add](#)

	Extension Name	Date Created	Last Modified	Status	Edit	Delete
☐	U_FSH	11/07/2014	11/19/2014	Existing		
☐	U_STUDENTS_EXTENSION	11/07/2014	11/07/2014	Existing		
☒	U_VEX	01/09/2015	01/09/2015	Existing		



Choose a table to work with, or click "Add" to add a new table.
Then click "Next".

Add

	Extension Table Name	Table Type	Description	Date Created	Last Modified	Status	Edit	Delete
☒	U_VEX_STUDENTS	One-to-one		01/14/2015	01/14/2015	Existing		

Add Table

Table Name

Description

☒ Can have multiple records (For example a Student might have multiple records related to activities or programs in which they are involved)

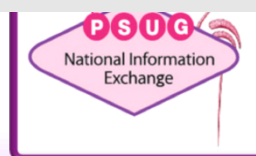
Previous

Next

► Step 4: Create New Fields for Data

Apply

Cancel



Manage Database Extensions

- ▶ Step 1: Choose Functional Area
- ▶ Step 2: Choose or Add New Database Extension Group for Students
- ▶ Step 3: Choose or Add New Database Extension Table for U_VEX
- ▼ Step 4: Create New Fields for Database Extension Table: U_VEX_VEHICLE

Click "Add" to add new Fields.
The database extension is not saved until you click "Add".

Add Field ✕

Name	VEHICLE_TYPE
Type	String ?
Length	40
Default Value	
Migrate Data From	
Description	Type of vehicle - Car, Truck, Motorcycle, etc

Note: "Type" is
a reserved
Oracle keyword

Add

Edit

Delete

▼ Step 4: Create New Fields for Database Extension Table: U_VEX_VEHICLE

Click "Add" to add new Fields.

The database extension is not saved until you click "Submit".

Add

Field Name	Data Type	Default Value	Description	Status	Edit	Delete
VEHICLE_TYPE	String(40)		Type of vehicle - Car, Truck, Motorcycle, etc	Pending		
MAKE	String(40)		Vehicle Make/Manufacturer	Pending		
MODEL	String(40)		Vehicle Model	Pending		
COLOR	String(40)			Pending		
LICENSE	String(40)			Pending		
NOTE	String(250)			Pending		

Previous

4 / 4

Submit

Building Links and Pages



Adding the 1:1 Fields

- 1:1 Table = U_VEX_STUDENTS
 - DRIVER LICENSE
 - PERMIT



Adding the 1:1 Fields w/ fragment

- 1:1 Table = U_VEX_STUDENTS
 - DRIVER LICENSE
 - PERMIT
- We'll add to “Modify Info”
 - `admin/students/modifydata.html`
- We'll use a fragment named:
`modifydata.vex.content.footer.txt`



modifydata.vex.content.footer.txt

```
<!-- Build hidden table with new driver lic and permit input -->
<table id="vexHiddenTable" style="display:none;">
  <tr id="permitRow">'
  <td class="bold">Permit #</td>
  <td><input type="text" id="permitfield" name="[Students.U_VEX]permit" value="">
</td></tr>
  <tr id="driverLicenseRow">'
  <td class="bold">Driver License</td>
  <td><input type="text" id="driverlicensefield" name="[Students.U_VEX]driver_license"
value="">
  </td></tr>
</table>

<script>
$j(function() {
  /* place driver Lic and permit rows after the locker data */
  $j("table tr").eq(4).after($j("#driverLicenseRow"));
  $j("table tr").eq(4).after($j("#permitRow"));

  /* delete the hidden table */
  $j("#vexHiddenTable").remove();
});
</script>
```



Add link to vehicle page to left nav

- The left navigation menu for a student is the more2.html page
- We'll make another fragment named: more2.vex.leftnav.footer.txt



more2.vex.leftnav.footer.txt

```
<script>
    $j(function() {
        /* Build the link for vehicles */
        var html = '';
        html += '<li>';
        html += '<a href="vehicles.html?
frn=~(studentfrn)">';
        html += 'Vehicles';
        html += '</a></li>';

        /* insert the link after the incidents link */
        $j("[href^='/admin/students/incidents/
home_student.html']").closest("li").after(html);
    });
</script>
```



Student Vehicle Page – Tlist-child

- **Tlist_child**: the stock way to interact with a one-to-many table.
- The format for the tlist_child HTML tag.
~[tlist_child:<CoreTableName>.
<ExtensionGroup>.<ExtensionTable>;
displaycols:<List of Fields>;
fieldNames:<List of Column
Headers>;
type:<FormatName>]

Student Vehicle Page – Tlist-child

```
~[tlist_child:  
<CoreTableName>.<ExtensionGroup>.<ExtensionTable>;  
displaycols:<List of Fields>;  
fieldNames:<List of ColumnHeaders>;type:<FormatName>]
```

- Needs to be inside a form tag
- displaycols = FIELDS
- fieldNames = HEADERS

Student Vehicle Page – Tlist-child

```
~[tlist_child:Students.U_VEX.U_VEX_VEHICLE;  
displaycols:VEHICLE_TYPE,MAKE,MODEL,COLOR,LICENSE,NOTE;  
fieldNames:Type,Make,Model,Color,License,Note;  
type:html]
```

Student Vehicle Page – Tlist-child

- Refer to `vex_vehicles_tlistchild.html` for the complete code for the page

Vehicle Page – Direct Access

- **DirectTable.Select**

- Tlist_child (or tlist_standalone) are tags that can be limiting. For more flexibility, you can use “Direct Access” via the DirectTable.Select tag

- You use the DirectTable.Select tag on a page where you are editing, adding, or deleting a record from a database extension table.



Vehicle Page – Direct Access

- **For a core table**, you usually pass an FRN (file record number) to the page. That is how PowerSchool knows which table and which record in that table to work with
 - `/admin/students/modifydata.html?frn=~(frn)`
 - `/admin/student/modifydata.html?frn=0011754`
 - 001 is the Students Table
 - 1754 is a fictional example of the DCID for a student



Vehicle Page – Direct Access

- /admin/student/modifydata.html?frn=0011754

- So now you can write code such as:

```
<input type="text" name="[01]home_room" value="" size="25">
```

- Because of the FRN, PowerSchool knows which table and student for this home_room field.

- Rendered view:

```
<input type="text" name="UF-0010701736"
```



Vehicle Page – Direct Access

- Database Extension Tables do not have defined table numbers.
- With the DirectTable.Select tag, you can identify the correct table and record, and then work with it much as with core tables.

`~[DirectTable.Select:<tableName>;<idColumn>:<idValue>]`

`tableName` = Name of Database Extension Table (NOT Group)

`idColumn` = name of primary key field, usually "ID"

`idValue` = the value of the primary key



Vehicle Page – Direct Access

```
~[DirectTable.Select:<tableName>;<idColumn>:<idValue>]
```

```
~[DirectTable.Select:U_APPLICATIONS;ID:41]
```

This means we're working with the record where ID=41 in the U_Applications table.



Vehicle Page – Direct Access

Normally you don't hard-code the ID value, you pass that along with a gpv. For child tables of the STUDENTS table, you pass along the FRN for the student record AND the child table record.

Your referring page would pass the data along to the edit page, probably from the results of a tlist_sql.

```
<a href="applications-edit.html?frn=~(studentfrn)&id=~(id;l)">
```

Then on your edit page, your tag would be:

```
~[DirectTable.Select:U_APPLICATIONS;ID::~(gpv.id)]
```



Vehicle Page – Direct Access

- 4 required elements:
 - The post value: applications.html?id=24
 - The DirectTable tag after the <form> tag but before any fields.
`~[DirectTable.Select:U_APPLICATIONS;ID:~(gpv.id)]`
 - Input tags: The proper way to reference a field when using DirectTable.Select is:
`<input name="[ExtensionTable]Field_Name" value="">`
 - The standard hidden input tag must be included in the form fields, usually near the Submit button. For the admin portal that tag is:
`<input type="hidden" name="ac" value="prim">`



Vehicle Page – Direct Access

- Example from our Vehicle Tracking Extension – vex_vehicles.html

```
~[tlist_sql;  
SELECT studentsdcid, id, vehicle_type, make...  
FROM U_VEX_VEHICLE  
WHERE studentsdcid = ~(rn)]  
<tr><td>  
  <a href="vex_editvehicle.html?  
    frn=001~(studentsdcid;l)&vehicleID=~(id;l)">  
...•
```



Vehicle Page – Direct Access

- Example from our Vehicle Tracking Extension – vex_editvehicle.html

```
<form action="vex_vehicles.html?frn=~(frn)" method="POST">
~[DirectTable.Select:U_VEX_VEHICLE;id::~(gpv.vehicleID)]
...
<select name="[U_VEX_VEHICLE]vehicle_type">
  <option value="Car">Car</option>
...
<input type="text" name="[U_VEX_VEHICLE]make" value="" />
...
<input type="hidden" name="ac" value="prim">
~[submitbutton]
```



Vehicle Page – Direct Access

- Adding a new record or deleting a record
 - See the **Database Extensions Advanced User Guide for PowerSchool 9.x** for details
 - ID: -1 for a new record
 - Special form with a special hidden input tag to delete

```
<input type="hidden"
name="DD-<tableName>.<idColumn>:<idValue>"
value="1">
```



Vehicle Page – Direct Access

- Explore the vex_vehicles.html and vex_editvehicle.html pages for the complete code.



Search Page – Angular GridWidget

- Link on Special Functions
- vex_vehiclesearch.html page
 - with the Angular GridWidget code, the “view”
- vex_vehiclesearch.json page
 - to provide the data, the “model”
- /scripts/vex_vehicles.js file
 - to provide the Angular module definition
 - The “controller”



Search Page – Angular GridWidget

- This is the part where I do my best to explain how an Angular Grid Widget works.



Create the Plugin



Creating the plugin

- 4 Ingredients
 - Database Extension U_VEX
 - admin/students/modifydata.vex.content.footer.txt
 - admin/students/vehicles.html
 - admin/students/more2.vex.leftnav.footer.txt



Creating the plugin

- refer to the “Advanced User Guide for Database Extensions”.
- The unpublished link to create a plugin is:

`https://<server>/admin/customization/CreatePackagePage.action`

*Note: The database extensions exist in the “user_schema_root” folder and end in .xml.



Create Package Zip File

Plugin Name

Vehicle Extension



*

Plugin Version

1.0

*

Plugin Description

Management of student vehicles

*

Publisher Name

Roger

*

Publisher Contact Email

roger@sprik.com

*

Plugin File Name

vehicle_extension_1

*.Zip

Select Individual Files

..... spectrum.js
user_schema_root
├── Activities.xml
├── u_aet_customalerts.xml
├── u_aet_liststudents.xml
├── u_vex.xml
└── U_StudentsUserFields.xml

Select Files By Search Mask

Selected Files

WEB_ROOT/admin/students/more2.vex.leftnav.footer.txt
WEB_ROOT/admin/students/modifydata.vex.content.footer.txt
WEB_ROOT/admin/students/vehicles.html
user_schema_root/u_vex.xml

Create Plugin Zip File

Install the plugin on other servers

- System – System Settings
- Plugin Management Configuration
- Click “Install” and select the zip file
- Enable the plugin



Plugin Important Notes

- The files installed by a plugin will be visible in CPM
- Updating a plugin to newer version usually should just involve deleting the old plugin and installing the new one
- Deleting a plugin removes any files it added
- Deleting a plugin will NOT remove any database extended tables and fields



